



HAL
open science

Designing for Ecosystems of Communication Apps

Carla Griggio

► **To cite this version:**

Carla Griggio. Designing for Ecosystems of Communication Apps. Human-Computer Interaction [cs.HC]. Université Paris Saclay (COMUE), 2018. English. NNT : 2018SACLS465 . tel-02157268

HAL Id: tel-02157268

<https://theses.hal.science/tel-02157268v1>

Submitted on 16 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Designing for Ecosystems of Communication Apps

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Université Paris-Sud

Ecole doctorale n°580 Sciences et technologies de l'information et de la
communication (STIC)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Orsay, le 11/12/2018, par

CARLA F. GRIGGIO

Composition du Jury :

Mme. Myriam Lewcovicz Professeur, Université de Technologie de Troyes, Tech-CICO	Présidente
Mme. Louise Barkhuus Professeur associé, The IT University of Copenhagen, Pervasive Interaction Technology	Rapporteur
M. Cliff Lampe Professeur, University of Michigan, School of Information	Rapporteur
M. John Tang Senior Researcher, Microsoft Research, Social Technologies Group	Examineur
Mme. Annie Gentes Maître de Conférences HDR, Télécom Paris-Tech, CoDesign Lab & Media Studies	Examineur
Mme. Wendy Mackay Directrice de recherche, Inria Saclay, Ex)situ	Directrice de thèse

Abstract

More and more, people communicate via not one, but a complex mix of apps. In particular, couples, close friends and families use multiple apps to express caring in diverse ways throughout the day. This calls for a new focus of study: besides observing how each app shapes communication, I argue that we need a deeper understanding of how people communicate via *ecosystems* of communication apps.

In Part I, I show that users' communication practices in one app are not only influenced by its contacts and features but also by the contacts and features in their other apps.

A first study shows that the contacts in an app affect the conversations with other contacts. To control this phenomenon, people isolate contacts in different apps: they create *communication places*, each with its own membership rules, perceived purpose, and emotional connotations. As relationships change, people move contacts in and out of their apps, driving communication places to redefine each other. People may break their places by bringing outsiders when the functionality they need exists only in one app.

A second study shows that people customize their communication apps to better express their identities, culture and intimate bonds with others. Beyond customizations, the features of each app nurture expression habits that transfer to other apps, thus influencing how people express themselves across their entire app ecosystem. App-exclusive features prevent consistent identity expressions across apps and interfere with relationship-specific communication styles.

Based on these insights, I proposed four design directions for supporting *ecosystems of communication apps*: allow multiple communication places within the same app, support relationships across apps, provide access to functionality from other apps, and enable user-owned—rather than app-exclusive—communication tools.

In Part II, I explore those design directions by repurposing three mechanisms currently available in mobile operating systems: *notifications*, which users can overlay on top of any open app; *gesture commands*, which

users could perform on any app that recognizes gestures; and *soft keyboards*, which appear in any app that accepts text input.

I repurpose notifications as peripheral awareness displays to build *Lifelines*, a dedicated communication channel for couples which shares visual timelines of contextual information, e.g. closeness to home, battery level, and steps. A longitudinal study with nine couples shows how each couple leveraged *Lifelines* in unique ways, finding opportunities for coordinating implicitly, starting conversations, and being more understanding with each other.

I repurpose gesture commands as *personal gesture shortcuts* to diverse functionality which users can perform in any app. I present a design environment and study user strategies for creating personal gestures in a comparative study of *Fieldward* and *Pathward*, two interaction techniques for creating gestures that are easy to remember and easy for the system to recognize. The results show large individual differences in users' gesture-creation strategies, reflecting their culture, their intimate bonds with special contacts and technology usage habits.

Last, I repurpose soft keyboards as *communication toolboxes* that users can carry with them from app to app. I present a design environment and explore its feasibility by building two prototypes: *The Shared Emoji Toolbox*, which allows sharing collections of emoji shortcuts, and *CommandBoard*, which combines gesture typing with gesture shortcuts to access rich sets of commands.

In conclusion, I argue that when people communicate via multiple apps, each app shapes how communication happens in others. We should shift from building isolated apps to designing mechanisms that help users preserve their communication places and express their identities and intimate bonds with others consistently across their apps.

Résumé

L'utilisation de plusieurs applications de communication au lieu d'une est de plus en plus commune. En particulier, ces applications permettent à l'utilisateur de communiquer de façons divers avec son partenaire, les membres de sa famille ou ses amis proches. En plus d'observer comment chaque application influence la communication, il est nécessaire de comprendre comment les gens communiquent au travers d'un écosystème d'applications.

Dans la première partie, je décris comment les pratiques de communication d'un utilisateur via une application sont influencées non seulement par les contacts et les fonctionnalités de cette application, mais également par les autres contacts et fonctionnalités de l'écosystème.

La première étude montre que les contacts dans une application affectent les conversations avec d'autres contacts dans la même application. Afin de contrôler ce phénomène, les utilisateurs isolent leurs contacts dans différentes applications : ils créent des "lieux de communication", ayant des règles uniques d'adhésion, des buts perçus et des connotations émotionnelles. Ces lieux de communication sont rompus lorsqu'un utilisateur ajoute un contact qui brise des règles d'adhésion d'une application, par exemple pour utiliser une fonctionnalité présente uniquement dans cette application.

La deuxième étude montre que les utilisateurs personnalisent leurs applications de communication pour mieux exprimer leur identité, leur culture et leurs liens avec d'autres personnes. Au-delà de ces personnalisations, les fonctionnalités d'une application influencent la manière dont l'utilisateur s'exprime dans d'autres applications. Pour cette raison, les fonctionnalités exclusives à une application empêchent les utilisateurs de s'exprimer de manière cohérente dans leur écosystème d'applications et interfèrent avec les styles de communication spécifiques à chaque relation.

Je propose quatre pistes pour explorer comment améliorer la communication via un écosystème d'applications : permettre la création de plusieurs lieux de communication dans une même application, soutenir une manière de communiquer propre à une relation à travers l'écosystème d'applications, accéder aux fonc-

tionnalités d'une application depuis les autres applications, et de permettre des outils de communication qui appartiennent aux utilisateurs et non pas exclusives aux applications.

Dans la seconde partie, j'explore ces pistes en réutilisant des mécanismes déjà disponibles dans les systèmes d'exploitations des téléphones mobiles : des notifications, des commandes gestuelles, et des claviers virtuels.

Je réutilise les notifications comme un affichage périphérique d'alerte pour construire *Lifelines*, un canal de communication dédié aux couples qui partage des chronologies graphiques d'informations contextuelles, comme la proximité de la maison, le niveau de batterie et le nombre de pas. Une étude longitudinale avec neuf couples montre comment chaque couple profite de *Lifelines* de manière unique, en trouvant des opportunités de se coordonner implicitement, de démarrer des conversations et d'être plus compréhensif l'un envers l'autre.

Je réutilise les commandes gestuelles comme raccourcis personnels pour diverses fonctionnalités que les utilisateurs peuvent exécuter dans n'importe quelle application. Je présente une conception envisageable et étudie les stratégies des utilisateurs pour créer des gestes personnels dans une étude comparative avec *Fieldward* et *Pathward*, deux techniques d'interaction pour créer des gestes qui sont faciles à mémoriser pour le l'utilisateur et facile à reconnaître pour le système. Les résultats montrent de larges différences individuelles dans les stratégies de création de gestes des utilisateurs, reflétant des différences culturelles et des liens intimes avec leurs contacts et leurs habitudes d'utilisation de la technologie.

Enfin, je réutilise les claviers virtuels comme des boîtes à outils de communication que les utilisateurs peuvent transporter d'une application à une autre. Je présente une vision de conception et explore sa faisabilité grâce à deux prototypes : La *Shared Emoji Toolbox*, qui permet de partager des collections de raccourcis pour des emojis, et *CommandBoard*, qui combine de la saisie gestuelle de texte avec des raccourcis gestuels pour accéder à un vaste ensemble de commandes.

En conclusion, je soutiens que lorsque les utilisateurs communiquent via de nombreuses applications, chaque application affecte la manière dont l'utilisateur communique dans les autres applications de l'écosystème. Nous devrions cesser de concevoir uniquement des applications isolées mais concevoir des mécanismes qui aident les utilisateurs à préserver leurs lieux de communication et à exprimer leur identité et leur intimité avec leurs proches de manière cohérente dans leur écosystème d'applications.

Acknowledgments

The Ex)situ lab at Inria was a vibrant, exciting place where I learned immensely from a lot of people over the last three years. First and foremost, I want to thank my thesis supervisor, Wendy Mackay, for her passionate dedication to teaching me how to do research. She also introduced me to the academic culture of the community that drives our field. I also thank her for running the weekly Ph.D. seminars where I got to learn immensely not only from her but from the rest of the Ex)situ students and faculty. She will always be a role model as a researcher, a supervisor and lab director.

I also want to thank my labmates for so many mind-opening conversations, for participating in pilot studies, for reading my paper drafts, for supporting me during tough times and even helping me survive the French bureaucracy. For this, I want to give a special mention to Alexandra Merlin, Miguel Renom, Jean-Philippe Riviere, Téo Sanchez, Andrew Webb, John MacCallum, Stacy Hsueh, Alex Zinenko, Nolwenn Maudet, Nicolas Taffin, Philip Tchernavskij, Nacho Avellino and Mai Ciolfi. Also, thanks for all the funny emoji and bot anecdotes on Slack <3.

I was also very fortunate to collaborate and co-author papers with amazing researchers throughout my Ph.D. I thank Joseph Malloch, Xiaojun Bi and Jessalyn Alvina for inviting me to participate in projects that greatly complemented the core vision of this thesis. I thank Midas Nouwens for being my qualitative research buddy—I hope our *lifelines* keep crossing in the future. Last but not least, I want to give a special thank to Joanna McGrenere, who I met during her sabbatical at Inria and became like a thesis godmother to me. Her critical input pushed and enriched every project in which we collaborated. She has a special talent: she helps you solve problems on your own by asking questions rather than suggesting solutions. I hope to learn how to do that for my future students some day.

I would also like to thank people I met before starting my Ph.D. who supported me throughout these years: Hendrik Heuer, for cheering me up and during stressful times with optimism; Marta Ceccinato, for being a "Ph.D. student role model" since I met her in CHI 2013; Mario Romero, for teaching me good research practices during my masters that I will apply for life; Carlos Lombardi and Nicolás Passerini for being my

friends as well as my programming, teaching, and life mentors; and Michel Beaudouin-Lafon for teaching me HCI fundamentals and sparking my interest in the field.

I thank my defense jury for investing their time in reading my work and attending my presentation—their constructive input provided me with rich, diverse perspectives that will guide my future work.

I will always be in debt with all the people that volunteered to participate in my studies. I am enormously thankful to them for their time and their insightful comments and stories—this thesis would not exist without them.

Finally, I will always be grateful to my mom, my dad, Vicky, and Pablo for always supporting my adventures and encouraging me to spread my wings. I also thank Lydia for her support, her generosity and for being an example of a strong, independent woman. I also thank Noelia, Laura, and Daniela for always cheering for me from far away and for being my “life peers”. And Germán: thank you for always being there and for helping me find my calling.

This thesis was funded by the ERC grant N° 321135: “CREATIV: Creating Co-Adaptive Human-Computer Partnerships”.

Contents

1	<i>Introduction</i>	1
	1.1 <i>Thesis Statement</i>	3
	1.2 <i>Research Approach</i>	3
	1.3 <i>Thesis Overview</i>	5
	1.4 <i>Publications & Collaborators</i>	7
	PART I USER PRACTICES IN ECOSYSTEMS OF COMMUNICATION APPS	11
2	<i>Background</i>	13
	2.1 <i>Self-presentation</i>	13
	2.2 <i>Relationship maintenance</i>	16
	2.3 <i>Co-adaptation between users and technology</i>	18
3	<i>Communication Places in App Ecosystems</i>	21
	3.1 <i>Related Work</i>	22
	3.2 <i>Method</i>	23
	3.3 <i>Results and Discussion</i>	25
	3.4 <i>Limitations</i>	35
	3.5 <i>Conclusion</i>	36
4	<i>Customizing Communication Apps: Breakdowns and Workarounds</i>	37
	4.1 <i>Definitions</i>	38
	4.2 <i>Related Work</i>	41
	4.3 <i>Method: Interviews</i>	44

4.4	<i>Method: Story Questionnaires</i>	46
4.5	<i>Customizations to communication apps</i>	49
4.6	<i>Breakdowns</i>	57
4.7	<i>Workarounds</i>	61
4.8	<i>Discussion</i>	64
4.9	<i>Limitations</i>	68
4.10	<i>Conclusions</i>	68
5	<i>Design directions for supporting ecosystems of communication apps</i>	71
5.1	<i>Implications of communicating via ecosystems of apps</i>	71
5.2	<i>Allow multiple communication places within the same app</i>	73
5.3	<i>Support relationships across apps</i>	74
5.4	<i>Provide access to functionality from other apps</i>	75
5.5	<i>Enable user-owned communication tools</i>	76
PART II DESIGNING FOR ECOSYSTEMS OF COMMUNICATION APPS		79
6	<i>Repurposing Notifications</i>	81
6.1	<i>Lifelines: a dedicated communication channel for couples</i>	82
6.2	<i>Related Work</i>	84
6.3	<i>The Lifelines technology probe</i>	86
6.4	<i>Method</i>	90
6.5	<i>Results and Discussion</i>	94
6.6	<i>Implications for design</i>	103
6.7	<i>Limitations</i>	106
6.8	<i>Conclusions</i>	106
7	<i>Repurposing Gesture Commands</i>	109
7.1	<i>Personal gesture shortcuts for communication app ecosystems: User Scenario</i>	110
7.2	<i>Challenges around creating personal gesture shortcuts</i>	113
7.3	<i>Related Work</i>	114
7.4	<i>Fieldward and Pathward</i>	116
7.5	<i>User Study</i>	118
7.6	<i>Results</i>	124
7.7	<i>Gesture Creation Strategies</i>	125

7.8	<i>Discussion</i>	130	
7.9	<i>Conclusions</i>	133	
8	<i>Repurposing Soft Keyboards</i>	135	
8.1	<i>The Trojan Keyboard: a soft keyboard with personal communication toolboxes</i>		136
8.2	<i>Supporting identity expression</i>	138	
8.3	<i>Supporting the expression of intimate bonds</i>	141	
8.4	<i>Extending gesture keyboards</i>	141	
8.5	<i>Prototypes: The Shared Emoji Toolbox and CommandBoard</i>		143
8.6	<i>Barriers to Implementation</i>	148	
8.7	<i>Conclusions</i>	151	
9	<i>Conclusions</i>	153	
9.1	<i>Contributions</i>	153	
9.2	<i>Co-adaptation in communication app ecosystems</i>		156
9.3	<i>Blurring the walls between isolated communication apps</i>		157
9.4	<i>Communication substrates</i>	159	
9.5	<i>Limitations and Future Work</i>	160	
A	<i>Fieldward vs. Pathward: Quantitative Results</i>		165
A.1	<i>Methodological Challenges</i>	166	
A.2	<i>Experiment Design</i>	168	
A.3	<i>Quantitative Performance Measures</i>	170	
A.4	<i>Quantitative Self-Reported Measures</i>	172	
B	<i>CommandBoard vs. Markdown</i>		175
B.1	<i>Inline Gesture Shortcuts</i>	175	
B.2	<i>User Study</i>	176	
B.3	<i>Participants</i>	177	
B.4	<i>Hardware and Software</i>	177	
B.5	<i>Procedure</i>	178	
B.6	<i>Quantitative Performance Measures</i>		181
B.7	<i>Data Collection</i>	182	
B.8	<i>Results</i>	182	
B.9	<i>Preferences Study</i>	185	
B.10	<i>Discussion & Conclusions</i>	187	

1

Introduction

People increasingly communicate via not one, but many communication apps (Duggan and Ellison, 2015; Smith and Anderson, 2018). They create *communication app ecosystems*: personal collections of communication apps and social media sites they access from their phones and computers, which connect them with hundreds of contacts with whom they hold diverse types of relationships. In particular, couples, close friends and families communicate via multiple apps in parallel to express caring in diverse ways and stay connected throughout the day (Yang et al., 2014). Extensive research on mediated communication studied how particular apps (including social media sites) and communication channels shape communication (e.g. Church and de Oliveira, 2013; Grinter and Eldridge, 2001; McRoberts et al., 2017; Nardi et al., 2000; O'Hara et al., 2014) and relationship dynamics (e.g. Kim et al., 2007; Shklovski et al., 2015; Zhao et al., 2012). Nevertheless, people increasingly adopt multiple communication apps for their daily communication, and the implications of communicating via many apps might differ from the knowledge we have about how each shapes communication. This demands a deeper understanding of how people communicate via *app ecosystems*, and design novel communication technology accordingly.

Recent research started investigating how users leverage multiple apps for their daily communication. Cramer and Jacobs (2015) show how couples communicate through “*a myriad of options*”, which allows them to express themselves in different ways and convey urgency or importance when sending the same message via many apps. They suggest

expanding couples' collection of communication means by designing novel apps that complement, rather than compete, with their existing ones. Scissors and Gergle (2013) show how couples switch back and forth between face-to-face and diverse mediated channels for strategically mediating disputes. They argue that switching between different channels (e.g. voice calls, instant messaging, email) helps partners manage their arguments and suggest *not* merging them into comprehensive platforms or apps. Zhao et al. (2016) discuss how people handle communication in their *social media ecologies*, balancing tensions between the content and audience of their messages with the different available functionality on each social media site. They argue that "*as users increasingly mix and remix use of different communication platforms, focusing on only one channel may conceal important insights*". Moreover, they suggest designing social media platforms that acknowledge each other and facilitate mechanisms for *segregating* and *permeating* contacts and content between them. Sleeper et al. (2016) describe how users share personal online content through different apps, and highlight how users sometimes *chain* multiple apps to leverage composite functionality or reach broader audiences beyond what one app can offer. They encourage designing apps that facilitate multi-channel strategies rather than trying to provide all possible features in a single app.

In summary, people benefit from mediating their communication via multiple apps but also need better mechanisms for strengthening or blurring the walls between the contacts of each app, *stitching* the functionality of different apps together, and leveraging the diverse capabilities distributed across their many apps when one app cannot meet specific communication needs.

In this thesis, I further investigate how *ecosystems* of communication apps affect relationships and expression, and I propose design directions for supporting communication practices that exceed the boundaries of each app. I repurpose three available mechanisms in today's mobile operating systems (notifications, gesture commands, and soft keyboards) to shift from a model of isolated apps to ecosystems of complementary apps and system-wide communication tools.

1.1 Thesis Statement

People increasingly communicate via not one, but many communication apps. In particular, couples, close friends and families use multiple apps side by side to express caring in diverse ways and stay connected throughout the day. Communication apps provide opportunities and constraints for communication that play a fundamental role in how we nurture relationships today. However, communication apps are designed around features, not relationships, and despite people's use of them side by side, they compete rather than complement each other. I argue that we need a deeper understanding of how people mediate their relationships and express their identities via *ecosystems of communication apps*, and that we should design highly appropriable communication tools that support users' communication style beyond what each app allows and restricts.

1.2 Research Approach

I developed my thesis in the context of the ERC CREATIV grant for creating co-adaptive human-computer partnerships. Co-adaptation is a phenomenon in which users *adapt* to the interactive system by learning its capabilities and constraints at the same time that they *adapt* it to their own needs (Mackay, 1990b, 2000). By designing systems that are easy to learn and to adapt, we help users transition from novices to experts, and we transform systems into powerful, flexible tools that users may even use in unanticipated ways.

In this thesis I incorporate co-adaptation not only as a descriptive theory but as a generative guideline to explicitly support users in *adapting to* and *adapting* the system (Beaudouin-Lafon and Mackay, 2018). I triangulate methods from theoretical, design and observational perspectives (Mackay and Fayard, 1997) to support the multidisciplinary character of my research, combining practices from design, engineering, and cognitive and social sciences (Figure 1.1) to inform the design of novel mechanisms for supporting ecosystems of communication apps.

From a **theoretical perspective**, I engage with theories and methods from HCI as well as from social and cognitive sciences, to ground the design of communication tools that belong to users rather than

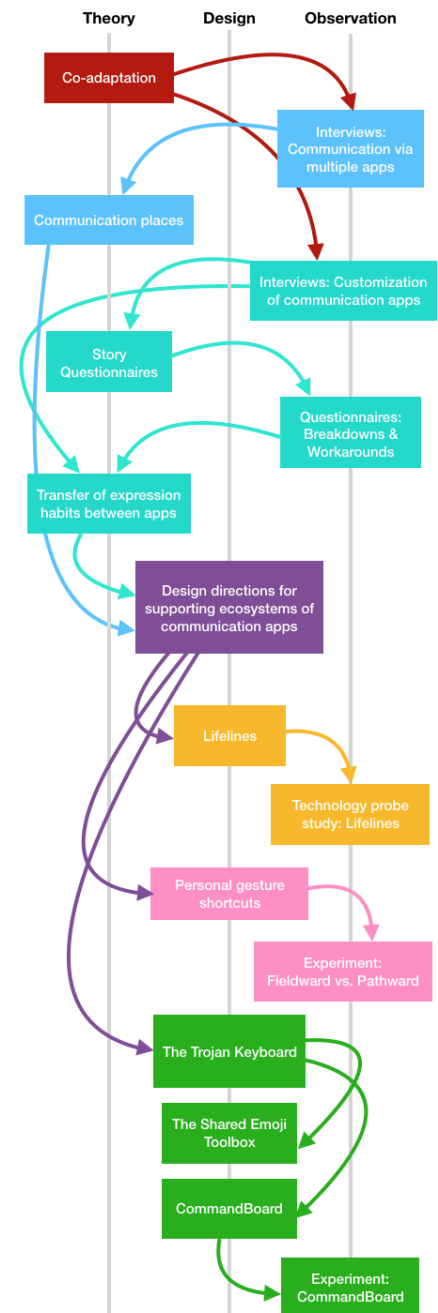


Figure 1.1: Method triangulation in this thesis.

■ Chapter 3
 ■ Chapter 4
 ■ Chapter 5
■ Chapter 6
 ■ Chapter 7
 ■ Chapter 8

apps, granting users greater control over their mediated expression, and thus, their mediated relationships:

- I take co-adaptation in interactive systems as a framework that inspires design ideas and research questions about how users *adapt* and *adapt to* their apps.
- I incorporate theories from social psychology such as impression management, hyperpersonal CMC and dialectical contradictions to help me interpret results from empirical studies, and design technologies that mediate communication in novel but grounded ways.
- I propose models of user behavior concerning communication app ecosystems and technology appropriation within close relationships, from which I derive a set of new design directions.
- I analyze qualitative data from interviews and questionnaires using thematic analysis (Braun and Clarke, 2006). For analyzing data from controlled experiments, I use multivariate analysis of variance for dependent measures and non-parametric tests for self-reported ordinal data.
- I propose a new questionnaire model called *story questionnaires* for collecting user stories in a structured way, at a larger scale than what interviews allow.

From an **observational perspective**, I conduct qualitative and quantitative research to inform and evaluate design artifacts and to shape my design vision:

- I conduct critical incident interviews (Flanagan, 1954; Mackay, 2002) to collect user stories about their mediated communication practices across multiple apps, and about their motivations and habits around customizing messaging apps.
- I conduct an online survey using *story questionnaires* to further collect stories about customizations and technology breakdowns linked to computer-mediated expression within close relationships.
- I conduct two lab experiments to evaluate three novel interaction techniques.
- I deploy a technology probe in a one-month study with nine couples to observe how they incorporate continuous sharing of contextual

information (e.g. battery level and closeness to home data) in their daily communication.

From a **design perspective**, I envision and build artifacts for exploring my proposed design directions, for assessing their technological feasibility, and for further observing users' behavior while interacting with novel technology:

- I design and implement *Lifelines*, a mobile app for sharing multiple, persistent streams of contextual information within couples.
- I propose *The Trojan Keyboard*, a design envisionment of a soft keyboard that augments users' communication app ecosystem with a shared space of custom communication tools, unique for each of their closest contacts.
- I implement *The Shared Emoji Toolbox*, a simplified version of the *The Trojan Keyboard*, to document how current app architectures allow and prevent app-independent communication tools.
- I collaborate in the design and implementation of *CommandBoard*, a set of techniques for augmenting gesture keyboards with a command input space so that they interpret commands apart from generating text output.
- I propose a design envisionment around *personal gesture shortcuts*, showing how users could treat gestures as cross-app tools that connect their communication functionality and contacts system-wide.
- I collaborate in the design and implementation of *Fieldward* and *Pathward*, two interaction techniques to help users create their own memorable gesture shortcuts, so the system reliably recognizes them.

1.3 Thesis Overview

Part 1: User practices in ecosystems of communication apps

Chapter 2 presents the theoretical background that grounds the research questions and design approaches of this thesis, reviewing related work to self-presentation, relationship maintenance and co-adaptation

between users and technology.

Chapter 3 describes a study about how people distribute their contacts in multiple but similar communication apps and introduces the concept of *communication places*: idiosyncratic constructs that users add to their communication apps, comprising membership rules, perceived purposes and emotional connotations. Users' communication places both *shape* and *are shaped by* the relationships that each app mediates, so when relationships evolve, the communication places in users' app ecosystem evolve as well.

Chapter 4 describes how people customize their messaging apps to express their identities, cultures and intimate bonds with others. I discuss how app-exclusive customizations and features prevent a consistent identity expression across apps, and interfere with relationship-specific communication styles. I argue that the features of each app nurture expression habits that transfer to other apps, thus influencing how users express themselves across their entire app ecosystem.

Chapter 5 proposes four design directions to better support communication via app ecosystems, grounded in the findings from Chapter 3 and 4: 1) allow multiple communication places with the same app; 2) support relationships across apps; 3) provide access to functionality from other apps; and 4) enable user-owned communication tools. Moreover, I propose exploring the last three of these design directions by repurposing mechanisms currently available in mobile operating systems: *notifications*, which can be accessed system-wide and can be overlaid on top of the app in use; *gesture commands*, which could execute functionality consistently across apps; and *soft keyboards*, which can carry diverse input functionality from app to app.

Part 2: Designing for ecosystems of communication apps

Chapter 6 illustrates how to repurpose *notifications* to support ecosystems of communication apps. It presents *Lifelines*: a dedicated communication channel for couples that repurposes notifications as peripheral awareness displays, sharing visual timelines of contextual information, i.e. closeness to home, battery level, steps, media playing, and traces of texts and calls. By using notifications, *Lifelines* provides functionality that belongs to the couple rather than any particular app.

Chapter 7 illustrates how to repurpose *gesture commands* to support

ecosystems of communication apps. It presents a design envisionment of *personal gesture shortcuts* for associating functionality to contacts and accessing diverse functionality from any app. It also presents two interaction techniques, Fieldward and Pathward, that help users create personal gestures that are easy to remember and easy for the system to recognize. A comparative study offers insights about users strategies for creating personal gestures.

Chapter 8 illustrates how to repurpose *soft keyboards* to support ecosystems of communication apps. It presents a design envisionment of *The Trojan Keyboard*, a soft keyboard extended with *personal communication toolboxes* that users can access from any messaging app. I also illustrate extended *gesture-typing* keyboards, which could allow users to refine and extend their communication toolbox by combining it with gesture-typed commands. I further present two prototypes, *The Shared Emoji Toolbox* and *CommandBoard*, and identify current implementation barriers as well as opportunities for building technology probes that can support early studies on how user-owned communication tools affect mediated communication.

Chapter 9 concludes with a summary of my contributions, discusses the main findings in terms of co-adaptation and broader HCI perspectives, and points to future work for addressing the limitations of this thesis and exploring new research directions.

1.4 Publications & Collaborators

Some of the content of this thesis produced publications in top international venues.

The interview study in Chapter 3, as well as a simplified version of some implications for design from Chapter 5 appear in:

Midas Nouwens, **Carla F. Griggio**, Wendy E. Mackay. *“Whatsapp is for family; Messenger is for friends”*: *Communication Places in App Ecosystems*. In Proceedings of the 35th Annual ACM Conference on Human Factors in Computing Systems (**CHI’17**).
DOI: <http://dx.doi.org/10.1145/3025453.3025484>.

Midas Nouwens led this project while I co-supervised his internship

with Wendy Mackay. He proposed the initial research questions, and we contributed in equal parts to the interview study, its analysis and presentation. I am the main contributor to the implications for design.

The interaction techniques and a version of the study presented in Chapter 7 and Appendix A appear in:

Joseph W Malloch, **Carla F. Griggio**, Joanna McGrenere, Wendy E. Mackay. *Fieldward and Pathward: Dynamic Guides for Defining Your Own Gestures..* In Proceedings of the 35th Annual ACM Conference on Human Factors in Computing Systems (**CHI'17**).

DOI: <http://dx.doi.org/10.1145/3025453.3025764>.

Video: <https://www.youtube.com/watch?v=F-Z8uj6GCSY>

I joined this project after Wendy Mackay and Joseph Malloch had designed and implemented the main concept around the Fieldward and Pathward techniques. I led the design of the experiment, and improved their design and implementation based on findings from pilot studies. I conducted the experiment, and analyzed the quantitative and qualitative data together with Wendy Mackay and Joanna McGrenere. The whole team contributed to writing the paper; I mainly contributed to the Methods and Results section.

The *CommandBoard* prototype described in Chapter 8 and the study in Appendix B appear in:

Jesallyn Alvina, **Carla F. Griggio**, Xiaojun Bi, Wendy E. Mackay. *CommandBoard: Creating a general-purpose command gesture input space for soft keyboards.* In Proceedings of the 30th Annual Symposium on User Interface Software and Technology (**UIST '17**).

DOI: <http://dx.doi.org/10.1145/3126594.3126639>.

Video: <https://www.youtube.com/watch?v=HNdI9EmxAvC>

Jesallyn Alvina and Wendy Mackay developed the main concept, and I helped refine the concept and design in later discussions and video prototyping sessions. Jessallyn Alvina implemented most of the technology, and I adapted Octopocus (Bau and Mackay, 2008) to integrate it to *CommandBoard*. Xiaojun Bi provided the gesture keyboard that enabled the prototypes. Jesallyn Alvina, Wendy Mackay and I designed the user study, for which I analyzed and described the qualitative results.

The *Lifelines* technology probe described in Chapter 6 and its study will soon appear in:

Carla F. Griggio, Midas Nouwens, Joanna McGrenere, Wendy E. Mackay. *Augmenting Couples' Communication with Shared Timelines of Mixed Contextual Information*. To appear in the Proceedings of the 37th Annual ACM Conference on Human Factors in Computing Systems (**CHI'19**).

Video: https://www.youtube.com/watch?v=jSodZ3Ac_-k

Wendy Mackay proposed the initial concept, which I refined based on recent related work and current technological limitations and opportunities. I led the project, designed and implemented the technology probe, and conducted the study. Midas Nouwens contributed to the design of the probe, its study, participant recruitment and data analysis. Joanna McGrenere and Wendy Mackay also contributed to the data analysis. The whole team participated in iterative discussions for identifying the main contributions of the study as well as writing the paper.

Chapter 4 is a more recent study that I have not published yet. I initiated the project, designed and conducted all studies, and analyzed the data. I discussed aspects of the methodology (especially *Story Questionnaires*) with Joanna McGrenere and Wendy Mackay, who also helped identify and articulate the main findings.

PART I

USER PRACTICES IN ECOSYSTEMS OF COMMUNICATION APPS

2

Background

This chapter describes the theoretical background that grounded and inspired the studies that follow.

2.1 Self-presentation

Goffman's self-presentation theory (Goffman, 1959) proposes that when people encounter others, they adapt their behaviour to project a particular image of themselves, attempting to control the impression that others form about them. People may want to project different images depending on whom they encounter and on what "looks good" to others. Goffman describes this behaviour with a dramaturgical metaphor: To selectively self-present to others, people "perform" acts in front of audiences. When performing for a particular audience, people engage in *front stage* behaviour, displaying a self that fits the appearance and manners expected of them. The *back stage* is where the performer can drop the act, relax and be more authentic. In everyday face-to-face encounters, people intentionally exercise strategies of impression management: they put on different "faces" for different audiences, choose aspects of themselves to display in front stage and keep other aspects in back stage.

In everyday computer-mediated communication (CMC), people can

leverage the technological limitations of the medium to have more control over their self-presentation. Walther (1996) proposed the *hyperpersonal model* of interpersonal communication, stating that CMC grants people greater opportunities for *selective self-presentation* (Walther, 2007; Walther and Burgoon, 1992), i.e. managing the impression one wants to achieve by carefully planning messages and concealing facial expressions and physical appearances. This model aligns with Hollan and Stornetta (1992) who propose to design communication tools so that they mediate communication “*beyond being there*”, i.e. to explore new technology-enabled opportunities rather than trying to imitate face-to-face conversations at a distance. The text-based, asynchronous character of CMC allows people to exaggerate desirable impressions and conceal others. O’Sullivan (2000) studied channel choice between telephone, answering machine, email and letters and reached conclusions that align with Walther’s selective self-presentation theory: in some situations, people prefer channels that increase the ambiguity of a message to “*shade and shape impressions*”. During face-to-face interaction, people give non-verbal cues via facial expressions and hand gestures that enrich communication, but may be hard to control voluntarily. In contrast, CMC affords strategic use (or absence) of non-verbal cues such as emoticons (Dresner and Herring, 2010) or emoji, which people may use deceptively to mask their true feelings (Kelly and Watts, 2015).

The greater control that CMC grants on people’s self-presentation inspired a wide range of studies about how they leverage the capabilities and limitations of technology to shape impressions. Compared to face-to-face interaction, people form less complete but more intense impressions about their communication partners (Hancock and Dunham, 2001). While narrow at first, impressions developed in CMC can develop into more complete views of the other over time (Walther, 1993). Non-verbal cues such as emoticons and emojis play a key role in user’s impression management (Constantin, 2002; Glikson et al., 2017), especially since, unlike facial expressions, users always produce them consciously and intentionally (Dresner and Herring, 2010). For example, negative emoticons may lead to negative interpretations of a message (Walther and D’Addario, 2001). On the other hand, positive emoticons help to form positive impressions (Byron and Baldrige, 2007), although overusing them may cause the opposite, as if the sender was bored, weakly engaged or hiding their real intentions (Yoo, 2007).

The impression about others in CMC interactions can influence the communication style, and vice-versa. Xu et al. (2007) suggested that

“one would use more emoticons when he/she regards the receiver to be inventive, curious and open to new ideas and change”. Moreover, paralinguistic cues such as style, politeness and emoticons influence impressions formed in CMC group conversations (Postmes et al., 1998). For example, people may try to convey positive impressions in chat rooms by demonstrating mastery of the chat environment and using custom screen names (Becker and Stamp, 2005). While people may intentionally use emoticons to elicit positive impressions, Glikson et al. (2017) found that in work-related e-mails, emoticons can form negative first impressions and perceptions of low competence, undermining future information sharing.

More recent research discussed users’ strategies to manage their self-presentation in social media platforms. Hogan (2010) revisits Goffman’s presentation of self to discuss impression management in the age of social media: apart from *performing* during synchronous communication, social media users create artifacts that they *exhibit* for asynchronous interactions (e.g. seeing a picture from last year in someone’s timeline). Social media sites also blur the boundaries between offline social networks (i.e. audiences), driving users to self-censor their posts on Facebook (Vitak and Kim, 2014) or adapting them to the *lowest common denominator* across their many collapsed audiences (Davis and Jurgenson, 2014; Hogan, 2010). Such efforts attempt to convey a consistent impression to diverse audiences; in contrast, McRoberts et al. (2017) argues that Snapchat Stories¹ “allows people to share quotidian & out-of-the-ordinary moment to experiment with self-presentation to a self-selecting audience (from close ties to strangers) without needing to maintain a consistent presentation of self”.

¹ Snapchat Stories are a sequence of ephemeral pictures and videos that are only visible to the user’s friends for 24 hours.

In this part of my thesis, I study how communication apps shape (or challenge) self-presentation and interpersonal interaction with the goal of finding opportunities for design. While most of the mentioned research focused on group chats and social media, I study users’ personal ecosystems of communication apps (including video calls, group messaging, social media platforms, etc) with a strong emphasis on mobile messaging apps and one-on-one conversations. Complementing the state of the art on the characteristics of CMC that people control to strategically manage impressions, I highlight the aspects that people *cannot* control, analyzing how they hinder self-presentation practices.

2.2 Relationship maintenance

CMC enabled novel and diverse ways in which people can develop and maintain relationships. For example, the introduction of mobile messaging via SMS texts and apps (e.g. WhatsApp) facilitated convenient, asynchronous communication at any time, anywhere. This encouraged people to send more mundane, small messages to each other throughout the day, blurring the beginning and end of conversations and creating a state of continuous *connected presence* (Licoppe, 2004) or *dwelling* (O'Hara et al., 2014). Instant messengers with online status information enabled contacts to be aware of each other's routines, helping navigate availability and enhance social connections (Nardi et al., 2000). Other forms of CMC such as social media sites also support initiating relationships without prior offline interaction, finding information about offline acquaintances (e.g. Ellison et al., 2011) and crystallizing offline relationships that would have otherwise remained ephemeral (Ellison et al., 2007). Moreover, the relationship dynamics that emerge in CMC may help designers inform how to design software: while a common goal in HCI is to design software that helps perform tasks faster and easier, researchers found that *effortful communication practices* can be highly meaningful in close relationships (Kelly et al., 2017). For example, Riordan (2017) found that investing time and effort in using emoji may help maintain and enhance social relationships online.

Within the HCI community, Shklovski et al. (2015) propose a relational lens based on *dialectical contradictions* (Altman et al., 1981; Baxter, 1988,9) when studying communication technologies. They focus on three salient tensions between opposing emotional requirements that shape relationship dynamics: *Autonomy-connection*: for relationships to exist, people must give up some autonomy, but too much connection risks losing their individual identities and thus damages the relationship. *Openness-closedness*: relationships need disclosure of information to develop intimacy, but also some closedness to protect personal sensitivities. *Predictability-novelty*: relationships need stability, but lack of novelty can lead to boredom and emotional distance. Zhao et al. applied this lens to study how couples manage the tensions between their individual and relational needs on Facebook (Zhao et al., 2012), and suggested to see beyond how people use technology and attend to how technology interacts with the *dialectical contradictions* of their relationships. Shklovski et al. (2015) suggest that designing with relationships in mind implies offering opportunities to manage such contradictions

via three types of coping strategies: *selection* (prioritizing one aspect of the contradiction over the other), *segmentation* (disclose information only to whom it is relevant), and *integration* (an attempt to address both opposing aspects of the contradiction simultaneously). For example, WhatsApp's *last seen* status, which displays the time a user last used WhatsApp, lacks support for *segmentation*: users can only disclose their *last seen* status to everyone or to nobody. My research aligns with the goal of designing with relationships in mind, with a special focus on close relationships, which often mediate their communication via a myriad of apps (Cramer and Jacobs, 2015).

As people adopted more diverse forms of CMC over time, researchers started to study how they leverage the availability of different types of CMC technologies. Some studies showed that communicate via different channels for different relational goals, with different audiences. For example, Kim et al. (2007) found that mobile phones reinforced strong social ties, while text-based conversations were mostly used for expanding relationships with weak ties. Houser et al. (2012) observed how users applied relationship maintenance strategies via different channels: positivity prevailed in communication via email and social networks, while openness dominated instant messaging. Other studies focused on how different CMC technologies fit along the stages of a relationship. Yang et al. (2014) discussed how Facebook is a popular medium in the early stages of a relationship, whereas instant messaging and mobile phone calls were adopted later as the relationship progressed.

I believe that cross-cutting relational dialectics and relationship maintenance strategies, users' self-presentation and aesthetic expression (e.g. linguistic style, use of emoji and media) may also influence their relationships, e.g. by developing secret jokes around an emoji (Wiseman and Gould, 2018) or orienting the selection of emoji to ensure it is relevant to the recipient (Kelly and Watts, 2015). As people increasingly adopt many communication apps, their mediated expression and communication capabilities increase, but so does the complexity of their app ecosystems and reachable audiences. With the goal of inspiring grounded designs that favor close relationships, the upcoming chapters show how users manage diverse relationships across their ecosystem of communication apps and how multiple apps contribute to (or challenge) the emergence relationship-specific communication styles.

2.3 Co-adaptation between users and technology

Mackay argues that the use of information technology is a co-adaptive phenomenon: users *adapt to* the capabilities and limitations of software, and at the same time, they *adapt it* to their own needs (Mackay, 2000). She studied co-adaptation of software in the context of user customizations on MIT's Project Athena Unix-based workstations, identifying triggers to customizing (e.g. noticing own repeating patterns, seeing an exciting customization by someone else, retrofitting changes from systems updates) as well as barriers that encourage users to keep their default settings (e.g. lack of time, poor documentation) (Mackay, 1991). She describes user customizations as the materialization of individual patterns of use that continue to influence user behavior over time, portraying them as a great example of co-adaptation in the use of software:

The possible customizations are constrained by the design of the software but may also be modified by users in unanticipated ways, as they appropriate the software for their own purposes. (Mackay, 1990b)

In the particular case of communication technology, the literature reviewed above also provides examples of co-adaptation: when users restrict their communication style to asynchronous text, they repurpose those limitations to selectively self-present to others; they learn to use the limited collection of emojis to express their emotions, but they also repurpose their meanings based on inside jokes with others.

Beyond describing mediated communication as a co-adaptive phenomenon, in this thesis I treat co-adaptation as a generative model that inspires new research questions and design directions: apart from adapting to and repurposing the limitations and capabilities of communication technology, how do users explicitly *adapt* (i.e. customize) software to their own needs? How can we design communication technology that not only influences how communication happens but also allows *users* to tailor it to idiosyncratic communication styles?

Moreover, when users adapt their software in collaborative environments, their customizations not only influence their future use of technology but also how others use it and adapt it. Mackay's 1990a early studies on sharing configuration files of a Unix system in the workplace portrayed customization as a social activity, showing how some workers enjoyed helping others to install or modify their customizations. Nardi and Miller (1991) studied collaborative use of spread-

sheets found that co-workers distributed computational tasks among users with different programming skills, through which they also shared domain knowledge. Echoing Mackay's recommendations for supporting customization sharing mechanisms, Draxler et al. (2012) proposes design principles for collaborative appropriation of software in the context of software development teams, including: sharing tool knowledge (i.e. browse the tools that other peers have installed), appropriation awareness (i.e. communicate to others a recent solution to a problem or a convenient customization), and peer installations (i.e. exchange already installed tools between peers). MacLean et al. (1990) argued that to encourage system tailoring (i.e. customizing), users should be immersed in a tailoring culture. They created self-contained, customizable buttons that users could create with their own functionality and share by e-mail. More recently, Haraty et al. (2017) studied online customization sharing ecosystems (e.g. for customizing Minecraft and Sublime Text), identifying different user roles (e.g. sharers, re-users, reviewers, problem reporters, requesters, helpers, publicizers, and packers) and motivations for sharing customizations (e.g. building a reputation in the community, having an online backup of personal customizations).

In the next chapters, I study how people *adapt* and *adapt to* their communication apps as a means to better understand the role of personal ecosystem of communication apps in how they mediate their relationships and express their identities.

3

Communication Places in App Ecosystems

The mediated communication landscape has expanded significantly over the past decade, from a handful of distinct forms of communication (phone calls, email, texts) to hundreds of communication apps. *Mobile Instant Messaging (MIM)* alone comprises a wide variety of apps, including *Messenger*, *WhatsApp*, *iMessage*, *KakaoTalk*, *WeChat*, *Line*, *Viber*, and more. Researchers often point to the qualities inherent in each app, such as its functionality or cost, to explain the effect of the medium on users' communication patterns. For example, Rost et al. (2016) discuss the link between the presence of message history and chit-chat, and Grinter and Eldridge (2001) describe how the cost of text messaging can lead to short, abbreviated, and dull exchanges. Yet these apps are often very similar, with nearly identical functionality. Even so, Cramer and Jacobs (2015) found that users consistently use equivalent apps such as *WhatsApp* and *Messenger* alongside each other, but in idiosyncratic ways. This suggests that knowing the inherent qualities of an app is insufficient for understanding how it affects communication patterns.

Scissors and Gergle (2013) argued we should consider channel selection relative to the other apps in the user's ecosystem rather than focusing on just one, and Jung and Lyytinen (2014) showed how it is the ecology—the relations among the different media and their surroundings—that shapes channel choice. We believe this perspective should be extended beyond channel choice: Considering a user's

ecosystem of apps should help us to better understand how use of mostly identical communication apps diverges over time, which factors create and maintain these differences, and how the ecosystem influences the relationships among communication partners.

This chapter presents an interview study with 18 participants that explores (1) why users distribute their social relationships across different apps; (2) how individual communication apps are used with respect to other communication apps in the user's ecosystem; and (3) how users establish patterns that maintain these distributions.

3.1 Related Work

Multiple communication apps for one contact

Research discussing multiple communication channels often focuses on how users choose among various options available. Only recently have researchers asked how users take advantage of this multiplicity and how the use of one communication app informs the use of another. Some studies describe the process of channel switching or channel blending: how one coherent conversation takes place over multiple channels. In professional contexts, Su (2009) revealed how employees go through "communication chains", i.e. interactions through different channels in rapid succession. In personal contexts, Scissors and Gergle (2013) described how couples switch back and forth between different channels during conflicts to leverage the particular qualities of different forms of communication. Cramer and Jacobs (2015) extended this beyond conflicts and showed how couples combine different channels throughout their relationship to communicate importance or add emotional value to a message. These studies focus on the use of multiple apps for the purpose of a single conversation or contact, but leave unexplored the use of multiple apps to manage multiple relationships.

Multiple communication apps for multiple contacts

Studies that focus on how people take advantage of multiple channels to manage multiple contacts are often related to email use. A large-scale survey with university employees by Capra et al. (2013) revealed

that 88% of respondents had at least two email accounts. The most common reason for having multiple email accounts is to let users separate their personal and work lives (Cecchinato et al., 2016). Gross and Churchill (2007) expanded on these email studies and argued that research should take into account the user's full range of communication means. They examined both email and *IM (Instant Messaging)* to see how users employ multiple addresses and accounts to manage interruptions, deal with usability issues, and separate business and personal contacts. However, like most email-related research, the focus is on professional relationships rather than social communication and the many apps dedicated to its support.

One communication app for multiple social contacts

Most studies of how users manage their social relationships are conducted in the context of social networking sites. The presence of multiple contacts in the same environment is shown to cause tension if users belong to different social spheres, with the diversity of contacts corresponding to the level of tension (Binder et al., 2009; Zhao et al., 2016). Shklovski et al. (2015) discuss how users deal with these tensions through selective self-disclosure or each site's built-in privacy settings. These studies show that such conflicts arise because person-to-person communication is public to all other contacts in the environment. However, this does not explain why users distribute contacts across apps when information is *not* broadcast, as with most dedicated communication apps.

3.2 Method

We conducted an interview study to investigate how people manage their growing collections of communication apps to manage their social relationships. We are particularly interested in why they decide to use specific apps to communicate with particular contacts or groups.

Participants

Participants included 10 women and 8 men ($n = 18$) between the ages of 18 and 40 (mean = 25) who came from twelve different countries in Asia, Europe, North America and South America. They were currently living in Argentina, France, the Netherlands, Sweden, the United Kingdom, or the United States. The participants' occupational backgrounds include: clinical supervisor, student, secretary, social media manager, journalist, copywriter, lecturer, unemployed, and pre-school teacher.

Procedure

We recruited participants through social media, snowball sampling and by approaching strangers in public spaces. No monetary compensation was offered.

We conducted semi-structured interviews using a variation of critical incident technique (Mackay, 2002). We asked participants to describe specific, recent, memorable stories about the interplay among different people (contacts) in their communication apps. Interview questions focused on who the participants communicate with, the channels they choose for their different contacts, what rules they use and how they feel about those channels, and how their use of a channel corresponds to the other channels they use.

Examples of questions include *"When was the last time you deleted someone from an app?"* and *"Did you answer someone through a different app than the one they contacted you with?"*. The resulting stories served as the point of departure for learning more about their overall associations with an app and the role it plays relative to the other apps in their ecosystem. Each interview lasted between 45 and 90 minutes. Eleven interviews were conducted live; seven interviews were conducted via Skype. All interviews were recorded and transcribed.

3.3 Results and Discussion

Two researchers independently and iteratively analysed the transcribed interviews using thematic analysis (Braun and Clarke, 2006). Concepts were identified using open coding and grouped into categories using axial coding. Categories that emerged from stories included: “feeling”, “rule”, “practice”, “breakdown”, and “workaround”. For example, one participant’s *WhatsApp* felt close (feeling) which meant she only wanted people that were important to her (rule) and she reinforced this by withholding her phone number from those she did not want (practice). When the participant’s father contacted her via a coworker’s phone (breakdown) she blocked the co-worker and deleted the messages (workaround). The interviews were re-analysed to confirm that these categories captured all salient data points.

We found that participants communicated via a wide variety of devices, including smartphones, laptops, desktops, and tablets. They used *Messenger* (17), *WhatsApp* (15), *SMS* (15), *Skype* (13), *Snapchat* (6), *Slack* (4), *Tinder* (3), *Grindr* (2), *GChat* (2), *iMessage* (2), *Telegram* (2), *Couchsurfing Messages* (1), *GroupMe* (1), *Line* (1), and *LinkedIn Messages* (1). Participants mentioned other communication platforms, such as social networks and e-mail; our analysis here focuses on private messaging using mobile (or multi-platform) apps.

Contacts contaminate apps

The contacts present in an app can influence other communication experiences, even if individual conversations are isolated from each other and no information is shared across contacts. Participants view apps as shared communication spaces and said that the presence of a particular contact can change the content of other conversations, such as when they copy a joke or match another person’s writing style. The presence of a contact in an app can also change their emotional state during conversations. One participant, who was relaxed when talking to a friend, became stressed in the presence of a past lover. Another’s conversation with her partner felt intimate until she noticed a co-worker in the same app.

The contacts in a shared communication space can also indirectly affect individual communication activities by changing participants’ com-

munication patterns. For example, one participant initially used *WhatsApp* to send two friends “whimsical” (P5) voice messages and impersonations. As *WhatsApp* became more popular, other people started to contact him there. Their presence made him uncomfortable, so he and his two friends switched to another app:

I didn't really use WhatsApp in a functional way. [The app] felt sullied because those voice messages are what I used WhatsApp for. It was specific people and it was all light hearted and fun. They just got kind of drowned out by all the other people who moved there. (P5)

The presence of other contacts also damaged or improved relationships with other contacts in the same app. One participant avoided a channel because of a barrage of messages from a particular conversation, which angered the other contacts that she inadvertently ignored. Another participant began to spend more time on an app to keep in touch with her travelling uncle, which revitalized her relationship with another contact:

I ended up with my previous boyfriend because of Skype ... because I was there he started talking to me. I'm pretty sure that if my uncle had never travelled the world this wouldn't have happened. (P12)

These results suggest that a shared communication space is not just a collection of individual conversations that occur in isolation. Rather, because these contacts exist in the same environment, the user's relationships with their contacts become subtly intertwined. The addition of a single person can alter what the user talks about with the other contacts, how the user feels during the conversation, how often the user enters the communication space, and how much time is spent within it. These particular ways of interrelating ultimately make up the dynamics of each relationship, which are affected by the particular collection of contacts present in the shared communication space.

Users control contacts

Participants were actively invested in controlling the presence of contacts in their apps, usually by leveraging and appropriating the app's technical features and constraints. We identified three primary strategies for keeping apps in their preferred state. *Preventing entry*
Fourteen participants described how they tried to prevent a person from entering their app environment. Most just lied, saying either that they did not have the app, or they did not use it. Some asked their

friends not to give out the identifier (such as username or phone number) without their explicit permission. One participant explained his process for deciding where to locate a contact:

When I meet a new person they do get a little interview like... "Would you be a psycho texter?" So I test the waters out a little bit where I'm like ... "Yeah I'll give you my number." or "Let's keep you on Snapchat or on Grindr." (P5)

(Re)moving intruders

Ten participants removed contacts from or moved contacts to an app. Participants removed contacts by blocking the person or permanently deleting the conversation. They either moved contacts implicitly, for example by replying via a different app or taking a long time to respond, or explicitly, for example by telling the contact where they should message them:

If [my friend] would message me on WhatsApp I would reply but I wouldn't do it myself. I would message him via Facebook Messenger. And then if he would continue to message me on WhatsApp I would tell him to not contact me on WhatsApp anymore. (P1)

One participant went to great lengths to ensure that particular contacts were included rather than excluded in an app. When neither his closest friend nor his parents had *Snapchat*, he said it "felt wrong", as if he was "cheating them". He convinced his friend to buy a better phone so she could use *Snapchat*. For his parents, he "sat them down to explain how it worked" and "coached" them:

When people don't have a certain channel where I'd like them to be in it feels like a shame and that it doesn't fit that they don't have it. It's something they should have and we should communicate that way. (P5)

Tolerating outsiders

When participants did not or could not exclude a contact from an app—either by preventing them from entering or (re)moving them after they did—they appropriated the features of the app to shape how these contacts appear:

If I'm not comfortable with them having contacted me, I just don't add them as contacts. So that just their number comes up. It's not that I don't want any contact with them, but I obviously didn't feel comfortable with them being part of my WhatsApp. I think it is a way of reserving WhatsApp and my phone for important personal information. (P6)

You can archive conversations and then they only pop up when something new happens. Or when you think to send them a new text. So I archive other people

so that my close friends and family are always at the top of the chat list. (P5)

Interestingly, the participants were aware that these practices had no impact on the contact's actual ability to communicate with them via the app. One called it "*being rude in private*" (P6), but acknowledged that it was just a way to manage the experience of the contact's presence.

In summary, these results suggest that people actively create and shape personalised communication spaces by managing contacts within and across their apps. Participants took advantage of the similar communication apps to, as one participant put it, "*build boundaries*" (P7) between social relationships. This echoes Chalmers and Galani's (2004) notion of the "*seamfulness*" of technology, where users appropriate its limitations to their benefit. Here, users appropriate the siloed nature of communication apps to manage and control contacts.

Defining Communication Places

Stable patterns of contact management occur across the different communication apps, based on how users organise and separate their contacts to manage social relationships. In addition, 13 participants attributed particular identities to specific communication apps, which fulfilled distinct roles in their everyday communication.

It's more likely that if someone sends me a message on Facebook and I am connected with them on WhatsApp that I will reply on WhatsApp. For some reason I associate WhatsApp with a much more easy, more immediate medium. And I have no idea why because functionality wise there is no real difference. In my mind it's just not what Facebook Messenger is about, it has nothing to do with the functionality. (P6)

In my head, WhatsApp is slow and old. I'm not sure why. I only really use it for groups, like the family WhatsApp group. So to talk to my friend there is weird. Whereas Facebook Messenger is white and happy and empty. It feels way more airy and I use it for all my friends. (P7)

We call apps with particular identities *communication places* i.e. personal and idiosyncratic constructs that users build on top of communication apps, which in turn shape their subsequent communicative actions and experiences.

We define communication places according to their:

- *Membership rules*: who belongs to them,
- *Perceived purpose*: what they are for, and
- *Emotional connotations*: how they feel to the user.

Communication places are personal to the user rather than inherent to the app. The same app may have different or even contradictory rules, purposes or connotations for different users. For example, some participants found text messaging “*a little more personal*” (P16) or “*intimate*” (P5), whereas others used texting “*more for practical reasons*” (P6), or “*for logistics*” (P1). Another said: “*Texts are like gold. So I don’t use it*” (P7).

Communication places affect how participants interpret the same message or behaviour:

I deleted this guy on Tinder. But not on WhatsApp. It feels more personal if I did it there. (P3)

Someone had passed my number on and they contacted me on WhatsApp and I didn’t feel comfortable with that. It would have been OK for them to contact me through Facebook, because on Facebook I am more comfortable with that type of contact. (P6)

Communication places reflect and sometimes affect the quality of the relationship between a participant and their contact:

I have a friend who I talk to almost every day. We work together and we live in the same building, but if I want to talk to him I message him on Facebook Messenger. We are close, but we are not close enough for WhatsApp. (P1)

I had Slack with my co-workers first and then my boyfriend said he wanted to try it so we started to use it. Using the same app makes it feel like they are the same kind of relationship in a way, but they are not. (P12)

I gave out my number to people on Tinder and added them on WhatsApp but then eventually that started feeling like I was letting them into my personal bubble. It’s like you’re setting a second step in your relationship with those people. (P7)

Communication places are constantly redefined, not only as the user engages with the app over time, but also in response to new app features, changed phone numbers or important life events such as moving to a new country. Each communication place has a reciprocal relation-

ship with its underlying app: On one hand, the user’s actions and experiences with the app shape the communication place and its attributes; on the other hand, the communication place influences which actions and experiences the user permits in the app (see Figure 3.1).

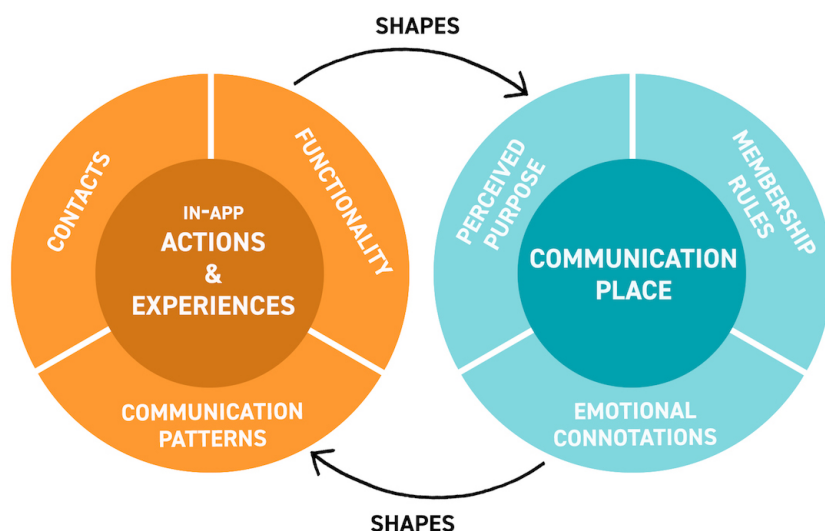


Figure 3.1: The reciprocal relationship between actions and experiences in the communication app and the communication place that is created on top of it.

Our use of the term “communication place” is influenced by Harrison and Dourish’s distinction between “space” and “place”. They define space as “*the structure of the world*”, in contrast to place, which they define as “*a space which is invested with understandings of behavioural appropriateness, cultural expectations, and so forth*” (Harrison and Dourish, 1996). Our findings support this distinction: Here, apps provide the structure for communication, independent of the user, whereas communication places encompass the rules, roles, and feelings that users apply to their apps.

Harrison and Dourish’s work addressed common practices for building *collaborative* virtual environments. As such, their definition and discussion of place centres around how users create places *collectively*. Our results show how *individual* users can also create personal places. Places are not only developed together with others *through* technology, but also by individual users *with respect to* their particular use of the technology.

Breaking Communication Places

Participants constantly tried to negotiate between the communication place they desire and the environment imposed on them by the communication app's design or their contact's behaviour. Mismatches between communication places and contacts create friction. Sometimes, users succeed in resolving them through the prevention, (re)moving, and tolerating strategies described above. However, nearly all participants experience situations where they cannot find a solution, and thus must shift the boundaries and settle for less-than-ideal communication places.

Relationships are nuanced —apps are crude

All but one participant who mentioned blocking, deleting, or preventing a contact from entering an app, also apologetically explained that it was not because of the contact, but because they were simply “out of place”. The social undesirability of these boundary-building strategies was further illustrated by the fact that five participants stopped themselves from removing a contact from an app because they felt it was too rude:

I wasn't sure what to do, whether I should block him. Because he is a very nice person, but... he is just one of those darlings you don't want to kill that is now permanent in a sense in my WhatsApp. And so I scroll down and I see his name. And it's not that I hate him, but I just don't want to see him when I go through my contacts and have to think about him. (P3)

One participant summarized how most participants perceived these measures:

It's of course very easy to ignore and block people, but those are very crude measures. It's just not very nice and not really acceptable to myself, even though it is really easy to do via these apps. So that is really a rule I create for myself, that I want to still be a good person and so I have to tell these people like “hey sorry I don't think this will work out” rather than just blocking them and being done with them. There aren't enough nuances to what is possible. (P7)

Users can only block, remove or delete contacts to remove a contact from a communication place without talking to them, but these methods are considered too absolute. Participants want more nuanced options that range from “I never want to see them” to “I just want to be less aware of them” or “I would like to see them once a month”. Human relationships require more nuance than is currently offered by the design of these apps and users compromise their values or communication places as a result.

Relationships are fluid —apps are inflexible

Five participants struggled to account for the dynamic nature of their relationships. While participants may initially put a contact in the right communication place, they may become “out of place” as the relationship changes. For example, P5 had two *Snapchat* accounts: one under a fake name for online friends who knew about his bisexuality and one for offline friends who did not. These communication places were kept strictly separate so he could be open with himself “*without it being traceable back to real life*” (P5). However, over time his relationships with both groups changed:

When I started to come out to my friends, switching back and forth between the two accounts became hard and I started to have those people on the fake account as friends who I no longer wanted to have as just online acquaintances anymore. (P5)

Merging them, however, required him to “*come out backwards*” to his online friends because he had to explain why they were in the “*fake account*” to begin with.

Another participant struggled to deal with the presence of a recently deceased family member in the communication place she established for close friends and work contacts: “*I didn’t want to be confronted with them when I scrolled down my WhatsApp. It was a sad thing to see on a regular basis, to be reminded*” (P6). However, she felt uncomfortable deleting the contact and history because “*it felt too sudden*” and she did not want to lose her precious messages. Eventually she “*needed to confront it*” and compromised:

I went through all the messages and I wrote down in a book any of the ones I liked, anything they’d said. I made a note of those in a little book to keep because I didn’t want to lose them. Then I deleted the history. (P6)

Contacts and message histories are currently locked inside apps, which can cause problems when relationships change. Over time, a contact’s app membership might be revoked or restored multiple times based on the changing nature of the relationship or communication place. This fluidity is inhibited by the way conversation histories are coupled with the app in which they occurred. Sometimes breaking that link and abandoning the built-up history and communication habits is worth the cost, as in the two examples above. Other times, the user will choose to compromise their communication place rather than enforce its boundaries.

Relationships are unique —features are generic

Five participants reported breakdowns caused by their need to access or escape a certain feature for a particular contact. For example, P4 has a rule to keep all *Grindr* contacts in that app. However, he often violates this rule so that he can receive notifications, a function available only in the paid version:

With a guy from Grindr who I only met once, I don't want to give him my number because then he has the power to show up on my lock screen and I don't trust him with that. But I still do sometimes. (P4)

This breach makes him “*very uncomfortable*” because friends can (and did) see the explicit messages he receives.

Additionally, apps often impose the same functionality on all contacts within its boundaries. For example, P7 started to avoid *WhatsApp* after she added *Tinder* contacts there, because *WhatsApp* would tell them when she was last online—which made her anxious. However, this also meant she missed out on her family’s *WhatsApp*-group conversations:

So I changed that feature where you can see when I was online or read the message. And that was much better because then I could respond to my family without feeling like “you were online and you didn't respond, why aren't you responding”. But it also meant I couldn't see if my family was online or read my messages, so eventually I switched it back on again. And it would have been better if I could have done that for individual people. (P7)

A participant’s need for a particular feature can force them to compromise their communication places in order to meet their relationship needs. Because features exist at the app-level rather than at the contact-level, participants cannot selectively apply features based on the requirements of a particular relationship. For example, users who create a communication place for their close friends and family may end up sacrificing functionality; and users who want a particular feature to communicate with one contact may end up “contaminating” a communication place that was meant for another group.

Discussion Summary

Our findings demonstrate how mediated communication is influenced not only by each app’s technical characteristics and features, but also by the *communication places* that users establish to manage their social relationships.

We show that:

- The presence of particular contacts in an app affects communication with other contacts;
- Users purposely distribute contacts across communication apps to control their impact;
- Users establish personal strategies for maintaining this distribution of contacts;
- Users construct *communication places* with membership rules, perceived purpose, and emotional connotations that affect the meaning of messages, the appropriateness of behaviour, and relationships between users and contacts.

A few participants describe the above activities as a linear process. However, most viewed them as interwoven activities with reciprocal effects. In general, the collection of individual activities and experiences within an app serves to establish the corresponding communication place, and at the same time, the communication place shapes which activities and contacts appear within its boundaries. Because the communication place affects who or what is allowed *inside*, it simultaneously pushes other contacts *outside* its boundaries. This in turn influences the “placeness” of the other apps in the participant’s ecosystem (Figure 3.2). In other words, users not only create an ecosystem of apps, but also an ecosystem of communication places.

Although the relationship between an app and its corresponding communication place is reciprocal, the relationships among different communication places in the ecosystem may not be. Membership rules for one communication place can easily push certain contacts into other apps, but may not receive contacts in return. For example, *Telegram* may serve as the communication place for a romantic partner, and drive acquaintances to *iMessage*. If the membership rules for *iMessage* change to include ‘friends and family’, the *Telegram* communication place may not be modified to include them as well.

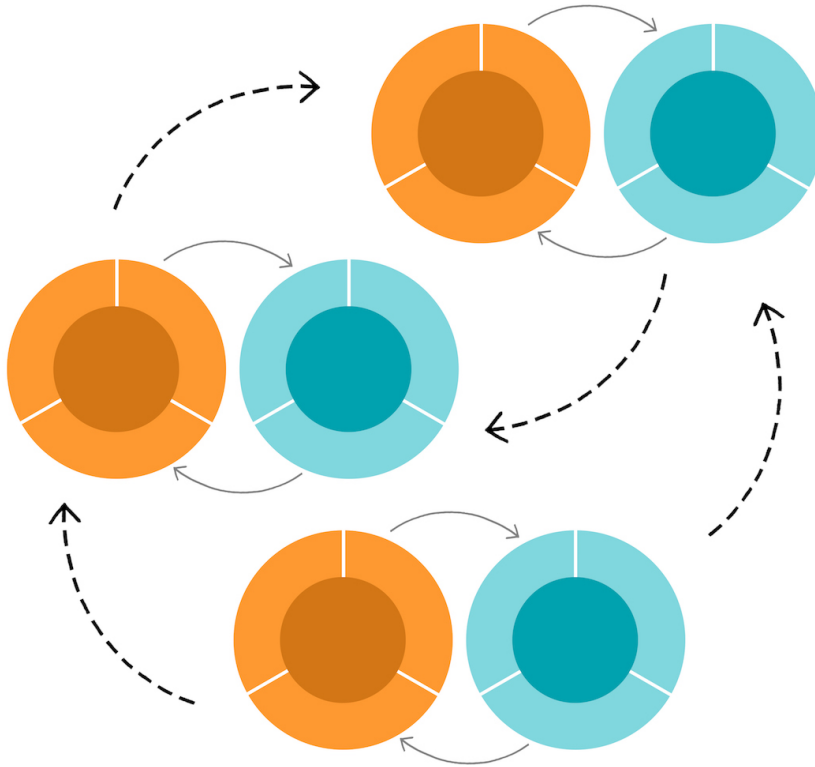


Figure 3.2: A user's ecosystem of apps and communication places: each influences the other.

3.4 Limitations

We do not claim that all users create communication places or manage relationship boundaries through communication apps. Nor do we believe that users are necessarily conscious of their attempts to control their communication environments. In this study, two participants did not create any rules or establish practices around their communication channels and never felt that the app environment affected their conversations. P2 communicated only with four close contacts and did not see the need to segregate them on different apps. P8, a freelance journalist, communicated with a large number of people, but used a “hodgepodge, make-shift system” instead of clearly delineated communication places and boundaries. For example, he freely exchanges *Skype* information with people he meets at a conference just so he can “engineer a non-awkward end to a conversation”, even if he has no intention of staying in touch. Despite using *Skype* daily for work, he is clearly far less protective of his communication environment than other participants. The fact that not all users create communication places, however, could be due to their lack of support in current communication

apps.

3.5 Conclusion

Users create personal app ecosystems that often include a variety of communication apps, many with highly similar features. These similar apps fulfil distinct communication roles within the ecosystem as users develop divergent use patterns for each. We found that the presence of specific contacts within a particular app changes the use of that app, and shapes its role relative to other apps within the communication ecosystem. We also found that users carefully consider which apps to use with which contacts, based on the app's role in the ecosystem.

Most users develop idiosyncratic associations with specific communication apps, based on their previous activity and experiences within the boundaries of the app. We call these *communication places*: Users appropriate the features and technical constraints of their apps to construct personal communication environments with unique membership rules, perceived purposes, and emotional connotations. Users shift the boundaries of their communication places to accommodate changes in their contacts' behaviour, the dynamics of their relationships, and the restrictions of the technology.

Users employ various strategies for managing the influence of individual contacts, including preventing certain people from joining an app, removing them from or moving them to other apps, or appropriating an app's features to affect how these contacts appear within the app. Unfortunately, current apps are not designed to support the creation and maintenance of communication places, and users often end up fighting the technology. Despite their best efforts, users sometimes end up with broken communication places, causing tension and damaging their personal relationships.

Communication places can be seen as blueprints for the user's ideal communication environment. Understanding the rules and emotions associated with communication places offers insights into users' communication needs and desires, as well as inspirations for the design of effective communication technology.

4

Customizing Communication Apps: Breakdowns and Workarounds

Users' app ecosystems include many apps that, at their core, seem the same—they support texting, calls, and exchanging diverse media. Nevertheless, they also feature differences that allow users to express themselves in diverse, rich ways across their app ecosystems (Cramer and Jacobs, 2015). For example, the ephemeral character of Snapchat encourages the sharing of mundane, spontaneous pictures in contrast to more flattering, groomed pictures in Facebook (Rost et al., 2016). Studies also showed that users adopt particular features of communication apps into their mediated communication practices and needs. For example, they use emojis to express emotions and clarify meaning (Cramer et al., 2016). While such studies focused on how apps shape mediated communication, i.e. how users *adapt to* the capabilities of their communication apps, we still know little about how users explicitly *adapt their apps* to fit their own communication style and needs (Mackay, 1990b).

Users may also repurpose app features into idiosyncratic communication practices, such as redefining the meaning of an emoji to reflect an inside joke with someone close (e.g. sending 🍕 to say “I love you” as in Wiseman and Gould, 2018) or, as seen in the previous chapter, leveraging the isolated nature of apps to draw boundaries between contacts

by distributing them in different *communication places* (Nouwens et al., 2017). Such *technology appropriations* illustrate how users adapt to the available features of an app and, through use, turn them into more personal means of communication. I believe that studying users' efforts to render their mediated communication more personal can provide rich inspirations for the design of communication technologies that help users adapt them to their own relational and expressive needs. Thus, I am interested in investigating not only how users repurpose their communication apps, but particularly in how they *customize* them.

In this chapter, I investigate the role of customizations in mediated communication, with a special focus in users that communicate via multiple apps in parallel, each with their own customization options. First, I review related work on expression in mediated communication and customizations. Then, I describe an interview study, followed by a survey using *story questionnaires*, a new method to collect stories of lived experiences online. Then, I describe the results of a thematic analysis of all data, which provide insights into how users customize their apps, the role of those customizations in their expression and relationships, the *breakdowns* (Winograd et al., 1986) that interfere with their expression and the *workarounds* they find to meet their communication needs across different apps. Last, I discuss how app-exclusive customization options and communication features nurture but also challenge users' communication practices across their many apps.

4.1 Definitions

This section first introduces key vocabulary relevant to the present study.

I call *expressive media* to emoticons and all pre-defined visual media that an app offers to send in a conversation, such as emojis, stickers, and animated GIFs. Expressive media can also come from outside an app, as when a user takes images of *memes* from the Internet to send in a conversation. I call it *expressive media* because they are visual means that users adopt as non-verbal cues that help them express themselves beyond what they can convey with text.

Emoticons are the oldest resource users have appropriated to add non-verbal cues to their textual, computer-mediated communication (Dres-

ner and Herring, 2010). They are combinations of text characters that evoke a facial or bodily expression, e.g. :) for a smiley face. Today, some apps detect common emoticons and translate them to emojis (e.g. replacing :) for 😊).

Emojis are a standardized collection of simple ideograms of facial expressions, hand and body gestures, and other objects and symbols including food, animals, sport items, and flags. Emojis are characters encoded in the Unicode Standard, which today features 2,789 emojis. As emojis are internally text, apps normally allow their use anywhere where text input is allowed, e.g. profile names, status messages, and conversation titles. Despite having their semantics standardized, the visual implementation of emojis depend on each operating system and app. Thus, the emojis available in one app often look different than the emojis of another app. Moreover, not all apps give access to all standard emojis. While it is not possible to add custom emojis to the Unicode Standard, some apps allow users to import small images and treat them as custom emojis (e.g. Slack).

Stickers are rich, graphical depictions, sometimes animated, of emotions, gestures, greetings, and other expressions. Unlike emoji, they cannot be used inline with text (Figure 4.1). They are not standardized, so each user builds their own collection of stickers in each app. Not all mainstream communication apps support them (e.g. WhatsApp, Slack, or HipChat). Those that do, offer “sticker packs” for users to add to their collection, or allow to import individual stickers from conversations (e.g. WeChat). In general, apps have their own “stickers market” and users can only install stickers from there, but Telegram and WeChat support importing external, custom sticker packs. Some apps like Google Allo and Bitmoji support customizing stickers with an avatar of the user (Figure 4.2). Bitmoji stickers can be accessed from within Snapchat, or from custom keyboards such as GBoard, which allows sending the stickers to any app.

Animated GIFs are looping animations, often showing short extracts from videos that became *memes*. Websites such as Giphy¹ allow searching and browsing animated GIFs and facilitate links to share them. They also help users create their own. Today, most apps offer a GIF search functionality (some source their GIFs from Giphy, e.g. Messenger and Slack) to help users send GIFs in their conversations.

I differentiate *expressive media* from apps’ *expressive functionality*, although they often are tightly coupled. I call *expressive functionality* to any app capability that enables expression, from writing text messages

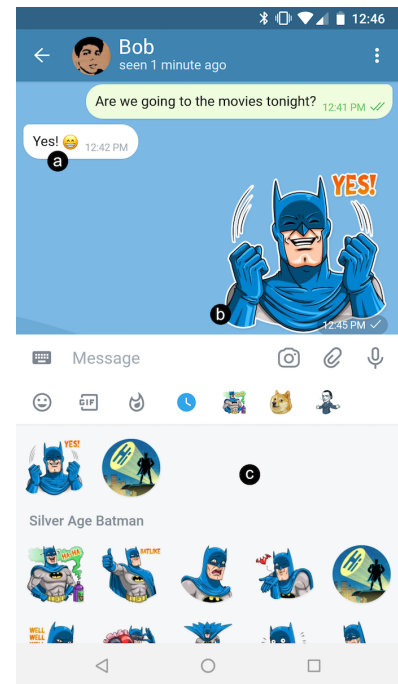


Figure 4.1: Emojis and stickers in Telegram. a) An emoji, used inline with text. b) A sticker. c) The stickers menu, showing the “Silver Age Batman” sticker pack.



Figure 4.2: Collection of Bitmoji in Snapchat, featuring my own avatar. Bitmoji’s stickers (<https://www.bitmoji.com/>) feature an avatar of the user in different scenarios, expressing diverse emotions.

¹ Giphy: <http://www.giphy.com>

and sending pictures to decorating self-made videos with hand-drawn doodles. Some expressive functionality depend on using expressive media. The two most relevant examples to the present study are *photo decorations* and *reactions*.

Photo decorations are text, hand-drawn doodles, emojis and stickers that users can *paste* on top of a picture before sending it (e.g. in Figure 4.3). Normally, these decorations can also be added to videos. Not all apps offer this functionality, and some offer less types of decorations than others. For example, Telegram only supports hand-drawn doodles, and Slack offers no decorations at all. Messenger and Snapchat allow users to add emojis and stickers, but a) the emojis available for decorations in Messenger look different than those in Snapchat, and b) Messenger offers a handful of stickers, while Snapchat offers a huge collection, including *Bitmoji* (Figure 4.2). This example shows how Snapchat and Messenger share the same functionality but differ in the expressive media available through that functionality.



Figure 4.3: Picture decorated with Snapchat, using a Bitmoji sticker (left) and text (right).

Reactions are (most often) emojis that users can attach under an existing message. Unlike replying to a message with an emoji, reactions do not take a new line in the conversation. Reactions are aggregated when more than one user reacts with the same emoji (as in Figure 4.4). Few apps support them, e.g. iMessage, Messenger and Slack, but their implementations vary. For example, iMessage allows users to react with six pre-defined symbols (Figure 4.6), Messenger with 5 pre-defined emoji (Figure 4.5), and Slack with any emoji available in the app, including custom ones. This is another example of apps that offer the same functionality, but using different expressive media.

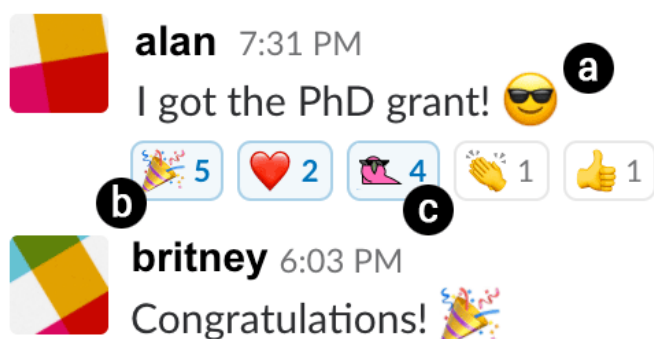


Figure 4.4: Reactions in Slack. Alan's message got thirteen reactions of five different emoji. a) Regular emoji. b) Emoji as a reaction (5 people reacted with the popourri emoji). c) Custom (uploaded by a user) animated emoji as a reaction .

4.2 Related Work

Adapting to communication apps and expressive media

Previous research studied how different apps shape communication in different ways. For example, the cheap, unlimited messages via WhatsApp opened the door to longer, more personal conversations compared to short, dry SMS texts used for coordinating activities or important messages (Church and de Oliveira, 2013). Also, distributing conversations in multiple channels helps organizing parallel conversations about different topics, useful for couples keeping track of errands, bills and social activities (Andalibi et al., 2017) or researchers coordinating conference travels and co-authoring papers (Perkel, 2017).

Users learn to express themselves through the expressive media available in their communication apps, which affects the outcomes of a mediated conversation (e.g. Hsieh and Tseng, 2017). People have adopted emojis for “adding additional emotional or situational meaning, adjusting tone, making a message more engaging to the recipient, conversation management, and relationship maintenance” (Cramer et al., 2016). Zhou et al. (2017) highlight that often, users can convey the intended meaning of a message only by including emojis or stickers, which allow them to personalize their communication in ways that would have not been possible with just text. Beyond expressing emotions, users may send stickers for functional and strategic purposes, e.g. to “look like a nicer person”, to replace greetings with a sticker, or to express their identities (Lee et al., 2016). Sugiyama (2015) describes how Japanese teens use them to “manage the communication climate as well as to construct and express their aesthetic self”. Miltner and Highfield (2017) shows how GIFs contribute in “the performance of affect and the demonstration of cultural knowledge”. Tolins and Samermit (2016) observe that animated GIFs are often used as complete responses to a message or as *co-speech demonstrations* that complement a message with visual, non-verbal cues (similar to when we move our hands while speaking). They argue that *texters* adopt GIFs featuring actions and expressions performed by others as “demonstrations of [their] own bodily expressive behaviour”.

While these studies offer insights about how apps shape communication, i.e. how users *adapt to* technology to personalize their mediated expression and self-presentation, I study how *users* shape communication apps.

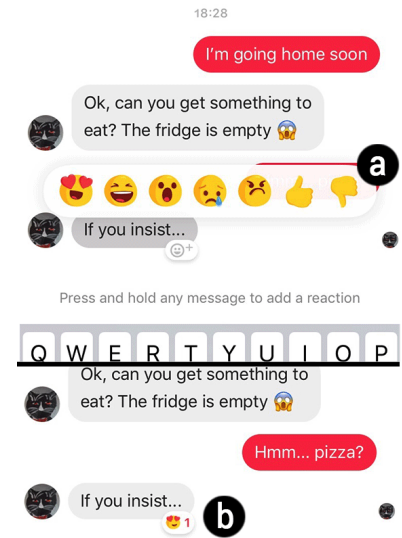


Figure 4.5: Reactions in Messenger. a) Long-pressing a message reveals the reactions menu, with 6 fixed emoji options. b) The last message has a “heart eyes” emoji reaction.

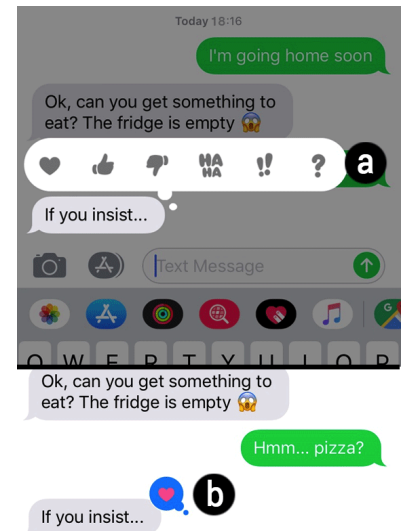


Figure 4.6: Reactions in iMessage. a) Long-pressing a message reveals the reactions menu, with 6 fixed emoji options. b) The last message has a “heart” emoji reaction.

Repurposing expressive media

Research shows how users repurpose text and expressive media to meet their idiosyncratic communication needs. A classic example are emoticons: text characters used to depict facial and bodily expressions. Park et al. (Park et al., 2013) describe how emoticon styles differ across western and eastern cultures, and how users vary the normative forms of an emoticon to express nuanced emotions, e.g. while :) means “happy”, a user may type :)))) to express greater happiness; while T_T signifies crying, a user may type T____T to express “crying a lot”. From the receiver’s perspective, Hankok and Dunham (2001) show how emoticons help to form an impression of the sender’s personality.

Repurposing of expressive media often relates to relationship maintenance practices. For example, the sender of a message may pick an emoji to reflect the intimate bond they have with the receiver (Cramer et al., 2016). Communication partners may even repurpose the meaning of an emoji when its appearance evokes an inside joke, e.g. using 🐟 because “Fish is our nickname of each other” (participant quote from Wiseman and Gould, 2018), 🍌 to show empathy², or 🦑 to signal a pedantic attitude³.

Unlike emoticons which rely on text, the availability and aesthetics of emojis, stickers, and GIFs depend on each communication app. For example, emojis look different across different apps and operating systems (Figure 4.7). These differences can change the perceived meaning or tone of an emoji between different apps, social media sites or even mobile operating systems. For example, Miller et al. (2016) asked participants to interpret different platforms’ representations of the same emoji: they described the “grinning face with smiling eyes” emoji from Google 😄 as “blissfully happy” while Apple’s implementation 🤗 conveyed “ready to fight”. This means that communication partners in close relationships, who often use many apps in parallel, may not be able to enact their inside jokes and intimate bonds in the same way across their app ecosystems.

Up to this point, the literature portrays mediated communication as a co-adaptive phenomenon: people adapt their expression to the capabilities and limitations of technology, and at the same time appropriate them to manage their self-presentation and relationships in more personal ways. However, most related work on appropriation looked at how users repurpose communication technology as given, rather than observing how they explicitly adapt it (i.e. customize it) to their needs.

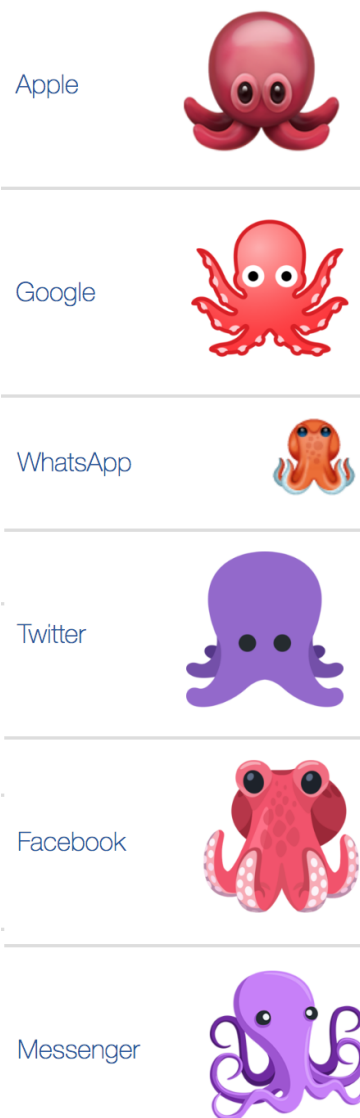


Figure 4.7: The octopus emoji in different apps, social media sites and operating systems. Source: <https://emojipedia.org/octopus/>

² In the weeks before the CHI’18 deadline, one of my colleagues was ranting on Slack about a problem with their paper. Mai Ciolfi replied: “There is no emojis for empathy, but here’s a flan 🍌”. Since then, we all use the “custard” emojis (which looks like an Argentinian flan) to express empathy with Slack reactions.

³ During a coffee break at the lab, Midas Nouwens and Phillip Tchernavskij were arguing about whether saying “octopuses” in English was correct, or if it should be the Latin-based “octopi” or the Greek “octopodes”. Since that day, the 🦑 emoji became a signifier of pedantry on our Slack, and we react with 🦑 to pedantic messages.

Customizing software

Previous research studied why people customize software. As seen in Chapter 2, triggers to customizing include noticing own repeating patterns, seeing an interesting customization by someone else, or retrofit changes from systems updates (Mackay, 1991). People perceive their phones are more usable when they customize their cases, lockscreen image, and order of installed apps, but “*not everyone makes the effort to adapt their smartphones to suit their individual needs, preferences and contexts*” (Tossell et al., 2012). Oulasvirta and Blom (2008) argue that, among other reasons, people customize technology to express emotion and identity in relation to others. Blom and Monk (2003) present a theory of customization of appearance in software, and offer a *checklist* for determining how likely it is that users will customize a system: for Instant Messaging services, a reasonable customization cost (i.e. effort), the social and emotional value for the user, and the availability of seasonal and media themes among the customization options increase the likelihood of users customizing.

However, the literature includes very few studies about how customizing mobile communication apps support users’ expression. Customization has been a widely studied topic spanning diverse domains, such as accessibility (e.g. Peissner and Edlin-White, 2013), software development (e.g. Findlater et al., 2008), multi-player games (e.g. Dyck et al., 2003; Sotamaa, 2010; Yee et al., 2011), word processors (e.g. Kahler, 2001; McGrenere et al., 2007; Page et al., 1996), and task efficiency improvement (e.g. Bunt et al., 2007; Findlater and McGrenere, 2004; Tabard et al., 2007). On the other hand, most research on customizing communication apps and social media sites focus on privacy settings (e.g. Stutzman et al., 2012), rather than on the myriad of customization options that may affect users’ expression and thus, help them manage the impression they transmit to others.

In summary, the literature mostly focused on how users repurpose and adapt to particular types of expressive media (e.g. emoticons, emoji, animated GIFs), the expressive media in a specific app (e.g. in WeChat, in Snapchat), as well as on how perceptions of expressive media vary across platforms (outside a particular context of use). The study that follows investigates how users explicitly adapt their communication apps to support their individual and relationship-specific expression, offering a detailed account of the type of customizations that users do across their entire ecosystem of communication apps.

4.3 Method: Interviews

We collected data in two stages: First, through an interview study with 12 participants and then, through a questionnaire with 17 respondents. Next, I describe the interview study.

Participants

Participants included 7 women, 4 men and 1 non-binary between the ages of 22 and 31 (mean = 27), all with occupational backgrounds related to software development or software design. At the moment of the study, they were living in Argentina (2), Finland (2), France (1), Germany (3), Switzerland (1) and the USA (2), and originally came from China, Argentina, France, Sweden, Russia, Germany, Canada, Hungary and the USA. The reported mobile operating systems included Android (8) and iOS (3), and the desktop operating systems included MacOS (10), Windows (5) and Linux (2). No compensation was offered for participating in the study.

Procedure

I recruited 12 participants via Facebook. The Facebook post asked what kind of customizations people did to their communication apps. I explained that a customization was “any change to an app that makes it work differently to how it worked when first installed”. I then invited the people that commented the Facebook post to participate in an interview. All interviews were conducted online, using Skype and Hangouts according to participants’ preferences.

Interviews lasted between 30 and 70 minutes. I asked participants about how they customized their apps (e.g. add custom stickers, change the background image of a conversation, install plugins, etc.), how they use such customizations with their closest contacts (usually best friends, family, and romantic partners) and how they communicate with those same contacts via other apps. I also explicitly asked about recent frustrations regarding app-exclusive customization options that they needed in other apps.

Data collection

All interviews were audio recorded. Optionally, participants sent us screenshots of their mobile communication apps showing configuration settings or conversations where they used a feature they reported. Demographic data was collected via an online questionnaire.

Analysis

I did a thematic analysis of the interviews, both through an inductive and deductive approach as presented by Braun and Clarke (2006). After extracting 291 interview excerpts and notes from the raw data, I did an inductive analysis of the data (open coding). Some of the resulting themes from this data-driven approach included: shared communication practices with close friends, customizing apps to reflect and reinforce relationship closeness, attempts to maintain a consistent expression style across apps, breakdowns in users' intended expression across their multiple apps, and related workarounds. I discussed the themes with the whole team. Similar to the analysis described by Vitale et al. (2018), we noticed that two themes were not always related to customizations but still offered interesting insights into how users perceived and dealt with app-exclusive features across their app ecosystem: breakdowns and workarounds related to users' expression across multiple apps.

Participants not only mentioned customizations that they did in one app but could not reproduce in another, but also reported frustrations about app-exclusive features that were not customizable. These breakdown stories did not talk about "bugs" or usability issues: they revealed unmet expectations about how an app should work based on the user's experience with another app, providing specific pain points to address with novel technology. Moreover, some participants also reported workarounds to their breakdowns, inspiring new mechanisms to mediate expression in app ecosystems. We agreed to broaden our initial scope, and continued studying how, beyond customizations, users leverage the expressive features of their app ecosystem to make their mediated communication more personal, as well as related breakdowns and workarounds. Rather than continuing with in-depth interviews about customization practices, we decided to collect more focused data via online questionnaires. Thus, we conducted a complementary study using *story questionnaires*, described below, to collect more stories around breakdowns and workarounds and enrich the cor-

responding themes.

4.4 Method: Story Questionnaires

I collected new breakdown stories via online questionnaires to get more respondents on very few questions. We conducted this complementary study to find more nuances in the type of breakdowns identified in the interviews, and to find different workarounds to those breakdowns that could further inspire opportunities for design.

While the interviews applied Mackay's variation of the critical incident technique (Flanagan, 1954; Mackay, 2002) to collect specific stories, I considered a different approach for the questionnaires. Even during interviews, it is often hard to obtain precise stories about lived experiences from participants. They tend to generalize their experiences in abstract complaints about technology or step-by-step descriptions about how they "usually" use their apps. Getting specific stories that narrate the actions, surprises, expectations and frustrations of a particular experience often requires a dialogue, probing from different angles. Questionnaires have fixed structures that do not afford dialogues with participants, and I struggled to find clear questions capable of collecting the kind of *stories* we wanted. Instead, I thought of mimicking a casual mini-dialogue with the participant: the questionnaire presents a story about a breakdown, hoping to remind the participant of a story of their own, and then asks the participant to tell their story. I call this method *story questionnaires*.

Story questionnaires prompt participants to write stories from their own experience by telling a story rather than by asking a question. Similar to how telling an anecdote to a group of friends triggers anecdotes from others, story questionnaires generate new stories from a stem story. While future studies could explore story questionnaires as a stand-alone data-collection method, in this first trial I use them to grow the stories from the interviews. Thus, the stem story is inspired in real breakdowns and workarounds.

Other methods used in psychology and medicine use stories as prompts, not to collect new stories, but rather to collect participants' opinions, values, and perceived social norms. For example, factorial surveys or *vignette analysis* are hypothetical stories or situations at-

tached to a rating task (Jasso and Opp, 1997; Rossi and Anderson, 1982). A vignette could present Amy, a public university student, and a situation: the government is cutting 5% of the budget assigned to public universities, and there is a protest against it. The rating task could ask whether the participant thinks whether Amy should participate in the protest or not. In this method, participants read a story but do not provide their own, as the main purpose is to collect their impressions rather than their own experiences on a subject. Moreover, the stories are constructed by the researcher to elicit opinions around particular situations, social norms and personal values. Another example is the *story completion method*, where participants have to complete a story based on a “stem sentence”, e.g. from (Clarke et al., 2017): “Jess is a 21 year old lesbian woman. She has recently met someone, and they have arranged to go on a date. Please write about what happened next.” In this case, participants write stories themselves, but they are made-up stories rather than lived experiences. Again, the goal is to collect data that reveals personal impressions and values around particular situations or characters, rather than collecting narratives of participants’ own experience around a topic. With story questionnaires, I present stories based on reported data from the interviews to collect new but related stories, with the goal of finding nuances around salient data as well as more diverse stories that can inspire richer design ideas.

Participants

We recruited 21 participants (8 male, 10 female, 1 gender-agnostic AFAB⁴, 2 preferred not to say) between the ages of 16 and 32 (mean = 24) via social media and Reddit’s Sample Size subreddit⁵. Participants lived in Argentina, Albania, Australia, Sweden, United Kingdom, Spain, Norway, USA, Austria, Israel, France, and Germany. Six participants were residing outside their country of origin. Occupations included systems engineer, business developer, data scientist, UX designer, researcher and unemployed. Most were students (14) ranging from high school to doctoral degree level.

⁴ AFAB: Assigned Female At Birth

⁵ <https://www.reddit.com/r/SampleSize/>

Procedure

After an inductive analysis of the interviews, we decided to enrich the themes about breakdowns and workarounds related to users’ expression across multiple apps. Based on stories of those themes, we created

a stem story for a questionnaire.

The stem story aimed at portraying a real breakdown and workaround from the interview data. We designed it so that it provided sufficient detail to help the participant empathize with the story, but stayed ambiguous enough to elicit diverse responses—similar to what Clarke et al. (2017) recommend for stem sentences for the story completion method.

Taylor loves dogs, so she added a lot of dog stickers to one of her messaging apps. Ben, one of Taylor's best friends, uses a different messaging app that does not support stickers. One day, Ben messaged Taylor to tell her he passed an important exam. Taylor wanted to reply with one of her favorite happy dog stickers, but she couldn't. She had to use a default smiling emoji instead, which felt less fun and less personal.

The stem story provided some context first: the sentence *"Taylor loves dogs, so she added a lot of dog stickers to one of her messaging apps"* presents a customization (adding dog stickers) and a motivation (Taylor loves dogs, it is part of her identity). the following context sentence states the source of the breakdown (Ben using a different app), and also mentions Taylor's relationship with Ben, hoping it prompts the participants to think of stories related to their own best friends. Then, the breakdown: *"One day, Ben messaged Taylor to tell her he passed an important exam. Taylor wanted to reply with one of her favorite happy dog stickers, but she couldn't."* And a (mildly successful) workaround, stating what she did, but also how she felt: *"She had to use a default smiling emoji instead, which felt less fun and less personal."* The stem story includes many elements that might resonate with the participant: a customization, a trait of their identity, a close relationship, a conversation where they could not express themselves as they wanted, the compromise they made and how that made them feel. We did not mention the name of specific apps to avoid reducing the story to "a problem with Whatsapp", which could constrain the diversity of the response stories.

We presented the stem story below a topic sentence that framed the kind of breakdown the story referred to:

Topic: problems with accessing the same custom stickers/emoji from different apps.

We then asked the participant to write their own story:

Tell us your story

Look at Taylor’s story above, and tell me the first story of your own experience that comes to mind. If it doesn’t remind you of anything, try to remember a story on the same general topic and tell me that. Try to be as specific as possible, mentioning the apps you were using, the emoji/stickers or other features you used (or tried to), your relationship with the person you were chatting with, etc.

Last, we invited the participant to complement the story with more details, feelings and other comments:

Any other comments? Let us know anything else you want to share about your story and how you felt. For example, if your story is about a problem: how frustrating was it? Did it happen more than once? If you found a workaround: are you proud of it?

Data collection

Apart from the story and additional comments, the online questionnaire asked for participants’ age, gender, nationality, country of residence and occupation.

Analysis

A second, deductive thematic analysis of all the data (from interviews and story questionnaires) categorized types of breakdowns and workarounds. We also complemented the initial themes with responses from the questionnaire.

4.5 Customizations to communication apps

We first introduce a summary of the types of reported customization options, and the apps where participants applied them. Then, we present four themes characterizing the role of customizations in mobile communication. Example quotes denoted with “I” refer to interview participants, while those denoted with “Q” refer to questionnaire respondents.

Table 4.1 lists the types of customizations reported in the study:

Customization	Reported apps	Description	Example
Add custom emoji	Slack, WeChat, HipChat	Import an image (static or animated) to be used as an emoji. The image may be authored by the user or not (as in the case of memes).	I also made an emoji of a thinking bear. Others started using that one instead of the original thinking face (I1).
Add custom stickers	Telegram, WeChat	Import an image to be used as a sticker. The image may be authored by the user or not (as in the case of memes).	My wife made some sweet stickers on Telegram from funny pics of my friends, including a ridiculous one on me. (Q4)
Add sticker pack	Messenger, Telegram, LINE	Install a pre-packaged set of stickers offered by an app. For this, users usually have to browse the available sticker packs for an app (e.g. as in Messenger), or copy a sticker pack from a contact (e.g. as in Telegram).	In Telegram somebody else created a sticker group of Pokemon, I found it and added it. (I5)
Change background of conversation	WeChat, Telegram	Set a background picture in the conversation with a contact or a group chat. Normally, only the user that made the customization sees the change, and the receiver is unaffected.	[I changed it] in a group chat with my best friends, and a chat with mom. Basically, all the pictures I choose belong to my favorite singer group. (I8)
Change background of the app	Whatsapp, WeChat	Set a common background picture for all conversations. Only the author of the customization sees the change.	I change the background on whatsapp just for myself. I changed it to a flower to look nicer and not too busy. (I4)
Change chat bubbles color	Messenger	Choose a background color for the chat bubbles of a conversation. All contacts in the conversation see the change and can change it again.	In Messenger I changed the conversation to green [bubbles] with my husband because he likes green. (I7)
Change emoji shortcut	Messenger	Change Messenger's blue "thumbs up" icon for an emoji. This button acts as an emoji shortcut, appearing next to the input field where users write their messages. Pressing the button sends the emoji as a big sticker. The longer the user presses, the bigger the sticker.	We both bike and we talk about bikes a lot, so I put a bike there [with a friend from college]. Or when you don't have anything to say or it's an obvious answer, you just send a bike. (I4)
Change conversation title	Whatsapp, Messenger, WeChat, Telegram, Instagram	Set a title for a conversation (rather than showing the names of the contacts involved), usually in group chats. All the contacts in the conversation see the changed title, and can change it again as well. The title may contain emojis.	With my group of friends I have five different groups: There's "misfit musings", "misfit mumblings", "misfit memories", "misfit meals" and "misfit melodies". It's the same people in these groups but we just talk so much that it's too much to put everything that we want to share with each other in one. (I12)
Change group's profile picture	Whatsapp	Set a group profile picture. All the contacts in the conversation see the changed picture, and can change it again as well.	If we take a family picture that we like we put that one [as a group profile picture on Whatsapp]. (I4)

Customization	Reported apps	Description	Example
Change notification settings	Whatsapp, Telegram	Set notification preferences for an app, or a contact from an app, such as choosing a particular ringtone, muting all notifications, allowing the notification to show up on the lock screen, or hiding the preview of the received message.	We use Telegram for some work groups, and I have those groups with different notification settings, and I mute everything else. All the alerts, and previews of the messages, and the vibration of the phone, it's all mute. So I can pay attention to the important stuff. (I5)
Change other's nickname	WeChat, Messenger	Change the displayed name of another person in a conversation or in an app. Particularly in Messenger, the changed nickname in a conversation is visible to all the contacts in the conversation.	We went for a shrimp cruise, so we changed the emoji to a shrimp, but then someone changed everyone's nickname to a shrimp emoji. There were shrimps everywhere! (I11)
Change other's profile picture	Whatsapp	Set a profile picture for a contact. Only the user that customized sees the change.	I hate it when my contacts don't have pictures, so if I talk to them frequently I change them myself. My sister for example. I think now she overwrote it. But when she calls, mine pops up. it's nice to have 2 pictures. (I4)
Change privacy settings	WeChat	Control what status information or content is publicly available or private.	WeChat has something like... "people next to you". If you open that, a lot of people that are close will be saying hi, so I turned that OFF. (I2)
Add chatbot	Slack, Telegram	Install a chatbot which adds functionality via a conversational interface.	We installed <i>Donnuts</i> : If someone reports a bug, you get notified. (I11)
Change chatbot settings	Slack	Customize a simple chatbot to recognize words in conversations and react with phrases, images, videos or emojis. This customization refers specifically to the Slackbot (Slack's integrated bot)	When you write a word it sends you a picture. For example, you write "dolars" and it sends you a GIF of a dancing person and money falling. (I6)
Create custom chatbot	Slack, Telegram	Program a new chatbot and install it.	Somebody created a bot that tells you when is the next company trip. (I1)
Change status message	Slack	Change the text that describes the user's current status, next to their nickname. The status message may contain emojis.	We use [emoji] flags in our status message to show in which country we're currently in. (I10)
Install custom keyboard	Tenor Keyboard, Bitmoji Keyboard	Install a soft keyboard from the mobile applications store. These keyboards often provide emoji, GIFs (e.g. Tenor) and stickers (e.g. Bitmoji) that users can access from all apps.	I installed a GIF keyboard before Messenger added Giphy [a service to search for GIFs] to respond with GIFs to a friend who used GIFs a lot. (I12)
Install plugin	WeChat, Messenger	Extend the functionality of an app with a plugin, e.g. a feature for creating polls or a game to play within a conversation.	In WeChat there's a plugin for sending a voting system, a bit like a doodle in a group. You can just add where you want to go, choose single or multiple vote, and you share it to the group. (I2)
Set dark screen mode	Telegram	Change the look and feel of an app to a dark theme.	I'm using the dark theme [on Telegram]. I'm using a dark theme overall in the phone, takes less battery and it's less aggressive for the eyes. (I9)
Set favorite contacts	Telegram, WeChat	Create a shortcut to frequent contacts. The user's favorite contacts appear first in the list of recent conversations.	I have, for example, my boyfriend and my parents on the top. So then, it's easier to talk to them all the time. (I2)

Table 4.1: Reported types of customizations.

Customizing for expressing one's identity

Some participants reported customizations in several apps to consistently express an aspect of their identity. For example, the “Squirtle with sunglasses” (Figure 4.8) was I5’s signature emoji/sticker. She used it in all her apps instead of the regular sunglasses emoji, for which she added a Pokémon sticker pack in Telegram, created a custom emoji in Slack to use it in reactions, and kept an image at hand on her phone’s media to send it in Whatsapp. Similarly, I6—a devoted fan of Britney Spears—liked expressing herself with funny Britney stickers, images and GIFs, which she collected over time to use in both Telegram and Whatsapp.

We also found examples of participants customizing their apps to tweak their self presentation to different social circles, expressing select aspects of their identity according to the audience that saw their customization. For example, I11 was part of two Slack teams and decorated her user name with different emoji in each:

I have like three big balloons in my hand, so I have the balloon emoji next to my name [on the Slack team for work]. My profile picture [on the Slack team for work] has balloons, so I have the emoji of a balloon. I’m also in another group [Slack team], it’s called “Android study group”, there’s a lot of Android developers world wide. There, I uploaded a *Nyan cat* [i.e. created a *Nyan cat* custom emoji, Figure 4.9] because I wanted it next to my name and there was no Nyan cat emoji. I like cats and I like Nyan cats, and it’s very me. I thought of adding it to my work [Slack] team as well, but they like that I have the balloon emoji, because I’m somehow the balloon person on Slack. (I11)

Some also talked about how they perceived others’ identities based on their customizations. For example, I3 associated custom animated emojis of “party parrots” (Figure 4.10) with one of her colleagues: “*He really likes birds and sort of started that culture [of the party parrots emojis]. Scrolling through... every parrot is from [him], who added them in the first place*”. I11 mentioned a colleague who created his own custom emoji collection featuring his own face: “*We have this designer, [Bob], that made emojis with his own face: “[Bob] approves”, “[Bob] is sad”*”

Last, some customizations helped users express their identities but outside of any communication context. In such cases, customizing



Figure 4.8: I5’s signature sticker/emoji (the Pokémon “Squirtle” wearing sunglasses, a popular Internet meme), which she used in all of her apps.



Figure 4.9: I11’s custom *Nyan cat* emoji, which she described as “*it’s very me*”.



Figure 4.10: A party parrot custom emoji. *The cult of the party parrot* is a collection of small animated emoji tailored for Slack (<https://cultofthepartyparrot.com/>), inspired in a popular Internet meme featuring Sirocco, one of the few remaining kakapo in the world.

was a way of making the communication app itself a more personal tool. For example, I8 changed the background of several conversations with close friends and family on WeChat with pictures of members of her favorite *boy band*.

In summary, users interested in expressing their identities via their communication apps used customizations that affected the appearance of their profile (e.g. adding custom emoji next to their names), the expressive media available to send to others (e.g. adding custom emoji for Slack reactions, adding sticker packs) and the appearance of the communication app itself (e.g. changing the background).

Customizing for expressing intimate bonds with others

Some of the reported customizations were used only in conversations within close relationships, reflecting the higher intimacy that participants had with those contacts compared to others. For example, I9 installed a sticker pack in Messenger featuring a deer, but only used them with close friends; for the rest, he uses some “*yellow face stickers*”. I4 also mentioned using *group Bitmoji stickers* “*with close people only*”: “*if I use it it’s a different level of connection. If you use a sticker with both [sender and receiver] it feels close*”. Customizations that reflect the intimacy of a relationship may help increase feelings of connectedness, especially when they reflect intimate knowledge that the sender has of the receiver:

I changed the conversation to green [in Messenger] with my husband because he likes green. I also changed the thumbs up for a kiss. I open the conversation with him I see the emoji with a kiss and... everytime I see it, it makes me feel good, makes me feel connected to him. (I8)

Customizations sometimes helped materializing inside jokes and special bonds that participants had with others. In such cases, the customization only made sense for very particular contacts. For example, I2 said that before graduating, she took a lot of embarrassing pictures with her classmates, which they turned into custom emoji in WeChat: “*For one or two weeks, our group chat was everybody’s ugly face. I still have some of those emojis, but I’d use them specifically in the group of my bachelor friends*”. I12 changed his Messenger emoji shortcut for a friend, reflecting a make-believe conversation style unique to them: “*With my friend [Carl], the emoji for our conversation is this [angry face] and we send it as big as we can where you hold it down. So we use it just use it it’s more like we just pretend to be angry about things and our conversation revolves around*

that a lot". I₃ described how, together with a close friend, they customize each other's nicknames to reflect their inside jokes, rendering their mediated communication as more intimate:

With my good friend [M]— this one is actually a running joke between me and [M], we changed each other's facebook names often. So I'm *Costco artisanal bread* because one day I was buying this bread and it was really good, so I sent him picture "Look at this bread! It's... artisanal! But it's from Costco...! And it's so good". I changed his to *innovator* because he always like... makes this very stupid jokes, like "I'm so innovative! Disruption!" and stuff. So I think nicknames above anything is a sign of higher intimacy, cuz it deals directly with the person's identity. (I₃)

Others customized their apps to highlight a few special relationships from the rest of their contacts. For example, I₂ *pinned* a few of her conversations for easy access: "[WeChat] has something like "always set on the top". So I have, for example, my boyfriend and my parents on the top. So then, it's easier to talk to them all the time." While I₂'s example responds to a rather practical purpose, others customized some conversations simply to express which contacts were the most special to them. For I₃, customizing Messenger's emoji shortcut or chat bubbles color was a means to humanize the conversations with a select group of people, contrasting the branded, generic aesthetics of the app: "Here are my actual friends, people I talk to on a regular basis. Then... because you can do that [changing the chat bubbles color] it like... creates a differentiation from the default blue, the default like thing, which is also very Facebook." I₉ also mentioned changing Messenger's chat bubbles color because he is "tired of the standard blue" and "it makes it more personal". As close relationships often span across many apps, users may repeat the same customization for emphasizing some contacts over others in more than one app. When describing how she changed group conversation titles, I₁₁ explained: "The same group of best friends I have on messenger I have it on Instagram, which I also renamed to the same thing". I₁₁ also described how Whatsapp's custom notifications allowed her to highlight messages from people she prioritized:

The people I talk to on WhatsApp it's 4. In WhatsApp you can change notification settings, I wish I had that on Messenger. When I was an intern at [company] we had a personal group in WhatsApp I set the notification light to blue, then I knew it was somebody on that group that was trying to reach me. If it's someone you're dating... you want to know how fast it needs replying. How much of my attention I should bring to it.

In summary, users customized their apps to express intimate bonds

with others by adding new expressive media (e.g. sticker packs) dedicated to close relationships, changing shared interface elements (e.g. changing nicknames in a conversation, changing Messenger’s emoji shortcut or chat bubbles color), or differentiating some contacts from others in their private app settings (e.g. setting favorite contacts).

Customizing to reflect local and organizational culture

Particularly on Slack, customizations helped reflecting participants’ local and organizational cultures. For example, I6 lived in a country where soccer-related metaphors and jokes are very common. She added a custom emoji of a “red card” to Slack: “*I have a reaction for slack, a red card. It’s a GIF of a referee showing the red card. I use it when the guys are “crossing the line”.*” I3 used emojis on Slack to evoke the rivalry between two universities from the United States:

I put the axe emoji by my name [i.e. on the status message] when I moved to Stanford, because Stanford’s mascot is a tree. As a whole, I really hate Stanford culture, it’s very much Silicon Valley. And Berkeley is all about public school, free speech... So I did that to signify “ohh I may be at Stanford now but I’m always a bear [Berkeley mascot] at heart”. (I3)

We also found that custom emoji on Slack played an important role in reflecting and developing the organizational culture among peers:

One of the GIFs I made was a sheep. Its head would disappear, and then it’d show the head of a bear. This comes from a popular phrase in Finnish culture, “a wolf in sheep’s clothes”, and the bear is like the mascot of the local cheap beer, which is the beer that we usually had at the office. It was surprising to see that other employees loved it so much that they used it to react to negative news, surprises, and a lot of things. It was very good for the social culture inside the office. (I1)

Customizing the *Slackbot*—Slack’s default chatbot that reacts to a list of customizable words and phrases—also served not only to reflect culture, but to influence it. For example, I9 (who works with an international team) explained: “*Our official language is English, and people tend to forget about that, they added something in the bot so that if you type !english and then the bot will just link to the Youtube video in the conversation, it’s a quote from Pulp Fiction*” (see Figure 4.11). I11 and some colleagues planned a customization to the Slackbot for encouraging a more gender-inclusive language in the office, which triggered a debate and brought awareness to the issue:

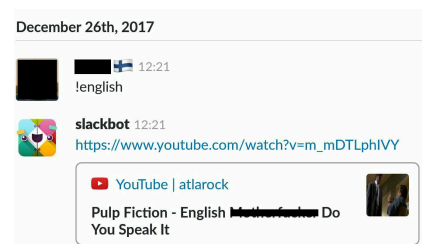


Figure 4.11: Slackbot reacting with a video to the word “!english”. Anonymized Screenshot sent by I9.

There was this conversation about “should you call people *guys*? Or is that non-inclusive?” They discussed a lot. Should we add this bot to say “guys is not gender neutral” or “you should just say people”? And then people who hadn’t been at that conversation got a bit strange when it got triggered to them. So that started a conversation, like... should we have this for everybody or not, because not everyone feels excluded if you say *guys*. But others were like “oh! I never thought about that, thanks so much for pointing it out”. I think it listened to either the word “guys” or like “hey guys”. And then it posted a message like “Oh, guys it’s not gender neutral, maybe try another word”.

We believe that custom emojis and Slackbot on Slack help build an organizational culture not just because Slack is a common choice for internal communication at workplaces, but because the new emojis and Slackbot triggers are available to all Slack team members and not just the user that created them.

New communication habits stemming from customizations

At first sight, users customize to adapt apps to their intended expression style. Nevertheless, we observed that some customizations also inspire new ways of communicating or new bonds within close relationships:

After a party we drank a lot, everybody was hung-overed and we wanted to share pictures. We called this group *detoxication*. Everytime we see something about detoxication we send pictures to each other. We changed the thumbs up for a watermelon [in Messenger], because it’s healthy. Just a joke⁶. (I7)

I changed it [Messenger’s emoji shortcut] for my boyfriend when I was angry at him, I changed it from the love sign back to the like. When we were good I changed it again. Once or twice he changed it, because he was angry, and then changed it back. We just use it as a communication of our feelings, but we never use the button, either *like* or *love*, it just communicates our feelings. (I10)

⁶Curiously, my friend Mai told me a similar story about a group of friends of her. They changed Messenger’s emoji shortcut to a watermelon, which became a symbol of their friendship. For her birthday, one of her friends gave the whole group a keyring of a watermelon that looked similar to the emoji.

Messenger offers many customizations that affect both sender and receiver, such as the mentioned emoji shortcut, chat bubbles color and nicknames. Such *shared* customizations led some participants to develop new codes of conduct around their use. For example, I10 liked having different colors in her Messenger conversations, but made an exception with her mother: “My mother changed it for me [to red, same color for I10’s boyfriend]. I know she wanted to express that she loves me, so I didn’t change it because that was her expression.”. If a customization

does not please one of the affected contacts, they may negotiate how to change it: *“With the family group [on Whatsapp], every time we take a family picture we change it. One time it was like “no! but that’s not a nice picture, why don’t you choose this one?” so my sister changed it.”* (I4).

In summary, while users may customize to express their intimate bonds with others, customizations may also generate new communication habits and bonds over time.

4.6 Breakdowns

We use the term “breakdown” for barriers to users’ mediated expression and unmet expectations about the features an app should support.

Frustrations around app-exclusive functionality

The most reported type of breakdown referred to functionality that users expected to find in one app, but was only available in others. Most often, they reported functionality that provided means of expression they had incorporated in their communication habits. The apps that lacked that functionality prevented them from expressing themselves as they would have preferred.

Many reported breakdowns revolved around *reactions*. Participants that got used to *reacting* to messages in Slack or Messenger often missed doing it in other apps. For example, I1 had created many custom emojis for Slack in his previous work, and now uses Flowdock in his new job, another chat app for teams: *“Flowdock allows you to add your own emoji, but it doesn’t really encourage me to do it. It doesn’t have reactions like Slack, so emojis only appear in the line of your message”*.

Some also complained about apps that lacked support for *stickers*. For example, I6 repeatedly tried convincing her colleagues to move from Whatsapp to Telegram so she could send them stickers, which are the feature she likes the most about Telegram. I9, who likes using *deer* stickers with close friends in Messenger, clarified: *“I would avoid using Messenger but it’s the only one that uses stickers”*.

Echoing findings from the previous chapter, users may have an ideal *communication place* (Nouwens et al., 2017) for their conversations with someone, but the functionality they need for a particular situation is in a different app. For example, many reported *polls* as functionality that they frequently need, but only few apps support. I2 said she needed WeChat's *poll* function in her Whatsapp conversations. I5 used both Telegram and Slack to communicate with her colleagues, but still complained about having *polls* in only one:

We're a bunch of nerds, so when we found the *poll bot* [on Telegram] and learned how to use it, we used it a lot, it's really easy to settle on something. A couple of times we had to decide something on Slack but didn't have polls. (...) We prefer to stay in Slack and just list the options and say "I like this, I like that". It's a bit caotic, but I don't want to go to another application outside of where I am. The context switch is a waste of time, and when we have a situation where we have to vote, it's something we want to settle quick. (I5)

Other breakdowns referred to practical functionality that was convenient from an individual user-experience perspective, but had nothing to do with communication. Examples of such functionality included searching through conversations, playing videos directly from a conversation, previewing URLs in the conversation before opening them, and changing the visual theme of an app. For example, I8 complained about Whatsapp lacking a "dark theme" setting, which she uses on Telegram: *"For me, it's vital. I have sight issues. I figured, if Telegram has it, Whatsapp will have it"*. While such features do not mediate communication or support expression per se, we believe they may influence users' channel choice when starting a conversation.

Frustrations around app-exclusive expressive media

Many apps (such as Telegram and Messenger) allow users to collect stickers and send them in their conversations, but participants regretted that the stickers collected in one app could not be used in others. Users that enjoy using stickers may not use *any* sticker, only those they identify with: *"I use stickers on Telegram but not on Messenger. I don't know who creates the stickers for Messenger but I don't like how they look. On Messenger they're like smileys from cartoons. On Telegram it's different, for example Angela Merkel or Putin"* (I7). For some, stickers were very personal collections that reflected who they are and what they like, and having those collections *locked* in only one of their apps restricted their expression. For example, I2 loves shiba dogs and

wished she could use her stickers from one app in the conversations of another:

I have a lot of doggy emojis in WeChat. There's tons of self-made emojis in WeChat so I saved a lot of them. And I'm so used to them. Sometimes when I chat on Whatsapp or Messenger and I talk with friends, and I know they also like dogs, I'm like "Oh, I wish I could just... like, switch apps" (I2).

Users may also find it frustrating when they cannot find an emoji they frequently use in other apps: *"On Messenger there's no good emoji for 😊 which is one of my favorite emojis. The closest one is 😊 which is not exactly the same. I cannot give one conversation example, because it happens every single time, for me to search again, thinking it may be there, and then it's not"* (Q8). Similar breakdowns occur when different apps implement emojis with different pictograms (Wiseman and Gould, 2018). While all emoji implementations intend to keep the same semantics, users may prefer expressing themselves via a particular version of an emoji, usually available in only one of their apps. I12 and a friend shared an inside joke around an emoji, which was not as fun when the emoji looked different:

With my friend [Bob], the emoji for our [Messenger] conversation is this: 🤨, and we send it as big as we can⁷. It's more like we just pretend to be angry about things and our conversation revolves around that a lot. (...) I don't use it much in Instagram because it's that faded yellow to red [🤨] instead of the foam red face. I want it to be red. (I12)

Some apps allow users to add any custom media from external sources, which ironically turn into yet another version of app-exclusive expressive media. For example, Q4 reported that his wife designed a set of stickers herself, which he loves and wants to use in all his apps, but can only import them to Telegram.

In summary, even when two apps support the same *type* of expressive media (e.g. stickers, emoji), differences in the particular expressive media available in each app (e.g. different sticker packs) or in their aesthetics (e.g. emojis in Android vs. iOS) can lead to breakdowns in users' expression.

⁷ As explained in Table 4.1, Messenger lets users set a default emoji for their conversation instead of their thumbs up button. When pressing this button, Messenger sends the emoji as a sticker. The longer they press the emoji, the bigger the sticker will appear in the conversation

Preferences over different flavours of the same functionality

Participants showed preferences over differences in the interaction details of “the same” functionality between apps. Even when two apps offer functionality that serves the same purpose (e.g. accessing an emoji library), users must learn how that functionality works in each. Some participants complained about the particular “flavour” of a feature in one of their apps. For example, I2 reported reordering her custom emoji and stickers in WeChat so she could find her favorites easily when opening the emoji menu. Later, she explained why she finds it frustrating to use emoji in HipChat:

You call the emoji by typing it. You type like a bracket and then... for example, smile, and a list of emojis shows up. I don't like this way, it's too difficult for me. For example, sometimes I try to find a "thank you" emoji, but I'm not sure if there is an emoji in the library, so I just try to type like bracket and then "thank" and if nothing comes up I try something else... and it's very frustrating.⁸ (I2)

To some users, such differences may be enough to refuse using some functionality in one of their apps. For example, I4 talks to one of her best friends via Snapchat and Whatsapp. She explained that they both loved using Bitmoji in Snapchat, but Whatsapp failed to provide the same experience: *“I used it [Bitmoji] on Whatsapp when I first installed it, but it's sent as an image and then stays in your media, so that's annoying. Then you look at your pictures and it's all full of stickers⁹. That's one of the main reasons I didn't use it anymore [in Whatsapp]”*.

These examples suggest that even when functionality of two apps supports users' favorite means of expression, users may prefer particular interaction or implementation of one over the other, sometimes even discouraging the use of the functionality in one of the apps altogether.

Asymmetric representations of expressive media

Participants not only reported breakdowns about differences across their apps, but also about differences between their apps and their closest communication partners'. Some apps use the operating system's implementation of emoji rather than their own, so an iPhone user chatting with an Android user may see different representations for the emojis they send to each other. I1 explained that after an update, his emojis changed but his girlfriend's did not. He did not identify

⁸ Personally, I have the opposite problem. I am used to typing shortcuts for emoji in Slack, so I easily recall the shortcuts of my favorite emoji (e.g. :heart_eyes: for 😍). I intuitively type :heart_eyes: on Messenger but nothing appears, because Messenger only allows emoji entry from their emoji menu. So I am forced to browse through all the emojis, despite I know which one I want.

⁹ Before GBoard integrated Bitmoji, the later had its own custom keyboard to send stickers from any app. As Whatsapp does not support stickers, trying to send a sticker from a custom keyboard results in simply sending an picture, which is stored in the user's media library together with other exchanged photos and videos.

with these new emojis and felt that they expressed different emotions to the previous ones:

I would change the emoji back to the old ones if I could, but I wouldn't use any custom ones that only I can see, since then I wouldn't be sharing the experience with her - it would be different. So now I still send her some of the affection related emojis like in the old days sometimes but it's mainly for her benefit. (I1)

Some participants could not see some of the emoji they sent to each other in their conversations: *"Trying to text my boyfriend when he had a Samsung and I had an iPhone was annoying because you can only send certain emojis to each other"* (Q10). The same happened with stickers, when the sender's app supported stickers but the receiver's did not:

A friend of mine sometimes sends me stickers via text messages. I think I's need to activate my iMessage in order to properly receive them, because as of right now they are just lines of code. I don't even know what those stickers should look like, but it's a shame I can't see them because they are part of our conversation that I am missing out on. (Q16)

This type of breakdown emphasizes that the aesthetics of emoji matters for supporting personal expression, suggesting that emoji representations should be customizable rather system-imposed. Moreover, the lack of symmetric support of expressive media between sender and receiver occludes non-verbal cues that make the conversation more expressive.

4.7 Workarounds

Participants reported compromises and clever tricks (Jalal, 2016) to overcome the limitations that some apps imposed over their expression.

Sending similar expressive media to the ideal one

Some participants found a compromise between the ideal expressive media for a conversation and the available options in the app.

I use WhatsApp. Every time I speak with my bf in real time, and he is

about to do something important, I tell him "I'll keep my fingers crossed for you". When we have such conversations online, I can't use an emoji for that (there is no crossing fingers emoji), so I send him the Vulcan Salute instead. It has become our own personal joke. I wished I could send crossing finger emojis though. (Q2)

Such compromises show how users adapt to the expressive media and functionality available in one app when they originally wanted to use what another app offered. I6 had a big collection of Britney Spears stickers in Telegram and she wanted to use them with her colleagues on Whatsapp: *"I tell my friends "C'mon! Let's move to Telegram so I can send you stickers... really!". So I send them GIFs"* (I6). Using different media to the ideal one may require additional efforts. For example, apps that support stickers allow users to create their own easy-to-browse libraries, but few allow them to create their own libraries of GIFs: *"I have a repository of GIFs [in my phone's media library] that I always use because I don't want to be searching."* (I6).

When two friends share inside jokes in a specific app, other apps may require a bit of imagination for evoking the same joke with alternative media or functionality:

I added an airhorn emoji [to Slack]. This is because [Bob, a colleague] and I are very good friends too, and talk half on Messenger, half on Slack. Sometimes in conversations with [Bob] I airhorn-react to the stupid *memes* he says on Slack, but I also remember sending him an image of an airhorn or an mp3 file of the airhorn sound on Messenger. (I3)

Most apps interpret ASCII emoticons as a shortcut for emoji, assuming users prefer the latter, e.g. 😊, over the rather abstract representation of an ASCII emoticon, e.g. :) . Such preferences may be personal, and users may attempt to trick an app to avoid such replacements, typing variations of their intended emoticons:

Sometimes the text-based emoticons seem closer to what I want to say than their default replacements. (...) Sometimes I put a space between colon and parenthesis in order to send a plaintext emoticon. :) (Q14).

In summary, when an app imposes limitations on how users want to express themselves, they find compromises in an effort to convey their message anyhow, adapting to the available expressive media or functionality.

Saving expressive media as a file to send it from any app

Many participants reported transforming stickers from an app to images, to then access them from other apps via their phone's media library. For example, I9 said he took a screenshot of a sticker in Messenger for sending it on Whatsapp (as an image), but only *"once or twice because it takes too much work"*. Q1 went to a great length to use his Telegram stickers on Whatsapp, using a "sticker downloader" bot (Figure 4.12):

It is common with Telegram and Whatsapp. If I'm in the need, I use a Telegram bot called "sticker downloader" which converts stickers to images and lets me send the sticker via Whatsapp. But Whatsapp doesn't accept transparencies and adjusts the image to a standard size, so it isn't quite the same. For example, one time the sticker I wanted to share was a sticker that someone had put in a sticker collection specific to my University, which had inside jokes fun to share (Q1)

Generating rich content in an app to send it via another

In cases where participants needed to access an app's exclusive functionality while having a conversation in a different app, some generated the content for a message in one app, saved it as a file, and then sent it via the other. For example, for editing videos with effects only available on Snapchat or Instagram:

I like to make fun (though that's arguable) personalized messages on Snapchat, as a reaction to certain events of my friends, or in situations where I want to share an event or activity that I do. The problem I have is that many of my friends don't use Snapchat, so I have to export the video to my camera roll and send it over via the app that my friends are using. It's a nice workaround that ultimately solves my problem, but it's not as straightforward so I tend to send less messages than I would have if I had some of my friends on Snapchat. (Q6)

I12 applied the same workaround, not for compensating a missing functionality in one app, but to access his preferred "flavour" of a functionality: *"Rewind"¹⁰ works better on Instagram, so sometimes I'll do that, but usually save it and put it on Snapchat*".

As mentioned before, Bitmoji stickers are available in any app, but Snapchat better integrates them with other functionality. I12 used Bitmoji on Snapchat to decorate a picture he wanted to send on Instagram: *"Snapchat seems to integrate them [Bitmoji stickers] really well. So*

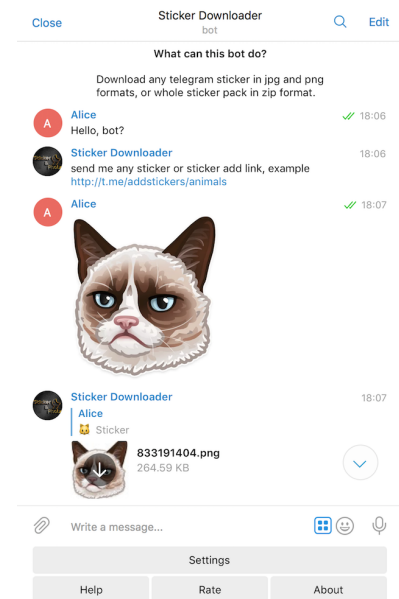


Figure 4.12: "Stickers downloader" bot for Telegram (sti, 2016). The user sends a sticker to the conversation with the bot, and the bot replies with a downloadable file of the sticker. This bot reported by Q1 has 12,263 reviews averaging 4 out of 5 stars on @StoreBot (sto, 2016), another Telegram bot for finding and rating bots. This suggests that Q1's needs and workaround are not an isolated case.

¹⁰ Rewind is a video effect for making videos that play backwards. The user records a video, and the rewind effect produces a new video that plays backwards.

because the other apps don't really let me put a bitmoji on a picture and I want that, if I'm sending it to somebody through Instagram who doesn't use Snapchat, then I'll use Snapchat, put the bitmoji on the picture, save that, then upload it to the other app". He applied this workaround for a friend who no longer uses Snapchat, but explained that he can still access her bitmojis on "her dead account", which he uses for decorating pictures with group bitmojis featuring both of their avatars together.

Generating content one app and sending it via another may not only come from expression needs. I₁ explained how he sometimes uses Whatsapp as an image compression tool when he wants to send a picture to people in other apps and consume little data from his subscription:

Whenever I have a picture that I'd like to share with multiple people, I would send it those who use WhatsApp first, and then send the compressed version that Whatsapp creates via Slack or Flowdock. (...) I was on the way to a sports competition with the company team on a beautiful Saturday morning, and I wanted to show the contrast created by walking along a trash-filled "party" street. So I sent it to my girlfriend [on Whatsapp] first before sending it to the company channel [on Slack], just to get the compression. (I₁)¹¹

¹¹ I₁ was not the only participant that sent content to his girlfriend first to then use it for other purposes. I₉ reported sending media to his girlfriend as a sharing mechanism between his phone and his computer, and he lets her know by writing "using you".

4.8 Discussion

Communication habits acquired in one app transfer to others

Our data suggest that users acquire expression habits from each app, which then try transferring to others, e.g. Q₈ complaining about the absence of her favorite emoji on Messenger, I₃ sending a picture of an airhorn via Messenger to mimic a Slack reaction, or Q₆ creating fun videos on Snapchat for sending them on Instagram. Each app's expressive media and functionality shape users' expression style, which they may try to reproduce in other apps. Thus, when an app introduces a new feature, it shapes not only how communication happens there, but in other apps of users' ecosystems as well.

The reported breakdown stories illustrate how customization options and other app-exclusive features nurture communication habits and shape an integral, personal communication style, which then no app supports fully. For example, I₁₂ needed Messenger's full-red an-

gry emoji in Instagram, Instagram's video effects in Snapchat, and Snapchat's custom stickers in Instagram. When one app introduces an exclusive feature, they may *enrich* users' expression. But once users incorporate that feature in their communication habits, they may feel that other apps *restrict* their expression. Thus, "power users" that communicate via multiple apps on a daily basis may be always missing a means of expression, no matter what app they use, especially when they invested effort in customizing them.

App-exclusive features prevent expressing a consistent identity across apps

Some participants customized their apps to make their mediated communication more personal and express their identities. Nevertheless, customization options are often app-exclusive: they can make the communication via each of their apps more personal, but always in different ways. Some stories suggest a common need to express one's identity in a consistent way across different apps. These observations support Sugiyama's 2015 discussion about using emoji and stickers to express one's "aesthetic self". We expand Sugiyama's findings by showing how some users struggle to maintain a *consistent* aesthetic self across different apps (e.g. Q8 being unable to use one of her favorite emoji on Messenger), and how others settle for adapting their self-presentation to what each app allows (e.g. I6 using emoji, stickers and images of the Squirtle Pok'emon in Slack, Telegram and Whatsapp respectively, because not all support custom emojis and stickers).

Taber and Whittaker (2018) argue that "*personality depends on the medium*" and discuss how the social media affordances of Snapchat and Facebook lead users to be more or less open about the content they share, respectively. They suggest that Facebook's persistent traits *encourage* users to present "*filtered versions of their offline selves*", while Snapchat's ephemerality allows them to be more authentic. While users intentionally manage how open or closed they want to be in different platforms or for different audiences (Shklovski et al., 2015; Walther, 2007; Zhao et al., 2016), we show that app-exclusive features *impose* users to present filtered versions of their *online* selves when they use many apps on a daily basis. While most research on communication app ecosystems focused on how users "*keep multiple dimensions of one's identity (and the audiences for each) separate from one another*" (Zhao et al., 2016), our results show how, at times, users invest substantial effort to express their identities consistently across different apps.

In summary, we show users' efforts in maintaining a coherent identity across apps—a *signature* in their use of expressive media—and how the current model of isolated apps with exclusive customization options and functionality often fails to support.

App-exclusive features prevent expressing intimate bonds across apps

Some participants customized their apps to express intimate bonds with others, especially within close relationships. Some also repurposed expressive media with their own meanings around inside jokes or shared activities. Nevertheless, these intimate bonds were often hard to express across users' app ecosystems, as they depended on app-exclusive features.

Miller et al. (2016) discussed how different representations of emoji affect the perceived meaning across apps and operating systems. We show how users may avoid using an emoji when they anticipate that different aesthetics might convey a different meaning than what should express an intimate bond with the receiver, e.g. I12 using the angry face emoji less in Instagram than in Messenger, because it looked different from the version their inner joke was based on. App-exclusive expressive media generate intimate bonds that are coupled with specific apps, despite close relationships often communicate via multiple apps. Thus, similar to how users restrict their identity expression according to the app they use, close relationships are also forced to filter the intimate, playful communication codes they develop in their app ecosystems as they move between apps.

Beyond app-exclusive expressive media, exclusive functionality also imposes limitations in how users express intimate bonds with others, e.g. P3 sending an image of an airhorn rather than *reacting* with an airhorn, and I12 using his friend's dead Snapchat account for composing a picture with their Bitmoji together. While Messenger does not support custom reactions (I3) and Instagram does not support adding Bitmoji to pictures (P12), participants felt it was worth the effort of mimicking and compensating for the missing functionality just to express the special bond they have with their friends.

While previous work showed how users' relationship maintenance practices differ across apps and social media platforms, our data illustrates how users try to transfer the communication patterns and bonds they developed in one app to others.

Apps alternate roles between communication channel and content-generating tool

Some participants overcame the expressive limitations of certain apps by first generating content in one app, and then sending it elsewhere. Our results suggest that users not only perceive the functionality of an app as part of what defines its *placeness* (Nouwens et al., 2017), but also as content-generating tools they can access while having a conversation somewhere else. Sleeper et al. (2016) show how users may *chain* different apps for cross-posting content to composite audiences, e.g. using Instagram for video and photo editing, posting the content on Instagram and then cross-posting it to Facebook. In stories such as P12's or Q6's the videos generated on Instagram or Snapchat were never shared via their original apps: rather than communication channels, Instagram and Snapchat played the role of video editing tools. This suggests that users perceive an app's *expressive functionality* as a separate dimension from its reachable audience (Zhao et al., 2016) or idiosyncratic purpose (Cramer and Jacobs, 2015; Nouwens et al., 2017; Taber and Whittaker, 2018). For example, Bayer et al. (2016) reported that participants perceived Snapchat as a lightweight channel for sharing spontaneous experiences (purpose) with trusted ties (audience); we show how Snapchat may also be seen as a tool for generating short, funny videos for sharing elsewhere.

In the previous chapter, we showed other types of breakdowns in communication app ecosystems: some participants compromised their *communication places* by adding "an outsider" to an app meant for other people, only to access exclusive functionality. Here, we show the other side of the coin: rather than bringing the contact to the functionality, participants brought the (result of the) functionality to the contact. Such workarounds may be rooted in attempts to preserve users' communication places.

In summary, the same app a user treats as a communication channel for a conversation may be treated as a content-generating tool for another conversation in a different app. Savvy users consider the capabilities of their entire app ecosystem when choosing how to express themselves, and find workarounds to access the expressive capabilities of one app while they communicate via another.

4.9 Limitations

We recruited participants that customized their communication apps, so the interviews and questionnaires describe practices and problems of savvy users. We do not claim that all users struggle due to app-exclusive features, or that they invest effort in clever workarounds that stitch multiple apps together. However, we believe that the insights from this study are particularly relevant to people with strong mediated-expression needs that communicate in diverse social circles via rich ecosystems of communication apps. As people increasingly adopt multiple apps in their daily communication, their online selves may become more critical in the impression they convey to others. Thus, the need for mechanisms that bridge the expressive media and functionality between different apps is likely to become more popular.

4.10 Conclusions

In this chapter, I studied how users customize and repurpose features of their communication apps to make their mediated communication more personal. Through interviews and *story questionnaires*, we found that people customize their communication apps for expressing their identity, their intimate bonds with others and their local and organizational culture. We also found that, at the same time that users adapt their apps to their preferred communication style, new communication habits can stem from customizations.

A closer look revealed that, beyond customizations, exclusive expressive media (e.g. emojis, stickers) and functionality impose limitations to users' expression across their app ecosystems. Moreover, while two apps may support the same expressive functionality, differences in their interaction or implementation may discourage their use. Asymmetries in the aesthetics or availability of expressive media between communication partners (e.g. having different emoji representations between Android and iOS) also impose barriers to users expression, changing the meaning or even hiding non-verbal cues in their conversations.

We found that some users invested great amounts of efforts in inventive workarounds to overcoming these limitations and express them-

selves as they wanted. Workarounds revolved around sending similar expressive media to the ideal one, saving expressive media as a file to send it from any app, and generating rich content in one app to send it via another.

The customizations, breakdowns and workarounds described in this study demonstrate how communication habits acquired in one app transfer to others. Users customize and repurpose their apps to express their identities and intimate bonds, but app-exclusive features prevent them from doing it consistently across their many apps. The inventive workarounds that participants reported around leveraging the diverse functionality distributed in their app ecosystems for a particular conversation in a particular app provide evidence of a need to maintain a consistent expression style across different apps—an impossible goal under isolated and competitive character of today's communication apps.

5

Design directions for supporting ecosystems of communication apps

5.1 Implications of communicating via ecosystems of apps

In this first part of the thesis, I showed that communication apps are not used in isolation: users build personal ecosystems of communication apps where each shapes how communication happens in others. The studies in Chapters 3 and 4 illustrate two aspects of this phenomenon: the contacts and the features (i.e. expressive media and functionality) present in each app. Next, I summarize the main findings of how apps in an ecosystem influence each other and their implications for design.

On one hand, we learned that **people create communication places in their app ecosystems to draw boundaries between contacts**. Users perceive that bare presence of particular contacts in an app affects the communication with other contacts. To control this, they create boundaries by isolating them in different apps that become *communication places*, each with its own membership rules, perceived purpose and emotional connotations. Creating multiple communication places

in their app ecosystems grants users more control over how communication happens with their diverse relationships and social circles.

However, apps not only isolate contacts—they also isolate functionality. This forces users to compromise their communication places by “bringing outsiders” to access particular functionality. Moreover, apps treat all contacts the same, neglecting the nuanced, unique needs that a user may have for each of their relationships.

On the other hand, we learned that **people acquire expression habits from the features of each app and try to transfer them to others.** When they develop expression habits around a particular app feature, they adapt (i.e. customize) other apps, adapt to their limitations and combine elements from different apps in an attempt to maintain a consistent expression of their identity, culture and intimate bonds with others across apps.

Nevertheless, users often feel frustrated when an app cannot afford to mediate the expression habits they nurtured in other apps. Features and customizations are often app-exclusive, preventing users from expressing themselves consistently across their app ecosystems. This is especially frustrating when communicating within close relationships, which often span across many apps.

These insights point to the following **implications for design:**

1. People should be able to draw boundaries between their contacts without sacrificing functionality: Users distribute contacts in different communication places, but as relationships change, some contacts may fit better in different places than where they started. However, moving contacts from one place to the other implies changing the functionality that mediated the communication with them. We should design better mechanisms that help users preserve their communication places, helping them isolate contacts without isolating functionality from contacts at the same time.
2. People should have more control over their self-presentation across their app ecosystems: while people may leverage different apps to convey different impressions, they should also be able to convey the same impression across apps if they wanted to. The diversity of app-exclusive features provides opportunities to adapt their self-presentation in terms of what each app allows and restricts, exploring different ways of expressing their identities. However, users should have equal opportunities for adapting their apps to support

a consistent self-presentation. We should design mechanisms that allow users to express aspects of their identities across apps consistently.

3. People in close relationships should be able to express their intimate bonds and dedicated communication style across all the apps they share: As communication partners acquire shared expression habits from their entire app ecosystem, each app should support the habits that others nurtured. We should design mechanisms that support relationship-dependent communication styles across apps.

In the following sections, I propose four design directions for addressing these implications and better supporting ecosystems of communication of apps.

5.2 Allow multiple communication places within the same app

The first study provided examples of how users construct communication places and avoid contacts “contaminating” their apps: they distribute contacts among different apps or multiple accounts for the same app. The first approach trades off functionality for contacts segregation. The second approach allows for the same app to “host” multiple communication places, but never at the same time.

Instead of having *one* Whatsapp and *one* Telegram, mobile operating systems and app designers could allow users to create multiple instances of their apps to distribute their contacts in separate communication places with the same functionality. Users could still take advantage of all the features a particular app has to offer, but their communication remains independent from contacts that belong outside that specific communication place. For example, a user could “turn off” the *Highschool Friends Telegram* during a busy day but still receive messages from the *Family Telegram*. These separate communication places could have their own distinct icons and notification badges (e.g. as in Figure 5.1), allowing users to communicate with the contacts in one place without being aware of contacts in another. Settings and customizations such as showing notifications on the lock screen, ringtones, avatars and nicknames, collections of stickers and availability status could apply per communication place rather than per app—or

even better, let the user decide. Moreover, the communication places created with the same app would all share the same functionality; thus, users could express themselves consistently across them.

5.3 Support relationships across apps

While users may want to distribute their contacts in different apps, close relationships often belong to many communication places at once, such as colleagues that are also friends, or romantic partners. In such cases, the same relationship is mediated in different ways depending on the app they use at a given moment. What if communication functionality was associated with people instead of apps?

Today, mobile operating systems offer an “address book” of contacts that is independent of the apps the user installs. Third-party apps like WhatsApp interact with this address book to access data from contacts, such as name, phone number and profile picture. Nevertheless, this interaction is limited to importing and updating static data. All settings related to a contact, such as notification preferences and customizations, are associated with the internal representation apps create for each contact. Thus, the little functionality that users can associate to contacts is app-dependent, generating different settings across the app ecosystem for what, in reality, is a single person that holds a unique relationship with the user.

Contacts, instead, could be cross-app entities with system-wide capabilities. Users could associate expressive media and functionality directly with *people*, not just apps. For example, a user could mute all notifications from all apps except those coming from her boyfriend, and set a unique ringtone to messages coming from him, regardless of the app. Contacts could keep track of the emojis used in conversations across the entire app ecosystem to allow users to browse recently used emojis per contact rather than per app. Operating systems could also centralize access to the many conversations a user has with a contact, the pictures shared across apps, and other exchanged content, providing access to the communication history with *a person* rather than communication history in an app. Moreover, designers could build tools that address relationship-specific needs and allow users to combine them with their existing apps.

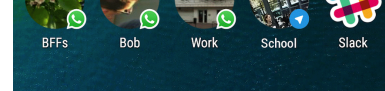


Figure 5.1: A communication app ecosystem with apps containing multiple communication places. The 🟢 icons are instances of *WhatsApp*. The 🔵 icon is an instance of *Telegram*. Apps could still be used as we do today, as in the case of *Slack*.

What kind of new communication functionality would best benefit from being associated directly to people rather than apps? How would communication partners negotiate and evolve their dedicated functionality over time? What kind of software architectures could enable mobile operating systems to represent contacts as cross-app entities? What kind of interfaces could best help users to associate functionality with their contacts? How would this change affect relationship maintenance online? I explore some of these questions in the following chapters.

5.4 Provide access to functionality from other apps

Communication apps try to differentiate themselves by offering exclusive features, but people use them side by side and often need the functionality of one app while having a conversation in another. For example, a participant from the study in Chapter 3 brought a contact from *Tinder* to *Telegram* so he could share a custom-made sticker with her. In Chapter 4, a participant created a video with a Snapchat effect and sent it to a contact on Instagram. In the first case, the participant broke their communication place by bringing an outsider. In the second, the participant preserved their communication place by generating content in one app and sending it in another—a workaround that requires too much effort, only works for features that produce conversational content, and may be reserved to savvy users that treat communication apps as content-generating tools.

Even though communication apps live in app ecosystems, they are designed as silos that encapsulate functionality so that users only access it through them. I propose breaking those silos, bringing down their walls, so that users can access functionality from other apps from within any conversation. Researchers have challenged the model of closed, isolated apps (Beaudouin-Lafon, 2004; Tchernavskij, 2017), and explored alternative models in multi-device ecosystems (Klokmore and Beaudouin-Lafon, 2009) and web environments (Klokmore et al., 2015). I encourage app designers to also share and compose functionality in the domain of communication apps.

Apps could provide universal access to features that support expression, such as stickers, editing tools for pictures, special video effects. Apps can already communicate with each other to share content, e.g.

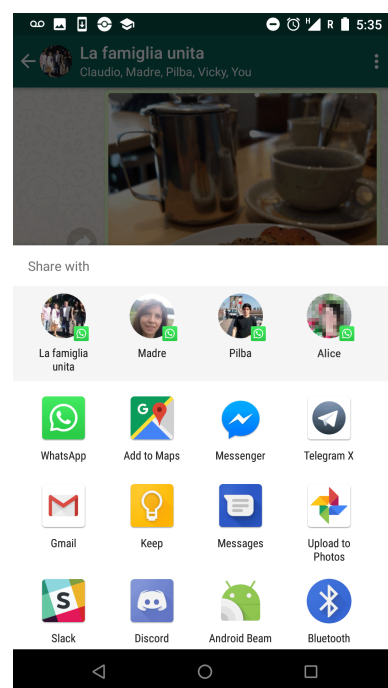


Figure 5.2: Options for cross-sharing a picture from Whatsapp to other apps.

a picture taken with the *Camera* app or a location from the *Maps* app can be shared directly with any messaging app, and a picture from a WhatsApp conversation can be forwarded to a contact in Messenger 5.2). Similar mechanisms could be used to provide universal access to the expressive media and content-generating tools that are currently *trapped* inside each app. For example, a user talking to her boyfriend on *Messenger* could take a picture and add *Snapchat* decorations without having to leave the app. A user that installs Instagram could access its video effects from Snapchat.

What kind of new interfaces and interaction techniques could help users access functionality from other apps? To what extent is this design direction feasible today, and how should apps and mobile operating systems change to enable it fully? What kind of new communication functionality would best leverage the possibility of being used from any other app? I explore some of these questions in the following chapters.

5.5 Enable user-owned communication tools

Today, apps define what functionality is available in the conversations they mediate, and users can only access functionality from within an app. What if, additionally, users could own communication tools that belong to no app in particular?

Mobile operating systems could allow users to have their own *communication toolbox* with expressive media and functionality that belong to no app in particular but can be carried from app to app. For example, a user's communication toolbox could include three of their favorite stickers, a speech-to-text tool, a self-made animated GIF, and a camera tool with various photo filters. The user could open this toolbox from within any app, and use each tool to provide direct input to the conversation. For example, the camera tool would produce pictures, and the speech-to-text tool would produce text, which are data types compatible with the type of input any messaging app can handle.

This design direction may even open the door to a new app-independent market of communication tools, which users could browse, select and customize to actively personalize the communication capabilities of their app ecosystems.

What kind of mechanisms and interaction techniques could help users access their own communication tools within their apps? What kind of new communication patterns would emerge from mixing user-owned communication tools with app-exclusive features? To what extent is this design direction feasible in today's mobile operating systems? What kind of new communication functionality would best leverage the possibility of using it across the entire ecosystem and customizing it independently of any app? How would user-owned tools affect self-presentation in app ecosystems? I explore some of these questions in the following chapters.

In Part Two, I repurpose three mechanisms available in current mobile app ecosystems for blurring the boundaries between apps and exploring the last three design directions proposed above:

1. Notifications, which display information outside of the apps and can be superimposed on any open app;
2. Gesture commands, which could invoke expressive media and functionality across any app that recognizes gestures; and
3. Soft keyboards, which appear in the context of any app mediates communication with text and can be extended with rich input functionality

PART II

DESIGNING FOR ECOSYSTEMS OF COMMUNICATION APPS

6

Repurposing Notifications

In this chapter, I propose using notifications for enabling relationship-tailored communication tools that augment the entire app ecosystem and belong to no app in particular. The *Lifelines* project, presented below, explores the intersection between **enabling user-owned communication tools** and **supporting relationships across apps**.

Notifications display events or important information about an app *outside* its main interface. In other words, notifications can show content from an app without opening the app. They usually appear in a *notifications drawer* that the user can pull down from any screen (Figure 6.1), and some users also choose to show them in their lock screen..

Communication apps typically use notifications for alerting about incoming communication, but they increasingly support richer functionality beyond showing the text of a message (Figure 6.1). For example, in Android, Messenger’s notifications show two buttons: one for sending a “like” and another for replying from within the notification. WhatsApp also shows a button for marking the message as “read”. Other types of apps use notifications to provide global access to their functionality, such as AzRecorder, which shows buttons of video recording commands instead of text (Figure 6.1), or Google Maps, which allows users to navigate a route with up and down buttons (Figure 6.2). These examples show that, despite their limited screen real-estate, notifications can serve as a versatile mechanism to access functionality from outside apps.

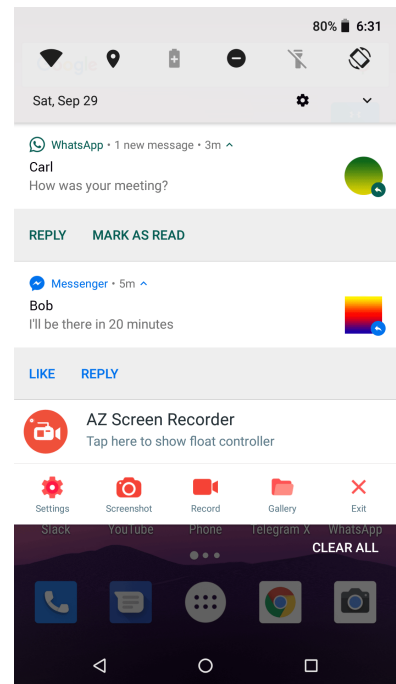


Figure 6.1: Android’s *notification drawer* showing three notifications: WhatsApp’s notification shows a “Reply” and a “Mark as Read” button. Messenger’s notification shows a “Like” and a “Reply” button. AZ Screen Recorder’s notification shows buttons for its video editing functionality, so users can screen-record while using other apps.

The following study presents *Lifelines*, a technology probe that repurposes notifications as peripheral displays of shared contextual information for couples. Through the creation and deployment of *Lifelines*, I explore the implications of designing communication tools dedicated to close relationships, as well as the design and technological challenges behind augmenting app ecosystems with notifications.

6.1 Lifelines: a dedicated communication channel for couples

Couples communicate via complex ecosystems of apps. Cramer and Jacobs (2015) expose how romantic partners intentionally leverage specific characteristics of diverse apps to adapt to each other's routines or add meaning to their messages: one app can become the "working hours" channel and another the "leisure"; importance or urgency can be conveyed by sending the same message via multiple apps; and a minimalist Yo signal¹ can mean "I'm thinking about you".

While couples communicate via multiple apps, commercial apps strive to become the *one* app where all communication starts and ends. Apps for couples (e.g. Couple, Between²) duplicate rather than enhance the basic functionality of mainstream apps, such as Facebook Messenger or Whatsapp. For example, partners must call, text and share their location from within the Couple app to access its exclusive features, such as collaborative live sketching and *thumbkisses*—a vibration that triggers when both partners place their thumbs on the same spot of the screen.

We want to provide communication functionality tailored to couples in a way such that it complements rather than competes with their existing communication app ecosystem. A common way of supporting couples' communication needs in the literature is to provide them with awareness of each others' context when apart (Hassenzahl et al., 2012), taking advantage of their intimate knowledge of each other (Bales et al., 2011; Bentley and Metcalf, 2007; Lottridge et al., 2009). Minimalist cues of partners' contextual information such as their location (Bales et al., 2011), motion (Bentley and Metcalf, 2007) or ambient sound (Lottridge et al., 2009) become meaningful through the lenses of their known routines. Researchers have successfully mediated awareness by streaming different forms of contextual information, which

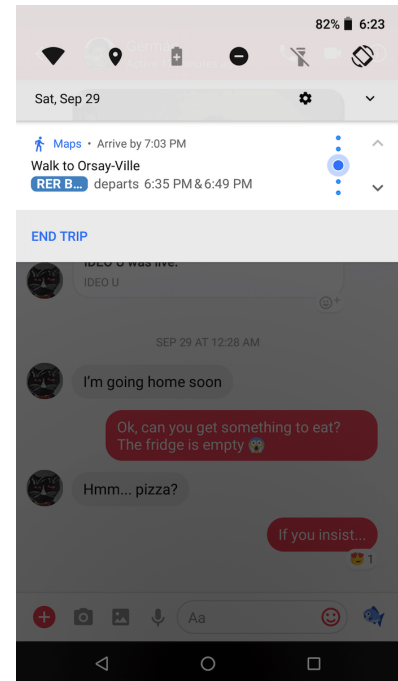


Figure 6.2: Google Maps notifications show "up" and "down" buttons to navigate the steps of a route, from origin to destination.

¹ Yo app: <https://www.justyo.co/>

² Couple app: <https://couple.me/>. Between: <https://between.us>.

helps couples find peace of mind, coordinate everyday tasks, increase feelings of connectedness and support other relational needs (Bales et al., 2011; Bentley et al., 2015; Bentley and Metcalf, 2007; Hassib et al., 2017; Lottridge et al., 2009; Schildt et al., 2016).

These studies focus on *ephemeral* streams of contextual information, which communicate live data only. We believe that *persistent* streams communicate richer information by revealing the duration and sequence of activities. Moreover, most previous work supports awareness via a *single* type of contextual information stream. *ContextChat* combines a collection of ephemeral streams within a chat application (Buschek et al., 2018), but it requires users to abandon their usual *communication places* (Nouwens et al., 2017) to benefit from their shared contextual information.

We built the *Lifelines* technology probe (Hutchinson et al., 2003) to explore how couples integrate *multiple persistent* streams of contextual information into their communication. *Lifelines* provides peripheral awareness via a notification to augment couples' entire app ecosystems with contextual information. We deploy *Lifelines* in the wild to generate empirical results about changes in couples' behaviour and inspirations for the design of future technologies.

We first review the related work on mediated communication for couples and close relationships. We then present *Lifelines*, a persistent visualization of multiple streams of contextual information that augments couples' communication app ecosystems. Next, we report the results of a one month field study with nine co-located couples. We show how *multiple persistent* streams supported rich interpretations about each other's context, helping couples communicate more often, be more understanding and stay in sync throughout the day. We also discuss how *multiple persistent* streams caused frictions in a few occasions. We propose implications for the design of future communication technologies for couples, and conclude with directions for future research.

6.2 Related Work

Couples' computer-mediated communication practices

Previous work shows the idiosyncratic information sharing practices of couples relative to other social relationships.

Couples often share sensitive data with each other, for convenience and as a symbol of trust (Matthews et al., 2016). Among diverse social relationships, users are most willing to share their location data with their partners (Consolvo et al., 2005). Research shows that couples share passwords (Singh et al., 2007), online accounts for entertainment and finance services (Park et al., 2018), calendars to manage common responsibilities (Neustaedter et al., 2009; Thayer et al., 2012,1), and their location to coordinate everyday tasks (Schildt et al., 2016). While lack of openness and trust may lead to monitoring behaviours (Marshall et al., 2013; Vinkers et al., 2011), trusting couples respect each other's privacy: having access to the other's data and devices does not imply its use (Jacobs et al., 2016).

Research also shows how couples hold parallel, mediated conversations on diverse topics (Andalibi et al., 2017), leave video-calls running in the background to feel each other's presence when apart (Neustaedter and Greenberg, 2012; Neustaedter et al., 2015), create shared accounts on social media (Zhao et al., 2012), and repurpose emoji with their own shared meanings (Wiseman and Gould, 2018).

These communication practices strongly suggest there exist couple-specific features that app designers should include, such as disclosing availability (Cramer and Jacobs, 2015; Neustaedter and Greenberg, 2012; Thayer et al., 2012,1), managing privacy settings for shared devices and content (Jacobs et al., 2016), supporting micro-coordination of everyday tasks (Bentley et al., 2015; Cramer and Jacobs, 2015; Schildt et al., 2016), encouraging reflection on the relationship (Zhao et al., 2012), and increasing connectedness by sharing awareness of their everyday activities (Neustaedter and Greenberg, 2012; Neustaedter et al., 2015). We focus on supporting micro-coordination, reflection, connectedness and availability cues.

Technologies for mediating awareness within couples and close relationships

Most commercial communication systems focus on exchanging messages. Close relationships create a sense of *connected presence* (Licoppe, 2004) by sharing small, casual messages throughout the day (Church and de Oliveira, 2013; O'Hara et al., 2014). However, this form of *connected presence* can break when one person is unavailable (Licoppe, 2004).

Technologies for mediating awareness “create a feeling of relatedness without direct communication” (Hassenzahl et al., 2012) by implicitly sharing presence cues between partners, e.g. music playing or ambient sounds (Lottridge et al., 2009). They can support connectedness when one partner cannot communicate explicitly (Licoppe, 2004) or during “empty moments” (Lottridge et al., 2009), i.e. times when one partner is involved in a mundane activity, such as queuing or riding the bus and misses their partner’s presence. Previous work explored diverse streams of contextual information for mediating awareness, e.g. location cues (Bales et al., 2011; Brown et al., 2007); travel time to a contact’s location (Bentley et al., 2015); drinking moments through a connected cup (Chung et al., 2006); playing music (Bentley and Metcalf, 2009; Lottridge et al., 2009); ambient sounds (Lottridge et al., 2009); appointments in a calendar (Tang et al., 2001), in-chat heart rate visualizations (Hassib et al., 2017); and motion in front of a camera (Riche and Mackay, 2007,1) or between locations (Bentley and Metcalf, 2007). Most research focuses on long-distance relationships (Branham et al., 2012), and few offer empirical results from longitudinal field studies (Hassenzahl et al., 2012).

Previous studies focus on how *single* streams support couples’ needs: location (Bales et al., 2011), motion (“moving”, “not moving”) (Bentley and Metcalf, 2007), playing music (Bentley and Metcalf, 2009) and heart rate information (Hassib et al., 2017) all increase feelings of connectedness, offer peace of mind (i.e., knowing that their partners were safe and sound) and signal availability. Heart rate (Hassib et al., 2017) and music (Bentley and Metcalf, 2009) also spark conversations about partners’ contexts. Other studies suggest that location and motion cues *reduce* direct communication, because partners consult the shared data rather than calling or texting about micro-coordination, i.e. “the exchange of information that allows for the on-going but mundane maintenance of everyday life” (Ling and Yttri, 2002).

Most of the above research explores *ephemeral* streams, as do most commercial applications that share live location information (e.g. *Life360*, *Couple* and *FindMyFriends*). Hassib et al. (2017) compare both *ephemeral* and *persistent* modes of sharing heart rate information within text conversations. Most participants preferred and felt more connected in the persistent mode.

We explore how couples' communication changes when sharing a combination of *six* types of *persistent* streams that, to the best of our knowledge, have not yet been studied.

In summary, previous studies show how couples benefit from mediating awareness via *single, ephemeral* streams of contextual information. We explore more comprehensive ways of communicating context with *multiple, persistent* streams. Because couples develop communication practices around their many apps, we want to extend their app ecosystems to enhance their conversations without duplicating basic functionality.

6.3 The Lifelines technology probe

Lifelines is a mobile app that captures and shares six streams of data: BATTERY, HOME, MEDIA, STEPS, and traces of SMS and CALLS within the couple. Each partner owns a lifeline, showing the last hour of data as a colorful visualization banner. Both lifelines appear together in a peripheral display: either on Android's notification drawer (Figure 6.3) or the iPhone's widgets screen. Partners can individually choose which streams to share and customize the visualization of their data.

As a technology probe, *Lifelines* is not intended as a prototype to be evaluated, but rather as an initial draft of a new technology that we can embed into user's real-world context to study changes in their behaviour and inspire design ideas for future technologies (Hutchinson et al., 2003). Thus, *Lifelines* helps us answer three questions: (1) How do couples' communication change when sharing multiple, persistent streams off contextual information?; (2) How should we design mediated awareness technologies so that couples can adapt them to their idiosyncratic communication habits and needs over time?; and (3) How can mobile operating systems better support dedicated channels that

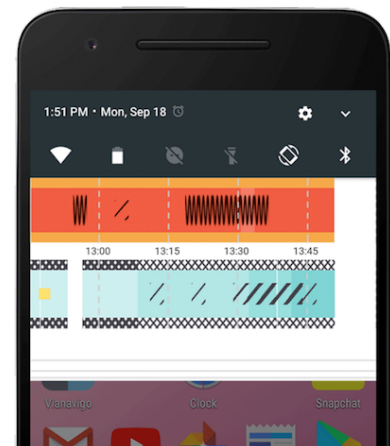


Figure 6.3: *Lifelines* peripheral display as a notification in Android phones. Each banner is a lifeline visualizing multiple persistent streams of contextual information: the user's (top) and her partner's (bottom).

augment couples' communication app ecosystem as a whole?

Multiple streams of contextual information

Daily routines involve diverse activities that cannot be represented with a single stream: couples that share their location may also value information about their behaviour *at* a location, e.g. if Alice saw data of Bob *walking at home*, she may infer that he is playing with the kids; if Bob saw data of Alice *listening to music at work*, he may interpret that she is trying to focus on her writing. We believe that sharing diverse streams can help couples better stay in sync, avoid interruptions and learn more about each other.

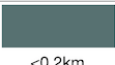
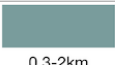
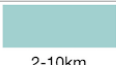
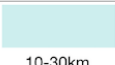
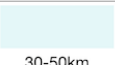





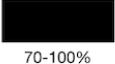
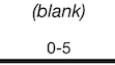
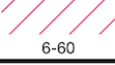




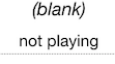
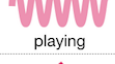

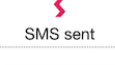
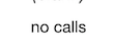

Stream	Description	Visualization Levels					
HOME	how close is the user to home	 <0.2km	 0.3-2km	 2-10km	 10-30km	 30-50km	 (blank) >50km
BATTERY	how full is the battery of the phone	 <15%	 15-30%	 30-50%	 50-70%	 70-100%	
STEPS	how many steps has the user made	 (blank) 0-5	 6-60	 61-115	 116-170	 171-225	 > 225
MEDIA	if audio is playing (e.g. videos, music, games)	 (blank) not playing	 playing				
SMS	if an SMS is sent to the partner	 (blank) no messages	 SMS sent				
CALLS	if there is an outgoing call to the partner	 (blank) no calls	 ongoing call				

Figure 6.4: *Lifelines'* six streams of contextual information. The table only shows an example visualization for each stream, users can customize all colors and textures.

We included six streams in *Lifelines* to help us observe how couples obtain context cues from multiple streams that share diverse aspects of their routines. As described in Figure 6.4: HOME and STEPS communicate different nuances of physical activity and displacement, potentially signaling key moments of partners' routines (e.g. leaving for work, lunch breaks or errands); BATTERY may help partners explain missed calls, and reveal traces of phone usage habits (e.g. a fast-dropping battery might signify playing a game, and a rising battery level might suggest the user went to bed); MEDIA reveals that the user is actively using their phone and signify diverse activities and habits (e.g. listening to music while cooking, watching videos on the train, or an alarm ringing in the morning); and CALLS and SMS visualize traces of past direct communication that may help contextualize the other



Figure 6.5: A lifeline of one partner heading home on the bus. She texts her partner at the bus stop (SMS: red zig zag). She plays a game (MEDIA: pink waves) which drains her battery (BATTERY: fainting cross hatch). She calls her partner (CALLS: yellow bar) when she gets off the bus and starts walking (STEPS: red diagonal lines). To save power, she turns off her location service (HOME: blue solid disappears)

streams. We selected these streams based on two criteria: a) the data should be easy to capture on both Android and iPhone phones; b) the data should help infer diverse types of activities and phone status from partners' individual routines.

A lifeline visualizes all streams as a stack of layers (Figure 6.5) to meet three design goals: a) to visualize multiple streams in a compact way, rather than allocating space for each; b) to help partners infer context from multiple streams simultaneously, as well as from individual streams; and c) to allow partners to toggle streams on and off without hurting the aesthetics of the visualization.

Persistent streams of contextual information

Historical data may help partners interpret richer context from the streams they share based on the duration and sequence of their activities. For example, long walks may indicate more significant activities (e.g. exercising, shopping) than just a few STEPS. Playing music for 20 minutes and then walking may signal a partner arriving home.

Lifelines persists all streams in a visual timeline (Figure 6.5). Every three minutes, it takes a snapshot of HOME, BATTERY, STEPS and MEDIA and adds a new 3-minute segment to the user's lifeline. SMS and CALLS are captured as soon as they happen. Sensing data every three minutes not only helps reduce *Lifelines'* battery consumption, but also requires abstracting and aggregating data. This adds ambiguity to the visualization (Gaver et al., 2003), better preserving partners' privacy and encouraging interpretation based on their knowledge of each other.

The *Lifelines* peripheral display shows the last hour of data, and tapping on it opens the associated *Lifeline* app where the users can "scroll back in time" up to 18 hours. We chose 18 because 24 hours takes too long to load and slows the phone.

Peripheral awareness of contextual information

We want to enhance couples' communication without competing with their favorite messaging apps. Therefore, we designed *Lifelines* to add novel couple-specific capabilities to their app ecosystem and did not include features from existing apps, such as calls and multimedia messages. We designed *Lifelines* as a peripheral display to serve two purposes: a) to be explicitly used in conjunction with any communication app, and b) to allow for serendipity with peripheral awareness of each other's context, i.e. partners may glance at *Lifelines* accidentally without opening a specific app.

Lifelines appears as a *sticky* notification in the notification drawer of Android phones, meaning it cannot be dismissed (Figure 6.3). As with any other notification, users can choose to show *Lifelines* in their lock screen as well. This means that *Lifelines* can be peeked at while in another app, rather than having to be opened separately.

iPhone users must swipe left from the *Lifelines* widget on the home screen or notifications drawer. This makes *Lifelines* more cumbersome for iPhone users, since it requires an extra interaction.

Customizing with Linebuilder

Partner may prefer sharing different, individual choices of streams, depending on what data is relevant in their routines and their privacy preferences. Moreover, we speculate that customizing the aesthetics their lifelines may help them express emotions or change the meaning of the data shared (e.g. a partner setting HOME to yellow to signal she is with her family, or blue if she is alone).

The *Linebuilder* (Figure 6.6) lets partners customize which streams to share asymmetrically. They can also customize the textures and colors of each stream. The customizations are persistent, i.e. when a lifeline design changes, it only affects the data shared from that moment on.

Implementation

We took advantage of the Automate³ app, which lets users create automation flowcharts and connects to many of the phone's sensors. This

³ llamab.com/automate

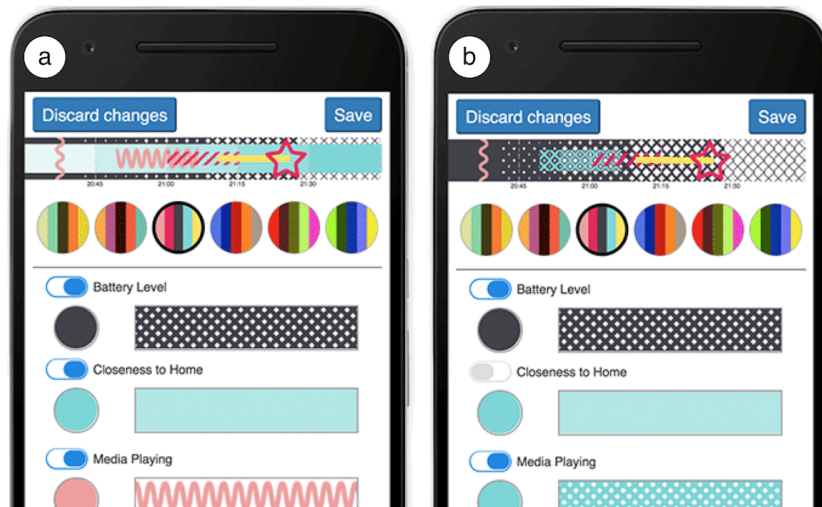


Figure 6.6: The *Linebuilder* lets each partner customize which streams they share, as well as the textures and colors for each stream. The top banner shows a lifeline preview. a) All streams are active. b) HOME is off, and MEDIA now appears as a blue crosshatch instead of pink waves.

greatly facilitated sensor data capture across many versions of Android. We collected iPhone data via the native *Lifelines* app, but were unable to capture outgoing calls and SMS messages because iOS prohibits third-party apps from accessing this data. This forced us to exclude couples who both had iOS devices. As long as one partner had an Android phone, we could register both incoming and outgoing SMS messages and calls, and use the Android phone's 'incoming' data to visualise the 'outgoing' data from the iOS phone.

Lifelines sends context data from each phone to a Node.js server which generates each lifeline on demand, i.e. when opening the *Lifelines* app, when iPhone users access the *Lifelines* widget, or when Automate updates *Lifelines*' sticky Android notification. The visualization is an HTML page built dynamically with JavaScript and styled with CSS. This lets us reuse the same visualization mechanism for both Android and iOS phones, without implementing native versions of each.

6.4 Method

We ran a one-month field study, using *Lifelines* as a mobile app technology probe (Hutchinson et al., 2003).

Participants

We recruited 9 co-located couples (9 women, 9 men, aged 22-43) via social media, mailing lists, and word of mouth. Participants were living in Argentina, Brazil, France, New Zealand, Switzerland and the United Kingdom, originally from Argentina, Brazil, Venezuela, Korea, China, Italy, the United Kingdom and the United States. All but one couple lived together, with relationship length ranging from 7 months to 8 years. One couple had a 9 year-old; another had 2 children under 5.

We only accepted couples with at least one Android phone, due to restrictions from iOS described above. Participants with Android phones could keep the premium version of the Automate app, with no other compensation.

Participants communicated via WhatsApp, Telegram, WeChat, Skype, Snapchat, Facebook Messenger, Google Hangouts, Twitter, and Email, and all but one couple stated that they rarely use SMS and phone calls. Three couples reported apps we had not anticipated: Couple, a couple-specific application; Zenly, for sharing live location and battery level with contacts; and Strava, for tracking and sharing running and cycling activities. All couples used more than one app to communicate.

Procedure

The study consists of four parts divided over a one-month period (Figure 6.7).

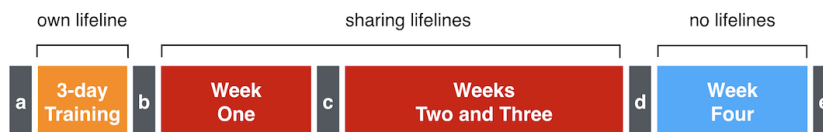


Figure 6.7: Overview of study timeline. Letters mark interventions: a) Setup Session; b) Partners start sharing their lifelines; c) First Interview; d) Second Interview, partners stop sharing their lifelines; e) Final Questionnaire.

Pre-study configuration

We ask participants to sign an informed consent form which describes the type of information collected and informs them of their rights. We collect their phone number and home address via a digital questionnaire to configure the SMS, CALLS and HOME streams and create their user accounts on the *Lifelines*' server. Finally, we send them email invitations to download and install the probe, including a description of the functionality of *Lifelines*.

Setup and three-day training

We walk the participants through the installation process of *Lifelines*, either in person or via teleconference depending upon their preference or location. Their lifelines in the *Linebuilder* app are prepopulated with dummy data (see top banner in Figure 6.6) to give them an idea of what a full lifeline would look like and the participants are asked to explore how *Lifelines* works by toggling data streams and choosing among different patterns and color palettes. Once satisfied, they save the design and look at their actual lifeline (with the data captured between installation and configuration) in the notification or widget. We ask them to perform actions that trigger a response on the lifeline (e.g., walking, playing media, calling or texting their partner) so they can understand how the app works. We explain how they can record their thoughts, interesting events and breakdowns via MEMOS: comments they can attach to a particular lifeline moment, acting as in-context diary entries. They fill out a second questionnaire about their current communication apps and devices. After the setup, the participants enter the three day training period in which they only see their own lifeline, so they can learn how to read the visualization and reflect on which data streams they want to share or hide from their partner.

Week One: Shared Lifelines

We send participants an email explaining that they will now see their partner's lifeline in addition to their own. We also remind them that they can modify their lifeline choices at any time and prompt them to record MEMOS of interesting events.

At the end of the first week, we interview each partner individually for 30-45 minutes by teleconference. We prompt them for specific detailed stories of what happened using Mackay's 2002 variation of the critical incident technique (Flanagan, 1954). We show all data collected from the previous week to help them remember, offering a historical view of both their lifeline and their partner's (Figure 6.8). The webpage with this data is only available to them during the interview.

We ask the participants to share their screen so we can see how they point at particular areas of their historical lifelines as they describe specific situations. Our questions focus on moments where *Lifelines* affected their (mediated) communication and tensions between the streams they would have liked to see on their lifelines and the streams that were shared. We asked about their favorite, least favorite, or most surprising episodes with *Lifelines*. We use the recorded memos, changes in their lifeline designs, technology breakdowns, and toggling of data streams as moments of interest, around which we ask them to

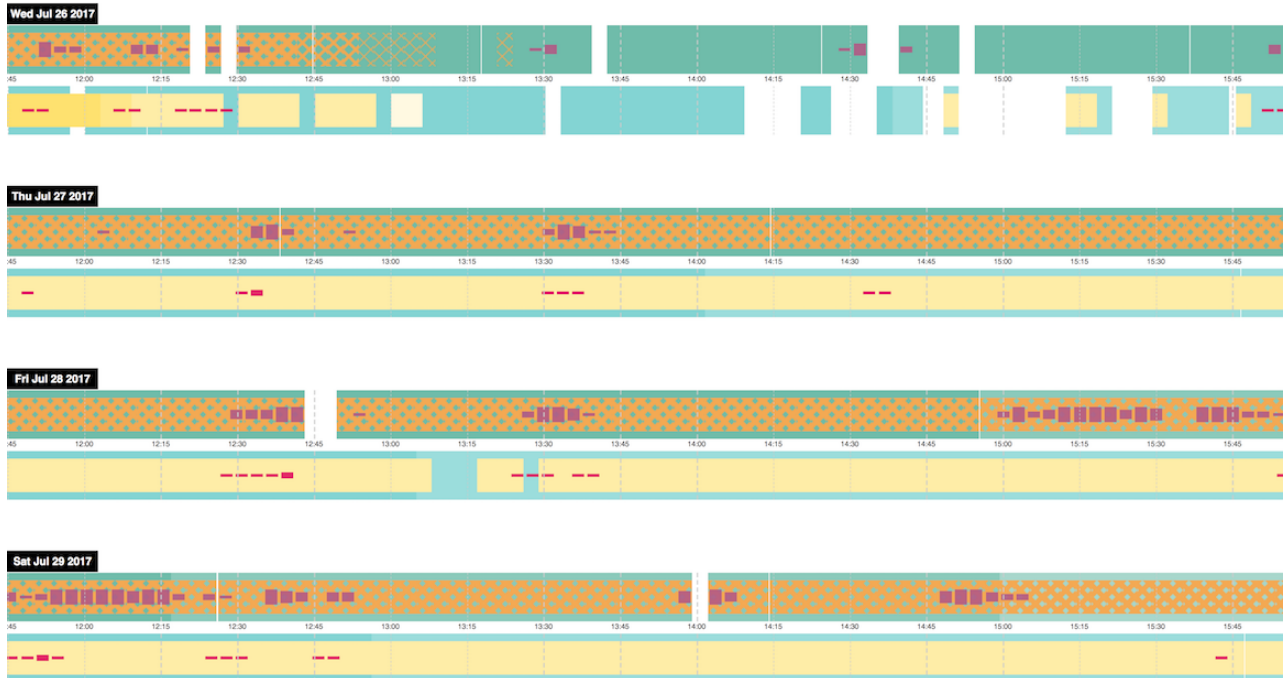


Figure 6.8: Webpage showing Lifeline’s historical data for one couple. Each day shows two timelines of 24 hours, one for each partner. Participants could scroll up and down to navigate days, and left and right to navigate hours.

explain why something was significant, why they made that decision, or how it affected their communication with their partners.

Weeks Two and Three: Shared Lifelines

Partners continue to share their lifelines for two more weeks. At the end of Week Three, we interview each partner individually (similar to Week One) and probe for additional stories about their experiences with *Lifelines*. At the end of the interview they uninstall *Lifelines*.

Week Four: Post Lifelines

We asked participants to complete a questionnaire one week after they stopped using *Lifelines*. We ask them what they miss, if anything, and whether or not they would prefer to continue using *Lifelines*.

Data collection

We log all the information that *Lifelines* uses to construct the display: GPS coordinates, number of steps, battery level, whether media is playing or not, and the timestamp of sent SMS and outgoing calls to the participant’s partner, and the qualitative data from their MEMOS. We also log configuration changes by each partner (toggling streams

on/off or changing the design). We are unable to report on the frequency participants looked at their lifelines, as we cannot distinguish between opening the notifications drawer or widgets screen to check *Lifelines* or for a different purpose (e.g. reading an incoming SMS or checking the weather). We also collect questionnaires and interview data, and audio and screen-record teleconference interviews.

Analysis

We coded approximately 22 hours of video (audio and screen recordings) from 34 recorded interviews, resulting in 368 salient interview excerpts. We conducted four interviews per couple, except for one couple who only participated in the second interview because of time limitations. Three participants preferred conducting their interviews in Spanish, and we translated their interview excerpts to English. One author performed open coding on the first batch of 12 interviews. A second author received the transcribed interview excerpts and the codes separately, and re-coded the data. The few disagreements in the re-coded data were discussed and incorporated into the analysis, producing 89 codes in total. The first author coded 16 additional interviews. The full research team read over the full set of interview excerpts and codes, and they were discussed over multiple meetings. After several iterations, we agreed on latent themes (Braun and Clarke, 2006; Brown and Stockman, 2013) presented in the section below. The remaining six interviews (2 from each of 3 couples) provided additional examples of the established themes.

6.5 Results and Discussion

We first present the most novel findings about how sharing *multiple, persistent* streams of contextual information affects couples' communication dynamics. Secondly, we touch on how customization of the streams enables individuals to express their identity and care for their partner, individual differences in privacy preferences, and participants experiences after *Lifelines* was removed.

Inferring context from individual vs. combined streams

Participants leveraged the multiplicity of data streams in two ways. First, by having access to diverse types of information they could choose the most appropriate data to look at when trying to answer a question or satisfy their curiosity. For example, Inez looked for isolated STEPS to call her partner while he was on a break at work, and whenever she saw traces from his MEDIA stream she asked him how he was liking his audiobook. Knowing *which* data stream was most appropriate to look at and being able to connect it to a very particular activity—STEPS with a work break or MEDIA with listening to an audiobook—was possible due to the intimate knowledge Inez had about her partner’s routines.

Second, having access to multiple streams allowed participants to combine information and reveal more detail about their partners’ context. While this happened only rarely (most of the stories centred around using just a single stream), the cases in which it happened shows how participants layered streams to distinguish between activities. For Hugo, if his partner’s MEDIA was playing while HOME was changing, it meant she was driving, but if HOME looked dark and was not changing, it meant she was watching videos from social media at home. Mona differentiated long traces of STEPS depending on whether her partner was close or far from home: if far, it meant he was on the university campus, if close, it meant he was walking the dogs around the block.

Sharing *multiple* streams of contextual information allowed partners to infer diverse activities throughout the day by looking at individual streams, and to disambiguate the meaning of a stream by combining it with others.

Triggering vs. replacing direct communication

Single stream studies describe how some information sharing *replaced* direct communication, e.g. knowing the partner’s location replaced asking “where are you?” (Bales et al., 2011; Schildt et al., 2016), whereas other types of information *triggered* direct communication, e.g. heart rate triggered “what are you doing?” (Hassib et al., 2017). We observed a different dynamic, where replacing and triggering communication did not depend on the *type* of data shared, but whether a stream *confirms* or *challenges* a partner’s knowledge of the other.

When a stream shows data that *challenges* partners' knowledge of each other, it sparks curiosity and *triggers* direct communication. Surprising data often exposed previously undiscussed topics, stimulating partners' relationships with novel information about their habits and interests: John asked his partner about her STEPS at home and learned more about her daily routines. He was happily surprised when she asked about his MEDIA use: "She saw on my lifeline that I was playing media a lot, and she asked me what it was. And it was because I had installed one of those farming games. And then she installed it, and now we're both playing. (...) The game created a new bond. So now we have this new conversation topic". Inez used STEPS and MEDIA as conversation starters: "I can ask him "Where are you? What are you doing? I can see that you're walking". So it did improve the communication because I have these small hints". Similarly but inverted, the surprising *absence* of CALLS and SMS triggered participants to reach out to one another: "You look at the lifeline and you say 'oh, there are no calls, no messages' and that encourages you to text him or call him" Dory.

When a stream shows data that *confirms* partners' knowledge or expectations of the other, it *replaces* the need for direct communication, in particular common check-in moments such as asking whether the other was safe, determining their availability, and managing micro-coordination. For example, Hugo knew his partner listened to music while driving, so he checked her HOME and MEDIA to see whether she arrived home safely. Rick used his partners' HOME stream (Figure 6.9) to estimate when he should have dinner ready: "I remembered her closeness to home started to show about 50km from home, so I know that means she'll arrive home in about an hour. I decided to order outside. So I estimated when she would be arriving (...) and actually she was at the door with the delivery guy. I didn't check with a message because she was driving, she wouldn't be able to answer".



Figure 6.9: Rick's partner going back home by car. She is outside the city around 21:00hs (HOME: pink crosshatch) and arrives home around 22:00hs.

Thus, whether a stream *triggered* or *replaced* communication did not depend on the type of data that was being shared, but on the value of the information for the participant. That relative connection between the stream and the way a participant used it settled into stable new communication dynamics: they would frequently use one stream to trigger and another to replace. The sharing of multiple streams of contextual information impacted couples' established communication

patterns and resulted in a new dynamic, with different kind of questions asked (or not) and different kinds of topics discussed.

Consequences of new communication dynamics

Previous work extensively discussed how sharing contextual information supports micro-coordination without direct communication (e.g. Bentley and Metcalf, 2007) and how it helps increase feelings of connectedness and reassurance between partners (e.g. Bales et al., 2011; Hassenzahl et al., 2012; Hassib et al., 2017). While our results largely echo the literature, an exception stood out: Hugo felt less connected to his partner because he missed her “checking on you” messages: “just by sending a message, it was a closer act than checking the app without talking to the other”. This prompted Hugo to hypothesize that “HOME is the most significant factor that takes place instead of sending a message”, so he turned it off as a test (Figure 6.10). Hugo’s partner noticed his

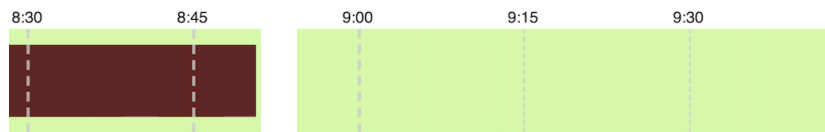


Figure 6.10: Hugo disabled his HOME stream (solid brown) around 8:50am. The persistent character of *Lifelines* reveals the moment he toggled HOME off.

HOME was missing and felt he was hiding something. She asked about this change, and Hugo explained that he turned his GPS off “to save battery”—since disabling the GPS and turning off HOME result in the same visualization, *Lifelines* offered Hugo *plausible deniability* in this situation (Lederer et al., 2004).

This story emphasizes that, when sharing contextual information, the shared streams must attain the right balance between *triggering* and *replacing* direct communication within the couple, otherwise, partners may feel more distant rather than more connected. In such cases, couples may require explicit discussions to agree on new codes of conduct around their new communication dynamics:

I think we’re in this transition of learning how to use the app and what is “polite” in a way. (...) Maybe using it longer I would learn how to use it properly, or rather just send a message instead of just waiting to receive a message (Hugo).

Inspiring deeper understanding between partners

Persistent streams of usage and phone state data allowed participants to augment ongoing conversations with a historical context. For exam-

ple, using recent information from their partner’s lifeline gave them a richer picture of the situation (Figure 6.11): “She was getting a bit drunk and dancing and having fun with her group mates. And there were some funny messages between us. And while that was happening you could see all these steps counting as she was dancing. And that was amusing ” (Owen).

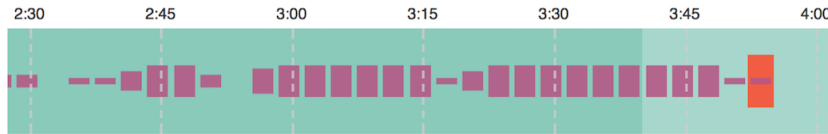


Figure 6.11: Owen’s partner dances (STEPS: magenta bars) as they text about her night out via Facebook Messenger.

Additionally, using information of the more distant past allowed participants to understand their partner from a different perspective: “I saw on the app that he listened to something until like 3 AM, so I was like “oh, you didn’t sleep very well, did you?” because I had all the information on the app. Otherwise I wouldn’t have known, or I wouldn’t have been as patient with him. So, this helped manage my expectations ” (Inez).

Some participants, after seeing persisted histories of unexpected activity, suddenly questioned how well they knew each other:

Sometimes you think the other one isn’t doing anything. In my case, I leave at 9 and I come back at 6. I left and came back and it’s like she’s been there the whole day. Instead if I look at the steps I realize that she’s been doing house chores, going out. It’s just a detail, but it changes your perspective (John).

The above examples illustrate the added value of persisting contextual data, adding temporality and sequentiality as an extra layer of information that enhanced partner’s shared understanding of each other.

Inferring context from patterns of missing data

Persistent streams of data also allowed gaps in the stream to become meaningful. Participants perceived the lack of data as yet another type of contextual information that informed them about: 1) their partners’ phone usage habits, and 2) the materiality of the technological infrastructure around them.

Partners interpreted missing data based on their knowledge of phone usage habits. Some Android versions prevented *Lifelines* from sensing and transmitting data when the phone had been unplugged and stationary for a period of time ⁴. Partners came up with their own ex-

⁴ Android’s Doze mode: Leaving a device unplugged and stationary for a period of time makes it enter in a *Doze* mode, which restricts apps’ access to the network and defers syncs and standard alarms to save battery. Source: https://developer.android.com/training/monitoring-device-state/doze-standby.html#understand_doze

planations for these technological glitches. In Barry's case, he treated the *absence* of glitches as a signal that his partner was actively using the phone (Figure 6.12):

You can see that there are many gaps, but when she's using the phone in her hand she's connected to Wi-Fi and you can see the full line. So I know when that's happening and "she's fine, she's using the phone". If she's using the phone, she has some peace time with the children not doing anything strange, or not running to the hospital or something like that. (Barry)

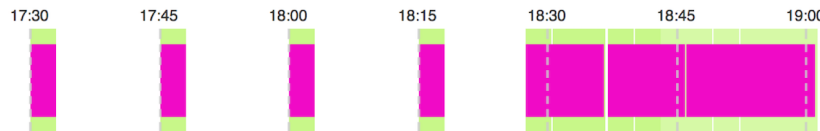


Figure 6.12: Barry's partner left her phone unattended, entering into *Doze* mode and revealing regular gaps of missing data. Around 18:30hs, she started using her phone, which resumes the normal functioning of *Lifelines*.

Participants also knew how the technological infrastructure surrounding their partners affected *Lifelines'* data streaming. For example, Dory knew her partner always lost reception in the area around his office: "When I see a gap on his lifeline, I know he arrived to work". Barry reported that the Internet connection often stops working in his office, but his phone remains connected to the Wi-Fi network. When this happens, he misses his partner's Whatsapp messages. He explained that because *Lifelines* also stops streaming data when this happens, his partner can easily figure out why he is not replying.

Customizing streams to express identity and care

Participants were able to express their identity and care for their partner in new ways through customizing the aesthetics of their lifeline. All participants customized their lifelines during the training period, and half customized them again at least once during the study. All lifeline designs were unique across the study, and some participants even asked for more color options and the possibility of adding their own texture designs. Customizability allowed the lifeline to reflect the personality and needs of the participant. For example, Kelly proudly explained that her partner was using his favorite soccer team colors. Nina changed her design to help her "spot new surprising things" about her own routines. For some participants showing caring was through sitting together and deciding on a design as a couple: Lucy and her partner playfully matched their lifelines, using the same palette but inverting which color went with which stream. Others did it by themselves, but with their partner in mind: Owen adapted his lifeline de-

sign to match the textures of his partner's, so she would understand his better. Three participants changed their design simply because their partners changed it first.

Individual differences in privacy preferences

Technologies for sharing contextual awareness raise privacy concerns, and our study provides further evidence that there are substantial individual differences in this area (Wilkinson et al., 2018). On the one hand, we had difficulty recruiting participants, with some candidates explicitly saying the idea was “creepy”. On the other hand, the people that did participate raised no overarching significant privacy concerns, and some wanted to share even more diverse data.

Opposite to concerns about monitoring behaviour in the literature (Bales et al., 2011; Bentley and Metcalf, 2007; Hassib et al., 2017; Mancini et al., 2011), we found that our participants were generally quite open with each other. A few participants showed concerns about their privacy regarding us—the research team—rather than their partners. Pete explained: *“I was just concerned about the server and whether that information was stored somewhere... but not about her”*.

Our participants wanted their partners to use *Lifelines*, e.g., to replace asking where they were or what they were doing, which echos findings by Schildt et al. (Schildt et al., 2016). We also found that participants expected their partners to look at their lifelines as an expression of caring: Mona and John felt reassured by thinking that if anything bad happened to them while commuting, their partners would know. Lucy even felt disappointed at her partner because he checked *Lifelines* less often than her. The most contrasting example to concerns about monitoring behaviour was a participant that shared her information for her partner's sake, having no interest in his data: *“I rarely look at the phone. So for me, Lifelines doesn't add too much value. But I think it's valuable for him. If I arrive too late, he gets worried, he texts me and I don't even look at the phone. So this gives him information about whether I'm still in class, or if I'm heading home”* (Nina).

We did find individual differences in privacy concerns around specific individual streams. For example, many suggested unpacking MEDIA into more revealing streams: *“Right now MEDIA doesn't show exactly what I'm doing. I want to show him if I'm chilling, relaxing, watching Youtube videos or Spotify”* (Amy). On the other hand, others felt that MEDIA

was, at times, too revealing, e.g. Owen felt embarrassed on a day he overslept, as MEDIA revealed an *alarm snoozing* pattern. Similarly, Mona explained that while commuting, she would appreciate sharing her exact location to feel safer, in case something bad happened. In contrast, Eva felt that the HOME stream exposed too much information, preventing her from surprising her partner by arriving home early or meeting him at a restaurant. These individual differences expose deeply personal preferences about how revealing or discreet each stream should be, which can be context dependent (Barkhuus, 2012).

Westin studied trends in individual differences around privacy needs and classified people as (1) *Fundamentalists*, (2) *Pragmatists*, and (3) *Unconcerned* (Kumaraguru and Cranor, 2005). While these categories refer to concerns towards companies collecting personal data, we find them helpful to describe the trends in privacy needs in our data as well. We suspect that our participants were largely from the *unconcerned* category. For example, Fay and her partner shared their exact location and battery level with Zenly in addition to using *Lifelines*. Some participants fit better in the *pragmatists* category. For example, Barry had concerns about MEDIA and STEPS, but decided to share them to show his commute patterns. *Fundamentalists* are probably less likely to want to share any contextual information with their partners. Some of the people who chose not to participate in our study are almost surely from this category.

Post Lifelines

One week after interrupting the use of *Lifelines*, participants answered questions about the data they missed sharing. Fifteen reported missing at least two streams. Another two missed only one (HOME and BATTERY, respectively). Some participants explained how they compensated for the absence of *Lifelines*: Barry checked the “last seen” status of his partner’s Whatsapp more often to see if she was using the phone or busy with the kids; Rick mentioned he really needed to ask “Where are you?” or “Are you at home?” before anything else; and Lucy considered sharing her location via Facebook Messenger for the first time. Barry also started to share his location via Google Maps while on the train, so he could relax instead of coordinating the pick-up, implicitly saying: “here, you check where I am”. Google Maps recently added battery level information when users share their location with others. As a result, Barry excitedly reached out to us recently to say how happy he was to have this functionality again after months

without *Lifelines*.

We also asked whether participants wanted to keep using *Lifelines*: Four couples did, three couples did not, and two disagreed. The main reason mentioned for answering “No” was the battery drain and slower phone performance caused by the probe. Some said they knew each other well enough that the *Lifelines* data was not important enough. One participant felt uncomfortable because they saw it as “surveilling” their partner. Nevertheless, half of those answering “No” still missed some aspects of *Lifelines*: Owen missed showing that his BATTERY and seeing his partner’s STEPS because it felt intimate. Inez especially missed MEDIA for texting “hey, what are you listening to?”.

Discussion Summary

We found that partners used the diversity of streams to infer diverse activities from their lifelines by looking at individual streams throughout the day, or by combining streams together to obtain more precise meaning. We also observed that, regardless the type of data shared, streams that *challenged* a partner’s knowledge of the other *triggered* direct communication, and streams that *confirmed* a partner’s knowledge of the other *replaced* direct communication. This introduced new communication dynamics that most leveraged to coordinate tasks implicitly, find reassurance and feel more connected. We also found that a poor balance between triggering and replacing direct communication can lead to feeling more distant rather than more connected.

Sharing *persistent streams* revealed patterns of missing data, from which partners obtained extra contextual information. Accessing past data also helped partners be more understanding with each other.

Partners customized their lifelines to express their identities and care for each other. We observed individual differences in how useful they found each stream, and the level of detail that would best support their communication and privacy needs. This points to further design opportunities around rich customizations and nuanced privacy settings for mediated awareness technologies for couples.

6.6 Implications for design

We deployed *Lifelines* as a technology probe to explore how couples appropriated *multiple persistent* streams of contextual information into their daily communication, and offer insights into the design of future mediated awareness technologies for couples. We offer the following suggestions, with a particular emphasis on the role of *multiple streams* (as opposed to single streams) and of *persistent streams* (as opposed to ephemeral streams).

Multiple streams

Allow partners to compose their own streams based on raw sensor data

Partners could tweak and create their own streams as they learn more about what data would best support their communication needs. Participants reported ideas for additional streams that could be derived from the same data we captured, e.g. John suggested that HOME was binary, only showing “at home” or “away”. Amy suggested differentiating between different kinds of MEDIA (alarms, music, audio messages, etc.). Custom streams could even aggregate data from both partners, representing the couple as a unit. Some suggested *distance from each other*, *moments together*, and *who did more steps*.

Help partners explore new streams that challenge or confirm their knowledge of each other

Encouraging partners to try new streams may help them discover data they had not anticipated to be useful. For example, a random *stream of the week* may help partners spark conversations and discover new things about each other, or recognize expected patterns of their routines that replace the need for frequent coordination and reassurance questions. Rotating through new streams periodically can also help maintaining partners’ interest in each other’s data, as they may find less surprises from the same streams over time. Exploring new streams may help couples such as Hugo’s, where the selections of streams they shared failed to balance the convenience of skipping frequent questions with the closer act of sparking conversations throughout the day.

*Persistent streams**Support discreet changes to privacy preferences*

When persisting data over time, gaps in the data stream convey technology-related issues (e.g. no reception), but also potential changes in a partner's privacy settings (e.g. toggling the HOME stream off). Rather than toggling individual streams off, designers could offer alternative ways of preserving partners privacy, e.g. entering into "incognito mode", where all streams hide temporarily as if there was no Internet connection. Users could also "freeze" their streams, repeating the last captured data for a limited time. As a different approach, visualizations of contextual information could mix persistent and ephemeral streams, allowing users to choose ephemeral representations for the streams they expect to toggle more often.

Support expressions of caring

Most participants expected their partners to look at their lifelines as an expression of caring. This points to an opportunity for letting partners explicitly indicate that they looked at each other's data. For example, some suggested leaving minimalist marks (Kaye, 2006; Riche and Mackay, 2007) or caring messages on a particular moment of their partner's past data. Other ideas revolved around saving pieces of data as "digital souvenirs" of nice moments together, capturing "a different perspective than a photo" (Owen).

Private repositories for contextual data

Services that mediate intimate, contextual information should allow users to store their data on their own devices, ensuring that only their partners have access to their data. This is especially relevant for persistent forms of contextual information sharing, which require storing past data to recreate historical views. Such capabilities may support users that want to share their contexts with their partners, but fear that third parties could track and exploit their data.

*Designing for app ecosystems**Enable communication permissions by contact, not only functionality*

Lifelines's iOS version cannot sense CALLS and SMS, as iOS prevents third-party apps from accessing this information for security reasons. On the other extreme, Android offers permissions that, when granted, allow third-party apps to identify *any* incoming call and read *all* details and content from incoming SMS messages. Mobile operating systems

should manage these permissions not only by functionality, but also by contact. For example, they could grant *Lifelines* permission to access outgoing calls to the user's partner, rather than all outgoing calls (like Android) or none (like iOS).

Support integration of communication apps with mediated awareness channels

Current commercial apps could better integrate with mediated awareness channels. They could offer open APIs for integrating calls and messages, or even accessing exclusive streams, e.g. times that a partner pressed the Facebook Messenger's like button, or screenshots of snaps from Snapchat. Non-communication apps could also offer streams, e.g. the remaining minutes of an audiobook, number of Pok'emon caught in the last three minutes, or the estimated time of a ridesharing trip. Some participants regretted that *Lifelines* showed no traces of their communication via apps, but no couple "moved out" from their usual communication places (Nouwens et al., 2017) to use SMS and CALLS instead. This highlights the importance of integrating rather than competing with existing apps.

Enable notifications as interactive spaces for hosting communication tools outside existing apps

Mobile operating systems should enable richer mechanisms for using contextual information in conjunction with other apps. Despite *Lifelines'* limitations as a probe, participants often combined it with their apps: Owen looked at his partner dancing while messaging on Facebook Messenger, and Barry crossed Whatsapp's *last seen* status with *Lifelines* to know since when his partner had left the phone unattended. Hugo mainly communicated via Whatsapp, and used *Lifelines* to know when his partner was watching Instagram videos (thus, available for hanging out). Dory explained: "I don't see it as an app, it's rather part of the phone". Some suggested rich interactions on their lifelines, which current notifications and widgets cannot support: Amy wanted to draw on her data, and Rick wanted to anchor urgent messages from Whatsapp to his lifeline. We believe that notifications can not only help augmenting app ecosystems with contextual information, but with other kinds of communication tools that belong to users instead of apps. We encourage vendors of mobile operating systems to explore novel mechanisms that can serve as peripheral displays and "hosts" of user-owned communication tools, enabling richer interactions than what notifications or widgets afford today.

6.7 Limitations

Our longitudinal field study provided grounded data from couples over a one-month period, which mitigated novelty effects, and captured stories from diverse life circumstances. While our sample of 9 couples does capture diverse cultures, representing different parts of Europe, North and Latin America, and Asia, it does have limitations. All our couples are middle-class and heterosexual. Perhaps most importantly, participants were self-selected, so they all had at least a degree of comfort sharing contextual data with their partners from the outset. *Lifelines* is not intended for all couples—it is a technology probe for studying how multiple, persistent streams affect the communication dynamics of couples that feel comfortable sharing contextual data. Designers and researchers should be aware it is likely that our sample—and thus our results—are biased in favour of people who have little privacy concerns and thus were happy to participate. While our data lacks examples of deliberate misuse, we believe that companies interested in deploying such technologies for the general public should include documentation that helps victims defend against abuses (Freed et al., 2018) and collaborate with professionals in the IPV ecosystem to create procedures for detecting, reporting, and retaining evidence of abuse (Freed et al., 2017). Since mitigating digital abuse via privacy settings may trigger other forms of violence (Matthews et al., 2017), context-data sharing technologies may have to restrict their streams to ambiguous representations that support plausible deniability.

6.8 Conclusions

In this chapter, I repurposed notifications as a mechanism for **enabling user-owned communication tools** and **supporting relationships across apps**. I explored both design directions through the design, development, and deployment of *Lifelines*, a dedicated communication tool for couples.

Lifelines augments partners' entire app ecosystems with contextual information. It leverages notifications as peripheral displays that partners can overlay on top of their existing communication apps. In this way, notifications enable *Lifelines* to act as a user-owned (or rather,

couple-owned) communication tool. While Lifelines does not provide direct input to couples' existing applications, our study shows how couples incorporated the shared information in their daily lives and changed their communication dynamics.

Lifelines also provided couple-specific functionality that different couples appropriated in diverse ways. Couples interpreted their data through the lenses of their intimate knowledge of each other, which informed their communication independent of the communication apps they used. As a probe, Lifelines also offered insights into how couples would like to integrate data and functionality from their existing apps into a dedicated channel just for them, e.g. showing communication traces and shared pictures from other apps, not just SMS and calls; and sharing their data only between them, not third-party services. These insights suggest that to best support relationships across apps, user-owned communication tools should also be able to access functionality from other apps.

Future work should explore the engineering of more interactive notifications that can capture richer input from users as well as provide content to their apps. By contributing the design and study of the Lifelines probe, I hope to inspire richer mechanisms for enabling user-owned communication tools that support relationships across their entire app ecosystems.

7

Repurposing Gesture Commands

In this chapter, I propose using gesture shortcuts for **providing access to functionality from other apps** and **user-owned communication tools** as well as **supporting relationships across apps**.

Enabling an app with access to functionality from other apps may easily lead to an overflow of buttons and menus. App designers cannot know in advance what other apps the user has installed. Thus, the extensive functionality and expressive media from other apps would likely appear in long menus and lists, which make functionality harder to access and discover (Omanson et al., 2010). Rather than cluttering the screen real estate with the myriad of communication functionality spread across users' app ecosystems, I propose that users create gesture shortcuts to their favorite functionality, which they can perform in any conversation to execute commands from any other app.

Gestures could also provide access to user-owned communication tools. Rather than displaying them as buttons or menus within each application (or inside notifications, as illustrated in Chapter 6), they could be triggered by performing a gesture. This also provides a consistent way of accessing user-owned tools across apps, as users could perform the same gesture for the same functionality regardless of the app they are using.

Last, users could associate gestures not only to functionality, but to

contacts as well, supporting their close relationships across app ecosystems. For example, users could associate gestures to system-wide settings for contacts (e.g. “mute all notifications from Bob”). They could also associate functionality from their apps and user-owned tools to specific people (e.g. “send voice message to Alice”).

Letting users design their own gestures may help them remember how to invoke each command. I believe such gestures should be highly customizable to increase users’ perception of ownership on their tools, and to let them express their identities and intimate bonds in the shapes of each gesture.

The following scenario illustrates how gesture shortcuts could help users better leverage the functionality and expressive media that is distributed across their app ecosystems.

7.1 Personal gesture shortcuts for communication app ecosystems: User Scenario

Celine is a 23 year old student in an international masters program. She uses Messenger for communicating with her family and closest friends from her home country, Whatsapp for her new international friends, and Slack for her classmates to coordinate projects and study meetings. She also uses Instagram to post nice pictures of her new city, and Snapchat to follow the everyday life of a couple of close friends. Daniel is Celine’s brother; they mostly communicate via Messenger. Emily is Celine’s roommate; they mostly communicate via WhatsApp, but also Messenger, Slack and Snapchat.


Celine loves her *group Bitmoji* stickers with Emily, which she can only access from Snapchat. The latest update of her mobile operating system introduced a new feature that allows her to browse functionality and expressive media from her communication apps, and design a gesture shortcut for each. When she heard about this, she picked three of her favorite *group Bitmoji* with Emily and created a gesture shortcut for each (Figure 7.1). Now she can access those stickers with a simple gesture, and use them in the same way as she would in Snapchat. For example, when talking with Emily on Whatsapp about an upcoming deadline, Celine can now perform the  gesture to add the “Life is Hard” Bitmoji to a picture, just like in Snapchat (Figure 7.2).



Figure 7.1: Celine designs three gesture shortcuts for three of her favorite *group Bitmoji* with her roommate Emily. The black dots indicate the beginning of the gesture. The Υ gesture resembles a Y, for the “Y tho” sticker; she likes using this sticker when Emily suggests nonsense ideas. The $?$ resembles a question mark; she sends this sticker instead of explicitly asking Emily to go out for a coffee. The \textcircled{C} gesture looks like an earring Emily lost once; they use the “Life is hard” sticker to call out each other’s complaints about mundane or fake problems.

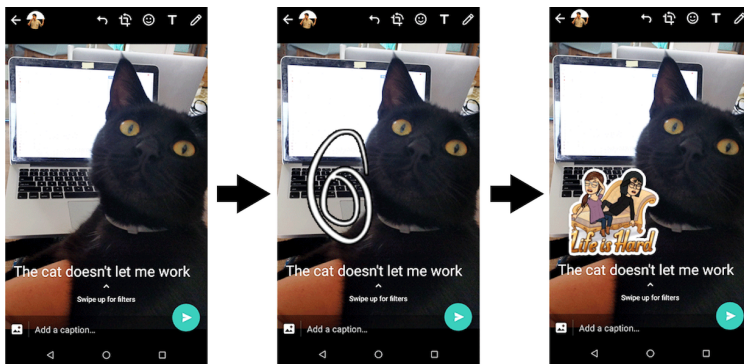


Figure 7.2: Celine is decorating a picture of her cat with WhatsApp to send it to Emily. She performs the \textcircled{C} gesture to invoke a sticker that is not available on WhatsApp. The “Life is Hard” sticker with Emily appears on the picture.

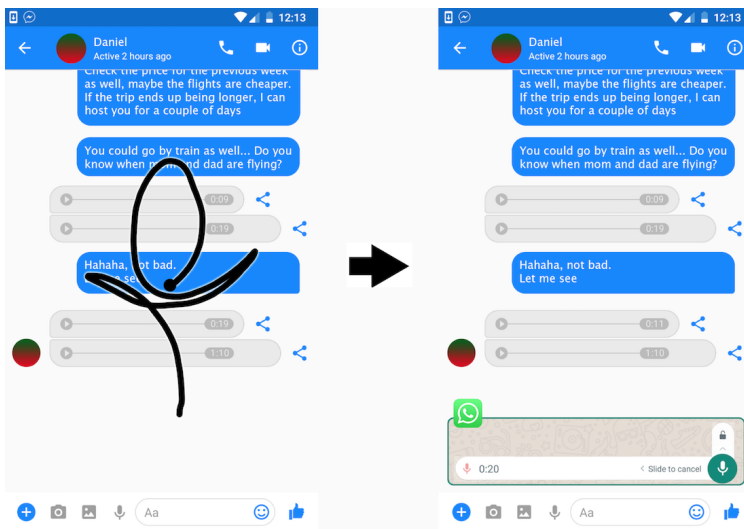


Figure 7.3: Left: Celine performs a “microphone” gesture on top of her conversation with Daniel on Messenger. Right: A WhatsApp voice-recording widget appears within Messenger, and she starts recording her voice message.

Celine and Daniel communicate frequently via voice rather than text messages, but Celine prefers WhatsApp’s interface for recording voice messages. For this, she created a shortcut to WhatsApp’s voice message recording with a gesture resembling a microphone. Now, when her brother texts her on Messenger, she can make the microphone gesture to bring up a WhatsApp voice-recording widget, record a message *à la WhatsApp*, and send it (Figure 7.3). This allows her to use her favorite functionality from WhatsApp without asking her brother to use a different app.

Moreover, Celine sees herself as a cat person, and loves using Messenger’s *Pusheen* stickers⁵ instead of emojis whenever she can. She also keeps a GIF collection in her phone’s media library, full of cat memes to react to messages in conversations. To access these expressive media from any app, she creates gesture shortcuts for her favorite Pusheen stickers and cat GIFs (Figure 7.4). This allows her to perform a Pusheen gesture on a conversation, or on a picture or video to add it to the picture or video, no matter the receiver or the app of the conversation.

Celine also created a gesture for a special “do not disturb” command: performing a cross gesture mutes all notifications from all apps, except for messages coming from Emily. This allows her to reduce distractions during working ours, but still receive messages from her roommate, which might be important or urgent.

This scenario illustrates how creating system-wide, personal gesture shortcuts helps Celine express her intimate bonds with others as well as her cat-loving identity. By associating gestures to functionality and expressive media distributed across her communication apps, she can access her favorite functionality from any app within any conversation, thus preserving her communication places.

However, current mobile operating systems and communication apps do not yet allow the implementation of system-wide, personal gesture shortcuts. This requires work from different angles: **a)** operating systems should enable mechanisms for detecting custom gestures consistently across apps, similar to how they facilitate the detection of swipe, pinch and pan gestures today; **b)** apps should provide open access to their internal commands (i.e. functionality and expressive media) and contacts to enable their use from other apps, **c)** operating systems should offer interfaces for browsing commands and contacts, so that users can associate them to gesture shortcuts; and **d)** operating systems should help users create gestures that do not collide with each

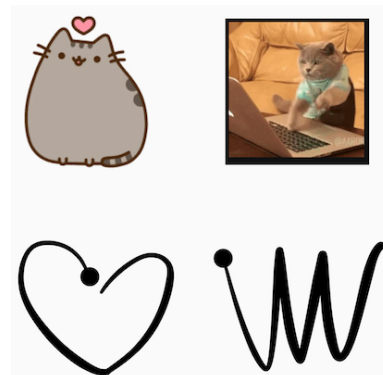


Figure 7.4: Left: a \heartsuit gesture is associated with Messenger’s *Pusheen* sticker. Right: a \mathbb{W} is associated with an animated GIF of a cat typing on a laptop.

⁵ Pusheen is a popular Internet cat character (see left of Figure 7.4). Messenger offers sticker packs featuring Pusheen. See <http://pusheen.com/>.

other, to ensure a reliable recognition.

Given that **a** and **b** highly depend on third-parties, and other apps such as Finger (Figure 7.5) provide interfaces for **c**, in this chapter I focus on how to help users create recognizable gestures. More over, I investigate users' gesture creation strategies to validate the value of personal gesture shortcuts for communication app ecosystems, and to inform the design of future interaction techniques that support the creation of custom gestures.

7.2 Challenges around creating personal gesture shortcuts

Users *can* create keyboard shortcuts, where each combination of keys can be identified unambiguously. It is easy to verify that mapping the same key combination to two different commands will cause a conflict, and other conflicts are not possible. Gesture input, however, is different. Every time the user performs a particular gesture, it varies, sometimes significantly. The system must compare each gesture to a set of predefined templates, using a specified distance metric.

If the performed gesture is “close” to a gesture template and “far away” from all others, the recognizer can safely assume that the close gesture is the intended one, and execute the associated command. However, errors can occur when gesture templates are very similar to each other. For example, a user may intend to draw a straight line, but curve the line, thus matching an “arc” gesture instead. To avoid this, the set of mapped gestures must be distant enough from each other to prevent accidental misidentification.

Adding custom gestures to the system presents several challenges. Each new gesture must be compatible with the existing gesture set, but what makes one gesture different to another may vary from recognizer to recognizer. As gesture recognizers are “black boxes”, users ignore the rules that can help them find new unique gestures, i.e. that do not collide with others. Recognizer features, such as scale, rotation, and direction sensitivity, may result in surprising collisions. For example, for the stock Android recognizer, the following two gestures are the same, but a user would easily perceive them as two distinctive shapes: a circle and a square (Figure 7.6). On the other hand, for

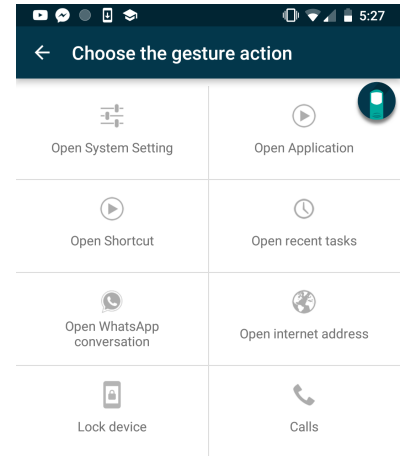


Figure 7.5: The Finger app allows users to define a custom gesture and associate it to a command. The figure shows the menu for navigating the commands users can map to a Finger gesture.

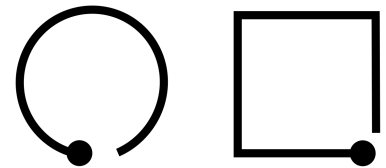


Figure 7.6: The stock Android recognizer interprets a circle and a square as the same gesture.

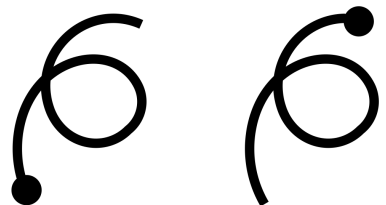


Figure 7.7: The stock Android recognizer, when configured to be *sequence sensitive*, interprets both curl gestures as different ones.

a recognizer that is *sequence sensitive*, the following two gestures are different, while a user might think they are the same (Figure 7.7).

For these reasons, we believe that users could benefit from guides that reveal aspects of the internal “rules” of gesture recognizers as well as the space already taken by existing gestures. After a review of related work, we present two dynamic guides that address these issues.

7.3 Related Work

Designing Gesture Sets

Current touch-based mobile platforms typically support *direct manipulation* gestures, such as pinching and dragging, to manipulate objects on the screen. Such gestures are usually simple, based on the metaphor of interacting with physical or virtual objects. I am interested in extending gestural interaction across users’ app ecosystems with their own, personal unistroke gesture shortcuts.

A key challenge is to design coherent *sets* of gestures that are easy to perform and remember, and are different enough so that the system can easily recognize them. One approach is to use a *Marking Menu* (Kurtenbach and Buxton, 1991), which organizes commands in a radial layout around the cursor. The user invokes a command by moving the cursor in the command’s direction. Although *Marking Menus* are up to three times faster than ordinary pull-down menus, they can only handle a limited number of items, usually between eight and 16. However, the same radial structure can also be applied to hierarchically organized commands, as in *Hierarchical Marking Menus* (Kurtenbach and Buxton, 1993). *Flower* (Bailly et al., 2008), and *Wave* menus (Bailly et al., 2007) also use a radial layout and support hierarchical organization, but use specific schemas to organize the gestures. Appert and Zhai (2009) argue that designers should avoid complex, hard-to-visualize recognition algorithms.

Other researchers have asked users to generate and interpret “natural gestures” (Mauney et al., 2010; Wobbrock et al., 2009) or as part of a participatory design session (Ruiz et al., 2011). Researchers who compare the memorability of free-form gestures find that user-defined gestures are both easier to master (Morris et al., 2010) and easier to

remember (Nacenta et al., 2013). While these studies focus on designing standardized gesture sets for future applications, we are interested in helping users design highly idiosyncratic gestures, which could be redefined over time.

Learning Gesture Sets

Kurtenbach and Buxton (1991) focused on the trade-off between designing for novices, who value discovery and learning, and experts, who value speed and robustness. Their *Marking Menu* offers an elegant solution—experts who already know the appropriate gesture just execute it, whereas novices pause to display a *Marking Menu*. Users see both visual *feedback* indicating the gesture they just performed, and *feedforward* indicating the directions of the remaining possible commands. Over time, novices learn the gestures associated with each command, without requiring conscious effort or a specific intent to learn. The system is forgiving—experts who forget a gesture may also pause and the menu appears to remind them.

Octopocus (Bau and Mackay, 2008) also provides dynamic guides around the cursor when the user pauses, but is not restricted to radial or other structured layouts (see Figure 7.8). It can handle any arbitrarily shaped gesture, so gesture set designers can include more meaningful gestures, e.g. drawing a ? for *help*. We hope to provide similar dynamic guides that support creating, not just learning gestures. Moreover, techniques for learning gestures could be used together with dynamic guides for creating gestures, to remind the user about infrequently used gestures.

Personalizing Gestures

Few commercial systems support user-directed gesture creation. Some tablets use multi-finger gestures to perform pre-specified functions, e.g. dragging four fingers to switch applications, but users can only disable this, not modify it. Previous work studied how to associate or remap gestures to commands. Systems with a small space of pre-existing gestures, such as *Marking Menus*, could be modified via simple configuration dialogs, e.g., *Maya's* (Autodesk Inc., 2016) *Marking Menu Editor*. Alternatively, users could modify the mappings via a configuration window that displays pre-existing gestures and their correspond-

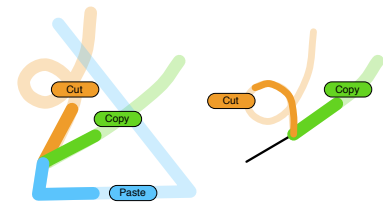


Figure 7.8: The Octopocus technique shows *feedback* of the gesture just performed, and *feedforward* of the directions of the remaining possible commands.

ing commands, as in *Scrybe* (Guay, 2010). However, neither approach helps users add new, personally meaningful gestures to the system.

Long et al. (1999) studied how users design their own gesture sets, and found that users lack understanding of the recognizer and often add strokes that are too similar to previously defined gestures. Horvitz (1999) developed a mixed-initiative system that suggests modifications when a user’s gesture is too similar to an existing gesture. Oh and Findlater (2013) studied personalized gesture-creation, and developed a design space for procedural modification of existing gestures.

In summary, there is a growing interest in allowing users create custom gesture shortcuts for their personal devices. Rather than eliciting standardized gesture sets from users, we are interested in helping users create their own personal gesture shortcuts and better understand their strategies to find memorable gestures.

7.4 Fieldward and Pathward

We created two dynamic guides that display graphical *feedforward* around the user’s finger. Both techniques support interaction between the user and the system’s gesture recognizer— each indicates which future directions will result in recognizable gestures. Users can explore this *negative space* of possible gestures to see if the gestures they find memorable are also easy for the system to recognize.

Both guides use color to indicate the suitability of a gesture or direction. A gradient from blue to red indicates increasing proximity to existing gestures¹. Blue gestures are unique; purple ones are ambiguous. Users may draw any gesture they like— neither dynamic guide constrains the user’s drawing in any way. The techniques simply reveal information about the uniqueness of each gesture from the recognizer’s perspective, and suggest possible completions.

We treat the recognizer as a “black box” with no recognizer-specific heuristics, which lets us support multiple implementations. Both dynamic guides are *recognizer-agnostic*, able to accommodate subsequent improvements to the recognizer, or even changing it entirely, without lowering effectiveness.

¹ The same colors were used for Synaptics Scrybe (Guay, 2010), avoiding problems with the most common types of color blindness.

Pathward Dynamic Guide

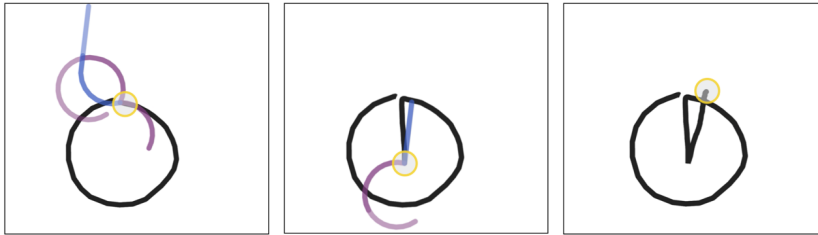


Figure 7.9: *Pathward* technique: a. The user draws a black circle: three possible completions appear at the touch point (yellow circle). b. The user ignores them and draws downward: two possible completions appear. c. The user follows the blue line upward to create a unique gesture.

The *Pathward* technique is inspired by the *Octopocus* (Bau and Mackay, 2008) dynamic guide, but with a twist. *Octopocus* shows the *positive space* of existing gesture-command pairs, and how to continue from the current touch point to correctly execute each gesture. We adapted this approach to display samples of possible recognizable gestures, drawn from the *negative gesture space* revealed by the recognizer. Since the number of possible gestures is prohibitively large, samples are constructed by concatenating a random selection of pre-generated gestural atoms consisting of simple lines and arcs. The recognizer evaluates each sample against the existing gesture set, and the set of samples is optionally pruned based on the recognition scores.

Like *Octopocus*, the *Pathward* technique calculates the offset from the gesture's starting point, and advances the displayed gesture suggestions by an equivalent path length (see Fig.7.9). The remainders of the gesture suggestions are shown originating from the current touch point, indicated by the yellow circle annotations (which are not seen by user). As the user draws, the shapes formed by the drawn gesture and each remaining suggestion are repeatedly passed to the recognizer for evaluation, and the colors of the suggested paths are updated to reflect their evolving recognition scores.

Fieldward Dynamic Guide

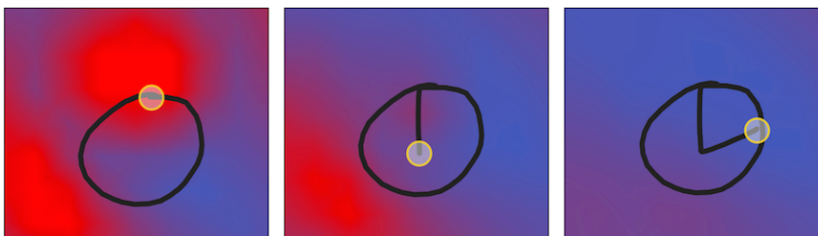


Figure 7.10: *Fieldward* technique: a. The drawn circle ends in a red zone, indicating the gesture is not unique. b. The user draws downward; then c. moves upward, ending in a blue zone to define a unique gesture.

The second dynamic guide offers a more holistic view of the negative gesture space, in the form of a dynamic heatmap. While *Pathward* suggests four pre-evaluated gesture completions, *Fieldward* evenly samples the entire display to complete the gesture with vectors originating from the current touch-point (see Fig.7.10). The score for each completion defines the color at the corresponding point in the background, resulting in a color field that indicates the zones in which a gesture would be unique (blue), ambiguous (purple), or collide with existing gestures (red). An OpenGL shader is used to display even gradients among the sampled points.

The most important difference between *Pathward* and *Fieldward* is the temporal limitation: the *Fieldward* technique can only display one linear step into the future of the current gesture. The *Pathward* technique suggests complete gestures.

7.5 User Study

We compared *Pathward* and *Fieldward* techniques, with a *No Feedforward* control condition on a two-part experiment, to answer the following questions: 1) Can dynamic guides help participants create memorable gestures that are also easy for the system to recognize? 2) Do participants prefer *Fieldward* or *Pathward*? 3) Which strategies do users use to define personal gestures, and are they affected by the choice of technique?

In this chapter, I focus on the qualitative results that answer the last question, and only summarize the quantitative results that answer the other two. For a detailed explanation of insights from pilot studies, the quantitative measures of the experiment and their analysis, please see Appendix A.

Participants

We recruited 27 participants (15 men, 12 women) ages 22-40. Three were left-handed; one right-handed participant preferred using her left hand for the experiment.

Category	Command Title	Command Description
Communication: Family	Call Mom	Dial Mom's phone number
	Group-chat Friends	Open group chat with my best friends
	Text late	Text my best friend that I'm arriving late
	Poke partner	Text my favorite romantic emoji to my partner
	Skype BFF	Start a Skype call with my best friend
	Forward parents	Forward this e-mail to my parents
Communication: Social-Media	Facebook share	Share current webpage on Facebook
	Check #TrafficAlert	Check all live tweets with hashtag TrafficAlert
	Instagram LifeInParis	Upload picture to Instagram with hashtag LifeInParis
	Publish video	Publish current video on Youtube
	Tweet photo	Post to Twitter the last taken photo
	Tag me	Tag myself on current Facebook picture
Context: Media	Dropbox PDF	Save current PDF file to my Dropbox
	Repeat Song	Repeat the last played song
	Next Episode	Watch next episode of the TV show I'm watching
	Translate text	Translate the selected text into my native language
	Song lyrics	Display the lyrics of the current playing song
	Bookmark page	Bookmark this page in my web browser
Context: Location	Show location	Show my current location on a map
	Navigation On	Turn on the voice navigation system on my maps app
	Nearby restaurants	Show nearby restaurants on a map
	Bus schedule	Check the bus time schedule to go from home to work
	Family location	Share my current location with my family
	Route home	Show route to go home on a map
Shortcuts: Apps	Snooze alarm	Stop alarm and re-set it for 10 minutes from now
	Take Selfie	Open the Camera app in selfie mode
	Set timer	Set a 5 minutes timer
	Record voice	Start recording a voice memo
	Play CandyCrush	Play my favorite game, Candy Crush
	Calendar today	Show my agenda for today on my calendar
Shortcuts: OS	Airplane-mode On	Turn Airplane mode On
	Dismiss Notifications	Dismiss all unread notifications
	Battery-saver On	Turn Battery-saver mode on
	Close all	Close all running apps
	Geolocation Off	Disable all geolocation services
	Phone vibrate	Turn the phone ringer to only vibrate

Table 7.1: Trial commands are drawn from six command categories.

Task description

In line with the envisioned scenario around system-wide gesture shortcuts described earlier, we designed a task where users can design their own gestures for associating them with commands for invoking functionality from their app ecosystem. In this task, the user sees a command and designs a gesture for it, e.g. a curl \odot gesture might be assigned to *Send voice message*.

Based on this scenario, we designed an ecologically valid task for the experiment. Table 7.1 shows three categories of common mobile phone commands: *communication*, *context* and *shortcuts*, each with two subcategories. We created six command titles from each subcategory, consisting of a verb-noun combination (e.g. *Tweet photo*), and a longer description (e.g. *Post to Twitter the last taken photo*) The *communication* commands often link functionality with specific contacts, e.g. *Poke partner*. While this thesis focuses on communication, this experiment included other command categories to evaluate the techniques and users' gesture-creation strategies in a more general way (see Fig.7.11a).

Trial description

Each trial begins by presenting the participant with a brief command, e.g. *Call Mom*, initially shown in black (see Fig.7.11b). The participant designs a gesture associated with that command that they can easily remember but that is still recognizable by the system. We want to elicit creative gestures to study users' strategies around the creation of personal gestures. Thus, participants may create any gesture they like, except for letters and numbers.

In the *Pathward* or *Fieldward* conditions, the associated feedforward technique is activated on touch down, and disappears on touch up. No feedforward appears in the *No Feedforward* condition. All three conditions provide recognition feedback by changing the color of the title of the currently drawn command. If the command is too similar to already registered gestures, i.e. with an inter-gesture distance score above 1.5, the command label turns red, otherwise it turns blue. If the participant approves the gesture, they can register it by re-drawing the same gesture two more times, without feedforward. This ensures that the recognizer has three good samples of the gesture. Each recording increments the number in the gray "REGISTER #/3" box (see Fig.7.11).

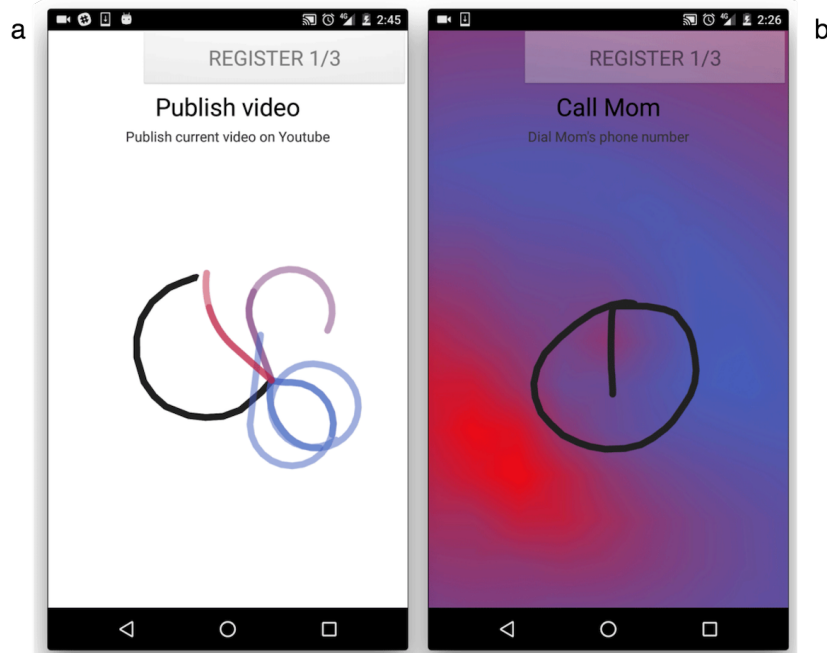


Figure 7.11: Dynamic guides help users create recognizable gestures.

a. *Pathward* suggests four paths: *blue* paths successfully complete the user's *black* gesture, *red* paths collide with existing gestures. b. *Fieldward* displays an interactive heatmap: linear completions ending on *blue* are recognizable, and on *red* cause collisions. *Purple* paths and areas are ambiguous.

If not satisfied with the gesture at this stage, the participant can try a new gesture, as many times as they like. The trial ends when the participant has successfully registered three recognizable samples of the gesture. If the participant cannot successfully register a second or third sample of the gesture, they can press a button to move on to the next command.

Hardware and Software

We used two LG Nexus 5 (4.95" display) and two LG Nexus 5x (5.2" display) smartphones. The *Pathward* and *Fieldward* techniques are implemented in Java and integrated into an application for the Android platform.

Touch events (*down*, *move*, *up*) from the Android system are used to build a vector that describes the current gesture. With each *move* event, gesture completion candidates are appended to the user's gesture. We calculate the distance between each candidate gesture and the existing gestures with the stock Android recognizer, which outputs a score signifying "how recognizable" (i.e. how far from others) is a gesture. Our pilot test results indicated that a score lower than 1.5 is sufficient for adding a new gesture with no collisions. This implementation sup-

ports only single-touch, unistroke gestures.

Experiment overview

Figure 7.12 shows the design of the overall experiment, which is divided into two parts.

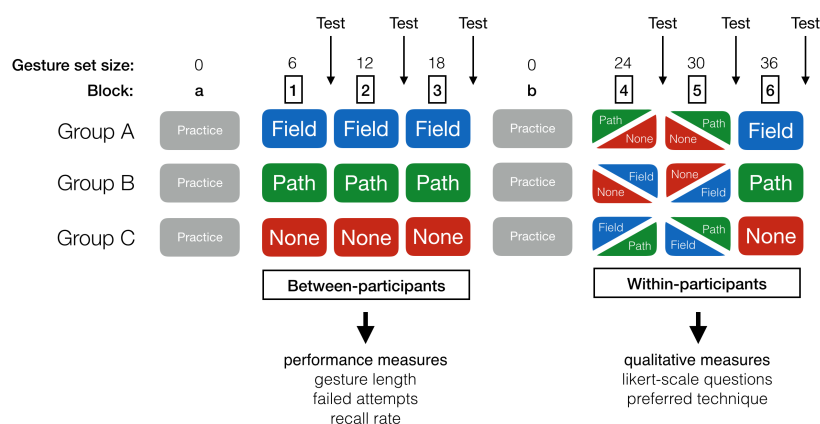


Figure 7.12: Experiment design: Experimental blocks 1-3 are allocated *between* participants. Experimental blocks 4-6 are allocated *within* participants, with blocks 4 and 5 counter-balanced with a latin square across participants. Block 6 repeats the technique in blocks 1-3. Each experimental block includes six trials that produce six new gestures, followed by a recall test.

In Part One, we treat the feedforward technique as a between-participants factor, where we compare performance measures such as gesture length, failed attempts to find a new unique gesture, or recall rate. During our first pilot studies, we tried comparing these performance measures in a within-participants design, but we found confounding effects between the techniques. For example: participants would learn a strategy to find a new unique gesture using Fieldward, and then use the same strategy under the Pathward condition. Thus, we agreed on comparing these performance measures with a between-participants design. Nevertheless, we were also interested in which technique participants preferred, for which we added a second part to the experiment.

In Part Two, where we treat the feedforward technique as a within-participants factor to allow participants to compare the techniques. We collect all the same data for both Parts One and Two, but the analysis for Part Two focuses on the qualitative data we collect through questionnaires and observations, where participants compare the techniques to each other.

Each part consists of a practice block, followed by three experimental blocks. Each block consists of six trials in which the participant

is presented with a command and asked to create a new gesture, followed by a recall test of the six created gestures. Each experimental block contains an example from each of the six categories in Table 7.1, counterbalanced for order within and across blocks. The gesture set size grows block after block. Thus, the task of finding a new unique gesture should get harder as the experiment progresses. By the end of the experiment, each participant has created a total of 36 gestures under experimental conditions (18 for each Part), plus the six created during practice block.

Data Collection

During the practice and experimental blocks, the smartphone runs a screen recorder that saves video of every performed gesture. The experiment log includes timestamps of the start and end of each trial, as well as the touch events for every gesture tried by the participant, the gesture length in pixels, the drawing time in milliseconds, and the list of inter-gesture distance scores from the recognizer. From this raw data we can extract the number of *recognizable* gestures performed, the number of *non-recognizable* gestures performed (FAILED ATTEMPTS), the number of *recognizable* gestures that users performed but decided not to register (USER-REJECTED GESTURES), and the number of successful recall tests (RECALL ACCURACY).

At the end of Part One, participants are given a questionnaire with a list of all 18 commands and are asked to describe the strategies they used to make their gestures memorable. During Part Two, participants see a list of 6 commands at the end of each block. As in Part One, they describe the strategies they used to make their gestures memorable. They then answer four Likert-style questions to assess each technique (reported below). At the end of Part Two, they choose the most helpful technique and explain their choice.

We took observational notes during all sessions and debriefed participants after the final trial, with a particular focus on what the participants liked and disliked about the techniques and about their strategies for creating memorable gestures. We consulted their written strategy descriptions to supplement our understanding when necessary. The salient themes presented in the Qualitative Results section emerged from regular discussions of the data among the research team, with frequent checks back to the source data.

7.6 Results

As this chapter mainly focuses on users' strategies around creating their own gestures, I only present a summary of the quantitative results. For a detailed report on the analysis and results, see Appendix A.

We found that:

- *Fieldward* gets less FAILED ATTEMPTS after registering the 18th gesture (Block 3), i.e. it helps finding recognizable gestures more effectively when the task is the hardest.
- Gestures get significantly longer block after block. Gestures created with *Fieldward* are significantly longer after registering the 12th gesture (Block 2).
- Most participants preferred *Fieldward* (15). Interestingly, only 3 participants preferred *Pathward*, and a third (9) preferred creating gestures with no feedforward at all.

In Part Two, participants were asked to rate four statements on a 5-point Likert scale, from strongly disagree to strongly agree. The statements asked whether the current technique helped them to: A) think of new gestures, B) discover recognizable gestures, C) discover memorable gestures, and D) adapt my memorable gestures to make them recognisable. Three questions resulted in significant differences across technique, based on analysis using a Friedman test:

- *Fieldward* helped to DISCOVER RECOGNIZABLE GESTURES more than *No Feedforward*. *Pathward* was in between, but with no significant differences to the other two.
- *No Feedforward* helped to DISCOVER MEMORABLE GESTURES more than *Pathward*. *Fieldward* was in between, but with no significant differences to the other two.
- *Fieldward* helped to ADAPT MEMORABLE GESTURES TO MAKE THEM RECOGNIZABLE more than *Pathward* and *No Feedforward*.

Question/Technique	<i>Field</i>	<i>Path</i>	<i>None</i>
A) Think of new gestures	4	3	3
B) Discover recognizable gestures	4	3	2
C) Discover memorable gestures	3	2	3
D) Adapt my memorable gestures to make them recognizable	4	3	3

Table 7.2: Medians of the answers from the 5-point Likert-scale questions about how much each technique helped on the listed aspects.

7.7 Gesture Creation Strategies

Participants showed diverse strategies for finding memorable gestures and making them recognizable. The following strategies point not only to improvements to dynamic guides such as *Pathward* and *Fieldward*, but also to opportunities for making everyday phone interaction more personal.

Prioritizing memorability over recognizability

The most common strategy for creating gestures consisted of thinking of a memorable shape and then, if necessary, tweaking it to make it recognizable. For example, in the *Fieldward* condition, P11 drew a smiley face for *Play CandyCrush*, and extended the line representing the face until it showed blue under her thumb. Some users did not mind that the extra segment had no particular meaning – they just expected to memorize it. For example, P4 used *Fieldward* to design a gesture for *Group-chat friends* (see Fig.7.13): “I started with a circle (remembering my circle of friends) and then went to a [blue] corner to make it distinguishable from other gestures.”

By contrast, other participants clearly wanted the extra segment to have meaning. For example, P11 using *Pathward* to design a gesture for *Next Episode* said: *Drawing a play button (triangle) [alone] didn't work. Then I used the [triangle] button followed by a path to add a little circle. In my mind that meant 'next thing' (see Fig.7.14).*

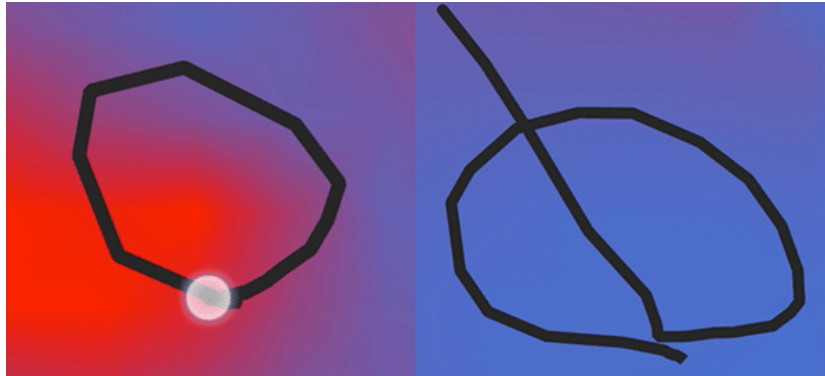


Figure 7.13: P4 first tried a circle for *Group-chat Friends*, and used *Fieldward* to continue the line towards a blue zone .

Participants who had a particular gesture in mind would often extend it to make it recognizable, which explains why the *Fieldward* technique produced the longest gesture lengths. *Fieldward* supports this strategy particularly well: when the task got the hardest (final block), *Fieldward* produced the longest gestures but at the same time had the fewest failed attempts. That *Fieldward* was also the preferred technique, despite the longer gesture lengths, indicates that participants do not necessarily favour shorter gestures. However, this may also be an artifact of the experimental setting. Over time, if users must frequently execute gestures that are long, they might decide to redefine them to make shorter versions.

In the questionnaire, participants ranked *None* highest for helping find memorable gestures, in contrast to *Fieldward* that ranked highest for discovering recognizable gestures. This suggests a tension between efficiently finding recognizable gestures and remembering them: “*Though it takes several times to figure out a recognizable gesture, it would be easier for me to remember the gestures I created independently*” (P21).

Participants appreciated the open-ended, flexible nature of *Fieldward*, which supported thinking of a gesture first, then considering recognizability. They also felt it gave them greater creative control. P8 observed that: “*Fieldward is free enough to help you to create a figure and remember it. (...) It’s hard to follow a path and remember the figure you made.*” P7 noted this limitation with *Pathward*: “*I get that the lines [in Pathward] wanted to help me, but ... but they help you ‘step by step’, and I had a complete gesture idea in my mind.*”

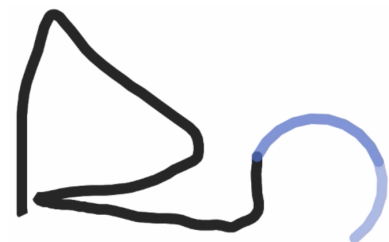


Figure 7.14: P11 added his own meaning (“the next thing”) to the add-on segment suggested by *Pathward* for the *Next Episode* command.

Feedforward techniques act as temporary scaffolding

Although the recognizer was treated as a black box, some participants were able to use the feedforward techniques to discern aspects of how it works. Some participants picked up “tricks” from each technique, such as copying *Pathward*’s tendency to add a curl to a gesture, or back-tracking to retrace the most recent segment, as revealed by *Fieldward*. Some participants treated feedforward as a temporary scaffolding, after which they felt sufficiently confident to no longer need it. For example, P7 used *No Feedforward* for *Check #TrafficAlert*: “I represented a Traffic mess by doing a lot of circles on top of each other. I remembered from using *Fieldward* that it looked like a feasible option.” P10 said: “In the beginning, the *Fieldward* was helpful to understand how to create recognizable gestures, but once I learned, I could do fine with *No Help* [*No Feedforward*].” These participants also indicated a preference for *No Feedforward* at the end of the study.

Reversing, rotating and repeating a gesture to preserve its shape

When participants found a good idea for a memorable gesture and failed to register it, they insisted by trying variations that preserved the original shape and concept of their mnemonics. These variations involved reversing the sequence of the gesture (i.e. drawing the same shape but starting from the end), rotating the shape, and repeating the same shape. For example, P11 first registered “*Kind of a heart*” for *Family location*, and also wanted to use a heart for *Poke partner*: “I couldn’t do the same than for family location, so I drew a heart in the other sense (rotated)”. For *Forward Parents*, she found yet another strategy to register a heart gesture again: “I did a heart twice, as I used a heart in the previous [command] related to family” (Figure 7.15).

Designing gesture grammars

Similar to how P11 consistently registered heart gestures for commands related to close relationships, other participants created their own “gesture grammar” for subsets of related commands. For example, P3 drew a curl gesture to represent sharing (*Forward parents*), and a winky eye to represent a picture (*Take selfie*). He then re-used both gestures as components of other compound commands involving either pictures or sharing something (*Tweet Photo*) (see Fig.7.16). P3 noted: “I

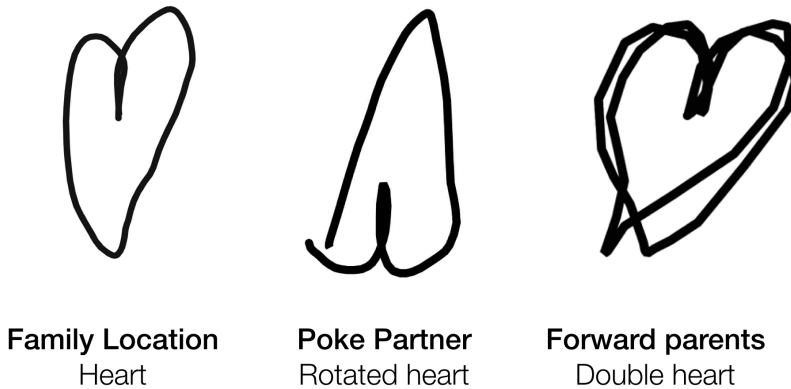


Figure 7.15: P11 varied a heart gesture for three different commands by rotating and duplicating the heart shape.

developed a kind of system of images that makes sense for me, and I tried to keep it consistent during the process."



Figure 7.16: P3 developed his own gesture grammar on the fly. a. *Forward parents*: Envelope, with a curl to indicate sharing. b. *Take selfie*: Winky eye for "picture". c. *Tweet Photo*: Winky eye with a curl gesture for sharing a picture.

Some participants realized they could have created grammars *after* creating gestures for related commands. For example, P7 described her *Publish video* gesture as "a curl going upwards, publish is "to go up"". She then noticed she could not use the same gesture for other publishing-related commands, such as *Facebook share*: "I couldn't use the curl going upwards ("going to the cloud") because it was already used. So I ended up writing an F like the logo.". As the experiment progressed, she detected potential conflicts. She described *Tweet photo* as "a little bird (circle + peak). But it wasn't so good because I would have have problems if I had to draw another gesture to tweet other thing.". This suggests that in realistic settings, users may go back to their gestures and redefine them to create their own grammars over time.

Finding a recognizable gesture first, assigning meaning later

In a few cases, participants selected a gesture suggested by a *Fieldward* technique and then – after the fact – assigned a meaning to the result. For example, P20 described his strategy for *Take Selfie*: *I followed the blue [field] and found a gesture that looked like a four, so that helped me to remember it.* Sometimes participants tried and missed drawing a particular gesture. Rather than discarding it, they would designate a meaning and keep it. For example, P3 wanted to draw a network of three connected circles for *Facebook share*. After drawing it, reminded him of male genitals. He still decided to keep the gesture, since he was not very fond of *Facebook*.

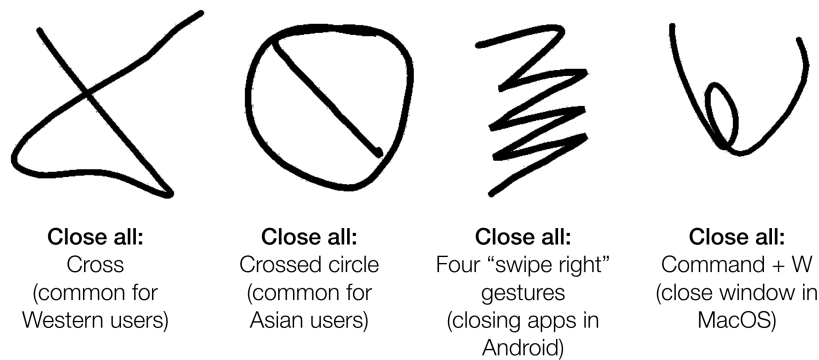


Figure 7.17: Different mnemonics for *Close all* include: a. Cross (common for Western users) b. Crossed circle (common for Asian users) c. Four “swipe right” gestures (Android gesture to close apps) d. “W” (MacOS keyboard shortcut, <CMD>-W, to close windows).

Evoking technology usage habits

Some gestures reflected previous interactions with computers or smartphones. For example, P11 described her gesture for *Dismiss notifications*: *“Tried a check mark. Didn’t work, so I did a double checkmark. In Inbox (Gmail) you do that and an email disappears.”* For *Close all*, P13 registered *“4 swipes to the right as if I was closing the apps on Android”*. For the same command, P7 based her gesture on a related keyboard shortcut: *“In my computer I close apps with <CMD>-W so I did something similar to a W”*.

We also observed several cultural differences (see Fig. 7.17) related to common icons. For example, most Western users represented *Close all* with an ‘X’, whereas most Asian users represented it with a crossed circle.

Representing personal relationships

For commands involving other people, participants created gestures that represent their personal relationships. For example, *Call Mom* evoked a diverse set of gestures depicting something that reminded them of their mothers or their relationship. P24: “When I call mom I’m very happy. I did a circle for my face, and then another curve for my smile.” P5: “It’s a bag because my mom likes bags.” P27: “A bow, that was the Charades code for my Mom when I was young.” P7: “I wanted to write an M, so I did a shape similar to an M. My mom is too complicated to think of something else.”

We observed this with other commands beyond *Call mom*. For *Poke Partner*, P5 registered “the sign * as in the emoji :*”². For *Skype BFF*, P12 explained: “my best friend play violin very well so I choose the pattern of a violin”. For *Groupchat Friends*, P7 described: “I drew the whiskers of a cat, because among my friends we call ourselves “cats””.

² The kiss emoticon :* is treated as a shortcut for the kiss emoji 😘 in some apps, such as Messenger or Slack

7.8 Discussion

The results with respect to the research questions posed earlier can be summarized as follows:

1. “Can dynamic guides help participants create memorable gestures that are also easy for the system to recognize?”

Yes. Both techniques help users discover recognizable gestures, with *Fieldward* helping significantly more than *No Feedforward* and *Pathward* in between. *Fieldward* is significantly more helpful than *Pathward* for adapting memorable gestures to make them recognizable, with *No Feedforward* in between.

2. Do participants prefer *Fieldward* or *Pathward*?

Participants clearly preferred *Fieldward* for most qualitative and quantitative measures. Surprisingly, some participants preferred *No Feedforward* to *Pathward*, which was only preferred when participants sought ideas for new gestures.

3. Which strategies do users use to define personal gestures, and are they

affected by the choice of technique?

Participants prefer to create a gesture they find memorable, and then adapt it to make it recognizable. Dynamic guides can help making memorable gestures recognizable, but may also reveal aspects of how the recognizer works, which users appropriate to find recognizable gesture variations on their own. Strategies for finding memorable gestures evoked common symbols, frequent interactions with other apps and devices, and intimate bonds with contacts.

Fieldward preferred, but No Feedforward also well liked

While *Fieldward* was the most preferred technique, *No Feedforward* also had reasonable support (9/27). Some of the users who preferred *No Feedforward* were those who no longer desired or needed the feedforward scaffolding, as noted above. For others it seemed that they simply preferred complete autonomy over the process. Creating gestures with no support (*No Feedforward*) resulted in more failed trials (initial attempts), but also gestures that were shorter in length. This appears to have been a reasonable trade-off for some users. For example, P5 preferred *No Feedforward* and explained: “While using no help, I adapted my gestures by trying variations on how to draw the same shape. Adapting the gesture with the field helped finding a recognizable gesture similar to my idea faster, but it lead to more complicated gestures.”

Need to support individual differences

We observed many individual differences in strategies and preferences, indicating the lack of a one-size-fits-all solution. However, users do not need one: providing options for different types of dynamic guides would allow users to take advantage of them only when needed. Moreover, the extensive individual differences illustrated in the results show that users create highly personal—and still coherent—gesture sets when using dynamic guides, even if they only help as an initial scaffolding.

Gesture creation in the wild

If users had gesture-creation capabilities on their devices today, we doubt that many would create 42 gestures in one sitting, as imposed by this experiment. Instead, they would probably add gestures as needed, and modify them later if they were difficult to remember. Although we did not see an effect of technique on immediate recall accuracy in our study, more research “in the wild” is needed to determine their effects on gesture recall over time. Similarly, it will be interesting to see which users take advantage of gesture creation. Will they be mostly “power users” (who create their own shortcuts), or will this be adopted by a broader audience? Finally, how important is gesture length in the wild? We expect that users will not notice length for infrequent gestures, but will probably optimize for length for frequent ones. Even so, the ability to create personally meaningful gestures likely outweighs slightly greater lengths. A field study could clarify these issues.

From co-adaptive dynamic guides to reciprocal co-adaptation to support gesture creation strategies

The gesture creation strategies listed above offer a great example of co-adaptation: users learned how to use the techniques, but rather than strictly following the guides, they appropriated the guides to make their own gesture ideas recognizable. Similar to how the user adapts to the dynamic guides and appropriate them, the dynamic guides could as well adapt to the user and further influence their behaviour. The dynamic guides could learn from users’ strategies for making their memorable gestures recognizable, and adapt their suggestions to leverage those strategies. For example, for users that create gesture grammars, the system could learn which segments of the gestures correspond to the “verb” and “noun” of the command, and suggest those segments with *Pathward* when creating new gestures for related commands. *Pathward* could also suggest, after a failed attempt, the same gesture with variations: with an extra segment, with a reversed sequence (i.e. starting from the end), mirrored horizontally or vertically, or even repeating the same stroke twice. Combining a “smarter” *Pathward* with *Fieldward* could also help users further adjust the concrete suggestions from *Pathward* with more open-ended tweaks inspired by *Fieldward*. Moreover, in realistic settings, the system could suggest combinations of frequently used commands and contacts for which the user could create gesture shortcuts.

7.9 Conclusions

In this chapter, I proposed personal gesture shortcuts as a mechanism for **providing access to functionality from other apps** and **user-owned communication tools**, as well as for **supporting relationships across apps**. A design envisionment illustrated how associating custom gestures to expressive media and functionality can help users express their identities and intimate bonds with others while using their favorite features from across their app ecosystems.

I presented Fieldward and Pathward, two techniques for helping users create their own gestures, and studied users' gesture creation strategies. The results show extensive individual differences, resulting in highly personal gesture sets that reflected their culture, technology usage habits and intimate bonds with special contacts. Most participants preferred thinking of memorable gesture ideas on their own (best supported by Fieldward) rather than following concrete suggestions (as in Pathward). This supports the idea of allowing users to create their own gesture shortcuts to their favorite functionality and contacts, rather than imposing a pre-defined gesture set.

While personal gesture may only facilitate *shortcuts* to the myriad of communication tools distributed in a user's app ecosystem, the design envisionment and empirical results in this chapter may encourage vendors of communication apps and mobile operating systems to enable new mechanisms for exposing apps' functionality to others and mapping contacts and functionality to custom, system-wide recognizable gestures.

8

Repurposing Soft Keyboards

In this chapter, I suggest repurposing soft keyboards for **enabling user-owned communication tools, supporting relationships across apps and providing access to functionality from other apps.**

Soft keyboards are services that provide input to the text widgets in an app. Unlike physical keyboards, they can adapt its layout to best suit the type of input that a widget expects, e.g. numeric text fields invoke a numeric keyboard. Soft keyboards exist independently from apps, and can show custom layouts which can display any UI component, not only character keys. For these reasons, I believe that soft keyboards can serve as versatile “trojan horses” that introduce user-owned communication tools into existing apps.

Today, app markets show hundreds of options for installing third-party keyboards, allowing users to change their default keyboard with a custom one. Most of these keyboards focus on offering extended emoji menus, but others add functionality for supporting text input and other means of expression. For example, GBoard (now the default keyboard in some Android devices) allows users to create their own GIFs and send them via any app. It also includes a search widget that allows users to *google* content on the web and send links to conversations (Figure 8.1), and a translation tool, augmenting any communication app with Google Translate’s capabilities (Figure 8.2). SwiftKey recently added the possibility of “pinning” and creating stickers¹, allowing users to define their personal collection of favorite and self-made stickers. Other popular keyboards are Tenor, for browsing GIFs

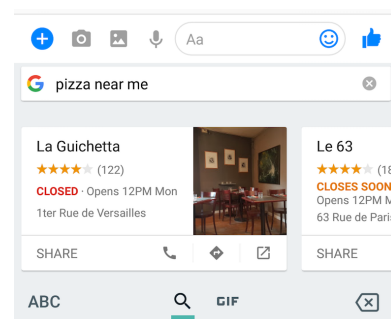


Figure 8.1: GBoard includes a search functionality, showing search results within the keyboard that can be shared into the conversation. The blue buttons and text widget belong to the Messenger app.

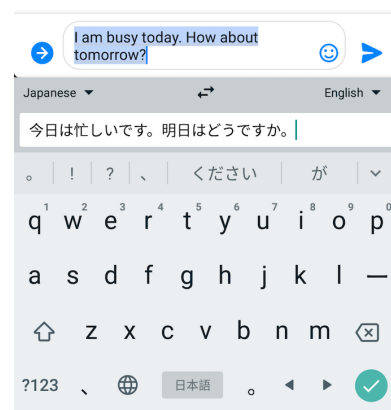


Figure 8.2: GBoard includes a Google Translate widget, which outputs translated text as the user types.

¹ SwiftKey 7 release on March 15th, 2018: <https://blog.swiftkey.com/swiftkey-7-o-big-change-small/>

(Figure 8.4), and Grammarly (Figure 8.3), for correcting the grammar of the typed text. These examples demonstrate the feasibility of using soft keyboards for richer functionality than typing text characters.

Researchers studied patterns of emoji use to inform improvements on emoji keyboards and personalize communication, such as ordering or automatically suggesting emoji according to the user's culture (western/eastern) (Pohl et al., 2017), country (Park et al., 2013), language (Lu et al., 2016) or gender (Chen et al., 2018). On the other hand, our studies suggest that such recommendations should incorporate (and perhaps even prioritize) users' individual habits and preferences—for emoji and other expressive media as well. I argue that beyond the statistically popular emoji for each gender, country or language, users develop personal attachments with specific media, so keyboards and apps should grant users explicit control over the expressive media they want fast access to. Moreover, I propose a new relational approach: rather than only supporting individual expression, soft keyboards could help express intimate bonds with others across apps, showing relevant communication tools according to the recipient of each conversation.

Next, I present a design envisionment for “The Trojan Keyboard”, an extended soft keyboard that supports consistent expression across apps. I analyze the implementation feasibility of “The Trojan Keyboard” through the construction of two prototypes—*The Shared Emoji Toolbox* and *CommandBoard*, and identify barriers imposed by current mobile operating systems to supporting relationships and user-owned communication tools across apps.

8.1 The Trojan Keyboard: a soft keyboard with personal communication toolboxes

What if users could own *communication toolboxes* they can carry with them from app to app? I present the design of *The Trojan Keyboard*, a soft keyboard extended with a personal *communication toolbox* that allows users to mix their own communication tools with those already offered by each app (Figures 8.5 and 8.6). This concept could fundamentally change how users manage their communication online: they could own more control over how they mediate their expression while maintaining the apps that connect them with their contacts.

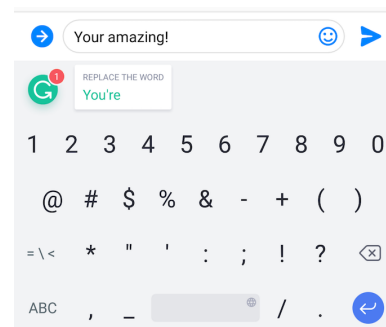


Figure 8.3: Grammarly suggests grammar corrections as the user types. Touching on the suggestions corrects the text in the app's input widget.

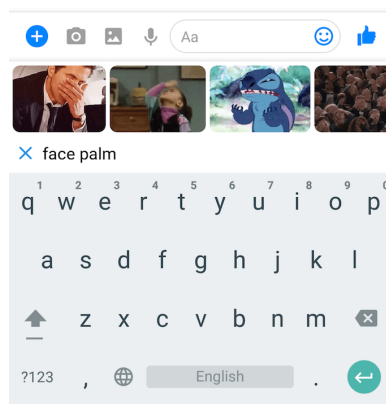


Figure 8.4: Tenor supports searching and browsing GIFs from within the keyboard. Tapping on a GIF sends it to the conversation.

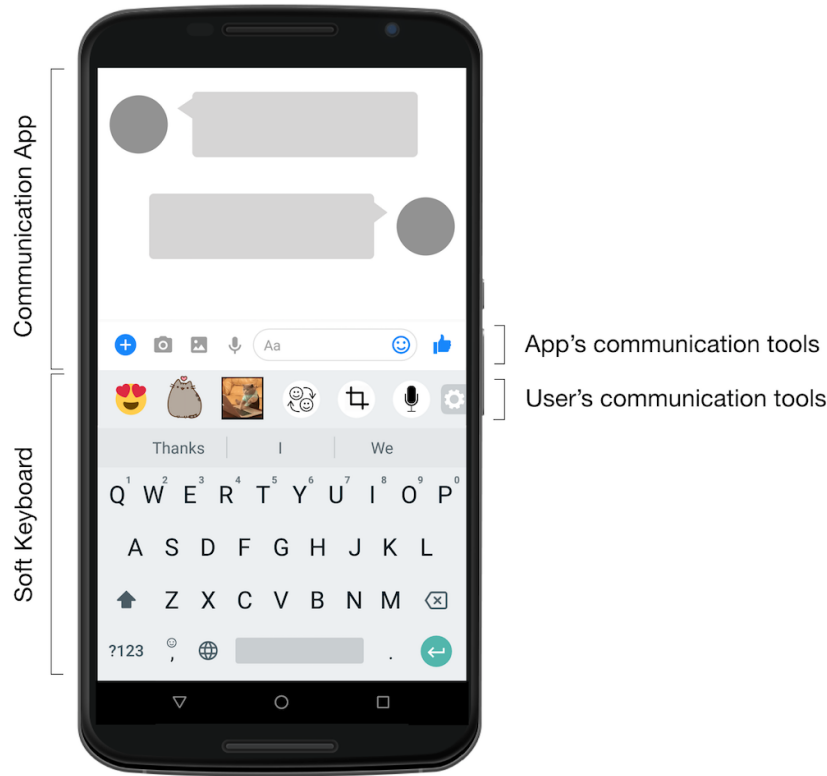


Figure 8.5: The Trojan Keyboard shows a “communication toolbox” above the text input area, where it hosts a custom collection of user-owned communication tools.

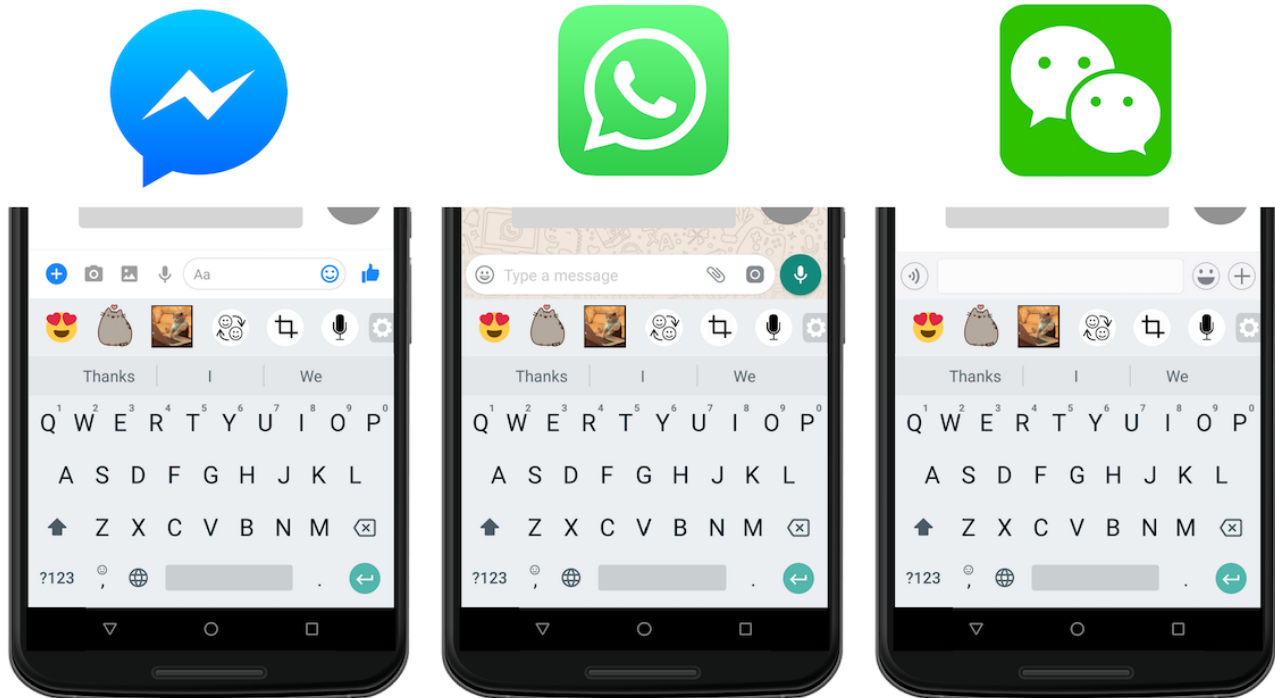


Figure 8.6: The user’s communication toolbox appears while using any app that invokes the keyboard, which allows mixing their own communication tools with those offered by each app.

For example, a user that texts her best friend via Messenger and Snapchat could send her favorite stickers from her "communication toolbox" rather than restricting her expression to Messenger's or Snapchat's stickers. A user that likes sending funny videos to his friends could use his own collection of special effects, filters, and editing tools in any of his communication apps. Moreover, communication toolboxes could be shared, allowing both sender and receiver to select the communication tools that best support their relationship across apps.

Next, I illustrate how *The Trojan Keyboard* could support identity expression, as well as expression of intimate bonds in close relationships. My main goal is to provide inspiration for future technologies that better support consistent expression in communication app ecosystems: first, I present a design vision; then, I discuss the technological and commercial barriers to the realization of my proposed design.

8.2 Supporting identity expression

In Chapter 4 we showed how some users may customize multiple apps and find inventive workarounds to consistently express their identities across apps. For example, one participant used a "Squirtle with sunglasses" as her signature emoji/sticker in Slack, Whatsapp and Telegram. *The Trojan Keyboard* would allow her to add the Squirtle to her personal *communication toolbox* and use it across apps. She could add expressive media and functionality to her communication toolbox by browsing a "communication tools marketplace" similar to current app stores, or by copying them from conversations (Figure 8.7).

The expressive media added to a user's communication toolbox could be used in different ways. For example, a user could simply tap on an icon for sending it as an emoji or long-press for sending it as a sticker (Messenger supports this functionality today, but having this functionality on *The Trojan Keyboard* would extend it to other apps as well). Moreover, long-pressing on a chat bubble could allow *reacting* to the message with user's expressive media, beyond the exclusive options from each app (Figure 8.8). Opening the keyboard when *decorating* a picture could also allow using the user's expressive media as decorations. Figure 8.8 shows a user decorating a picture on Whatsapp with *Pusheen*, a sticker that today is not available in that app.

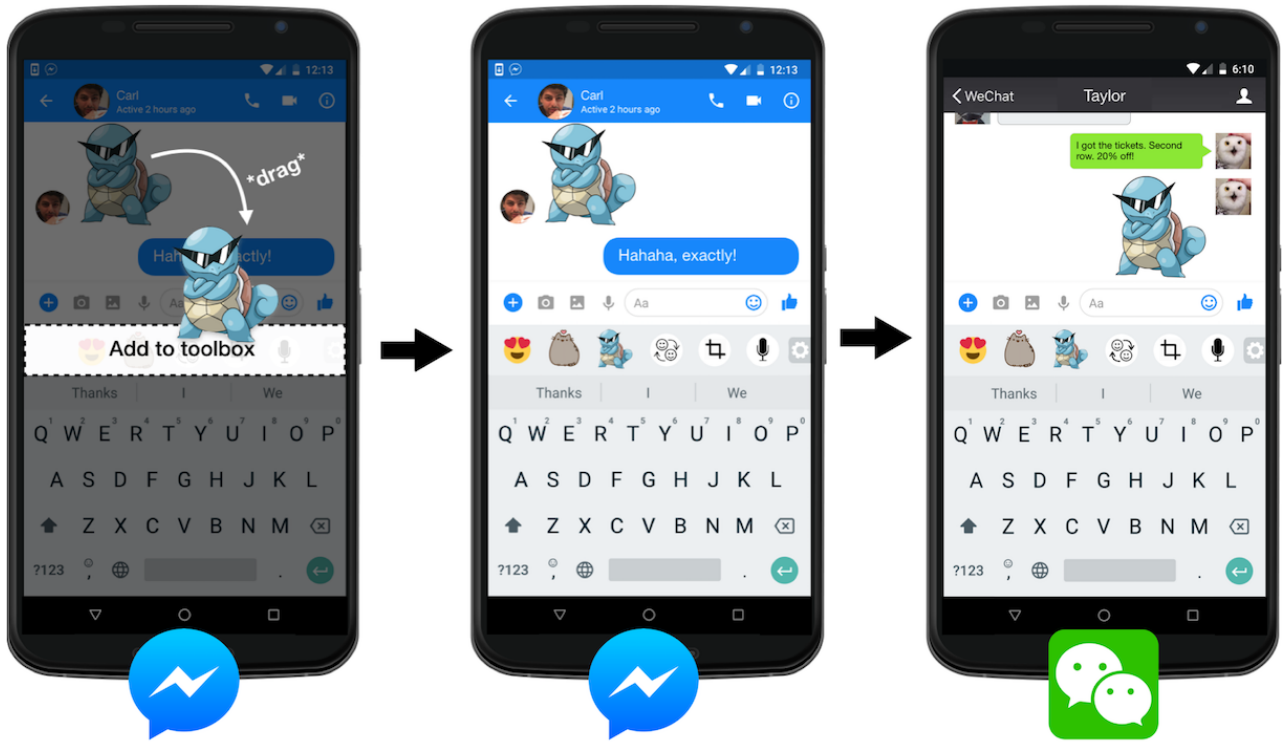


Figure 8.7: A user drags a sticker from a Messenger conversation to their communication toolbox, then sends it via WeChat.

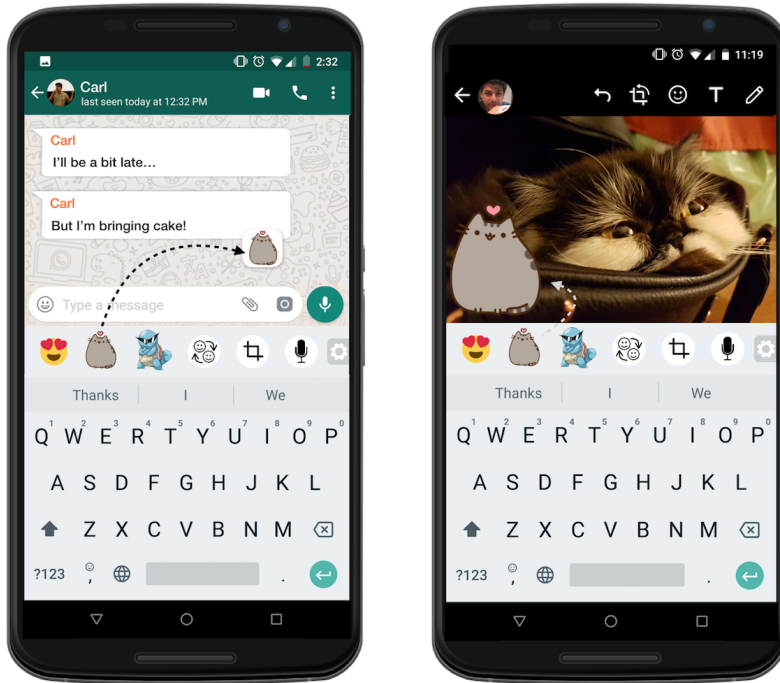


Figure 8.8: Users can react to messages (left) and decorate pictures (right) with their custom expressive media.

Users could also add tools that generate content (e.g. pictures, videos, audios) for their conversations (Figure 8.6), such as a face-swap tool 🔄 for creating funny pictures of two people with their faces swapped, or a crop tool ✂ for cropping pictures before sending them. While some apps might already include these tools, *The Trojan Keyboard* turns them into tools *owned by the user*, which work across the entire app ecosystem. Moreover, apps could expose some of their features as communication tools that can be imported to *The Trojan Keyboard*. For example: WhatsApp could allow users to import 🗣 to their communication toolboxes, so they can compose voice messages *À la WhatsApp* in any of their apps (Figure 8.9).

The customizations done to user-owned tools would remain valid regardless of the app where they are used. For example, in Chapter 4 we mentioned a participant that loved expressing herself with GIFs and stickers of Britney Spears. She could add a “MY GIF” search tool to her communication toolbox and customize it with persistent filters (Figure 8.10). After adding the “funny” and “Britney” filters, her “MY GIF” search tool becomes a “Funny Britney GIF search tool” that works across her apps.

In summary, *The Trojan Keyboard* could support users’ identity expression by enabling a personal communication toolbox where they can add and customize their favorite expressive media and functionality, which they could use consistently across all their apps.

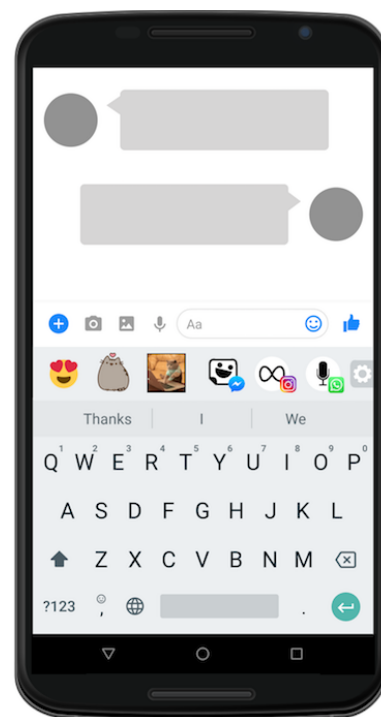


Figure 8.9: Apps could provide access to communication tools that users can import to their communication toolboxes, e.g. WhatsApp’s voice message recording 🗣, Instagram’s boomerang for looping videos 🔄, and Messenger’s stickers 📄.



Figure 8.10: Users can customize their communication tools to use them consistently across their apps. In the example, the user owns a GIF search tool called “MY GIF” that supports persistent filters: everytime the user searches for a GIF, the “funny” and “Britney” filters already apply. The configuration screen allows changing the filters and the GIF repositories to search.

8.3 Supporting the expression of intimate bonds

In Chapter 4, we also showed how users customized to express intimate bonds with a few special contacts. For example, one participant changed Messenger’s “like” button for a bike in a conversation with a friend, because they both love biking and share biking-related activities. Other expressions of intimate bonds involved emojis with special meanings, using group Bitmoji featuring both avatars (sender and receiver), or customizing the chat bubbles with the receiver’s favorite color.

The Trojan Keyboard could have a *shared communication toolbox* besides the user’s private one (Figure 8.11). Pairs or groups of users could add and customize communication tools together, similar to how groups add custom emoji to Slack or change the “like” button in Messenger, but valid for all of their apps. They could also customize the communication toolbox itself, e.g. by changing its background to a custom pattern (Figure 8.11). Both see all customizations to their shared toolbox across their apps whenever they open a conversation together. Users could choose to have shared communication toolboxes with each of their closest contacts, which would appear in all of their conversations, regardless of the app. To support both identity expression and the expression of intimate bonds across apps, *The Trojan Keyboard* would offer a button to switch between user’s private communication tools and the shared ones.

8.4 Extending gesture keyboards

Gesture keyboards let users gesture-type by drawing a line that connects all the letters (a word-gesture). This recognition process is conducted progressively as the user moves her finger: at each touch, the gesture keyboard generates a list of at least four suggested words, and outputs the most likely one.

Besides recognizing words, gesture keyboards could recognize *commands*. After gesture-typing the name of a command, the user could continue the gesture outside the keyboard and finish with a ↵ to trigger the corresponding communication tool. For example, Figure 8.12 shows a user gesture-typing the command “selfie” and executing it

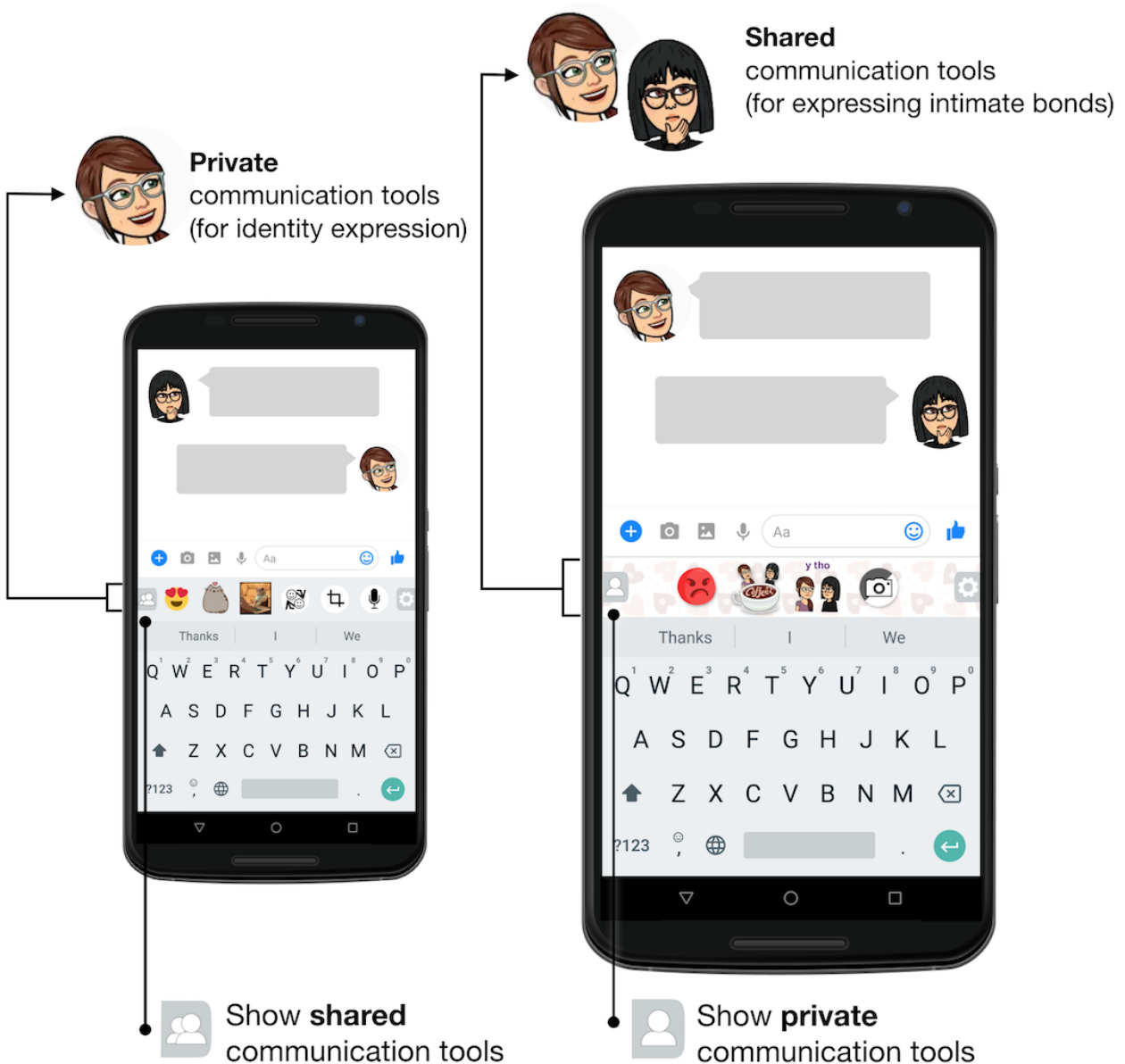


Figure 8.11: Users could switch between a private communication toolbox and a shared communication toolbox. Both sender and receiver add and customize the tools in the shared communication toolbox. In this example, one of the users added two Bitmojis of them together to their shared toolbox, and the other added a pouting face emoji and a “Black & White” camera 📷, which automatically applies a Black & White filter on pictures. The background with hearts is also visible and customizable by both.

with a \lrcorner gesture outside the keyboard, opening a selfie camera tool.

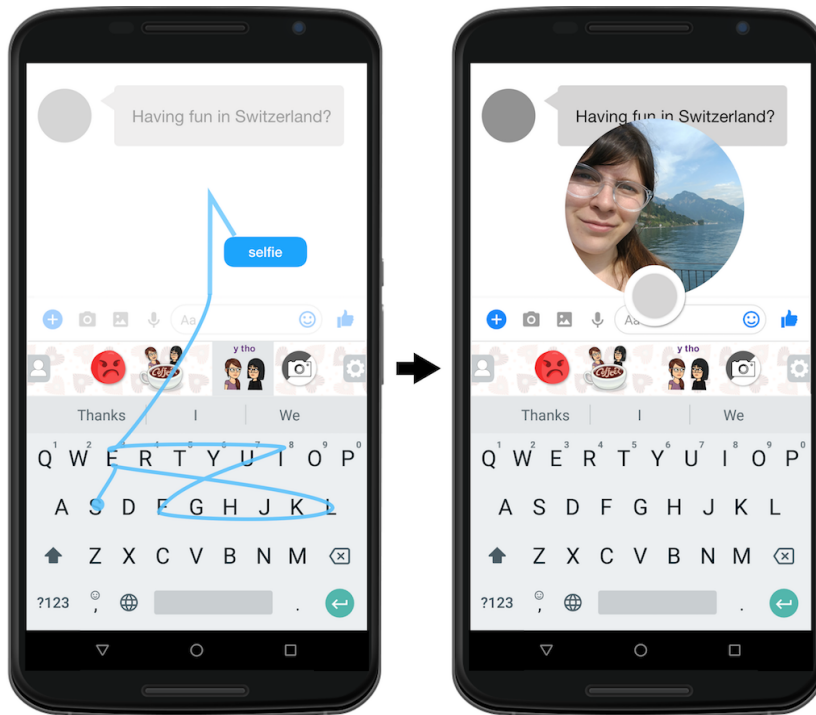


Figure 8.12: Gesture-typing “selfie” and performing the \lrcorner gesture outside the keyboard opens a selfie camera on top of the conversation.

Moreover, users could combine gesture-typed commands with their communication toolbox buttons. For example, gesture-typing the command “sticker” and then sliding through one of their expressive media icons sends it as a sticker (Figure 8.13); gesture-typing “reaction” uses the same icon for reacting to the last received message (Figure 8.14).

8.5 Prototypes: *The Shared Emoji Toolbox* and *CommandBoard*

I worked on two prototypes to help me reflect on the implementation challenges and feasibility of the envisionment above. This section describes the key features and technical details of two prototypes—*The Shared Emoji Toolbox* and *CommandBoard*. I discuss the implementation challenges and barriers to realizing my proposed design in the following section.

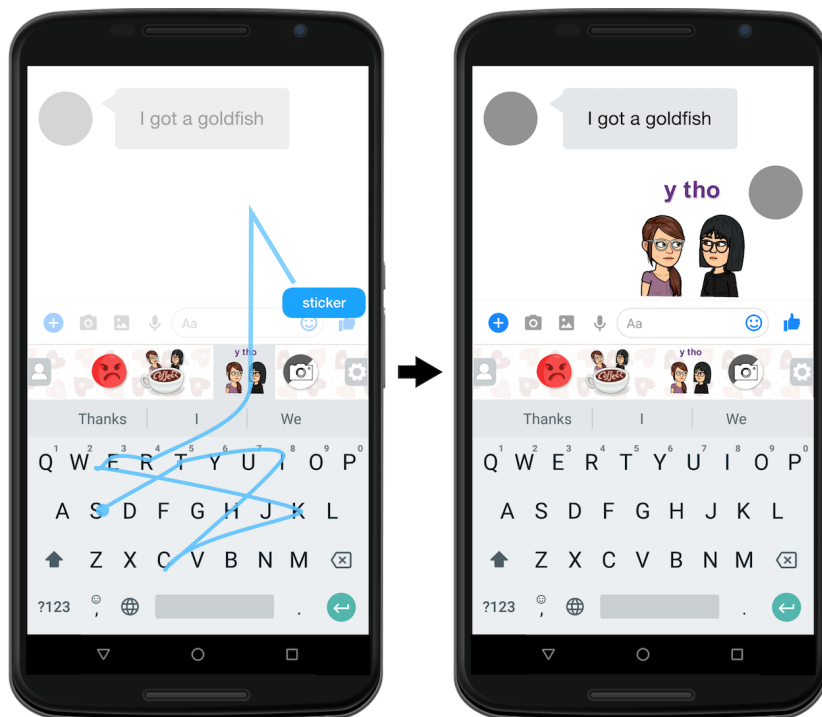


Figure 8.13: Gesture-typing the command “sticker” and then sliding through “y tho” sends it as a sticker.

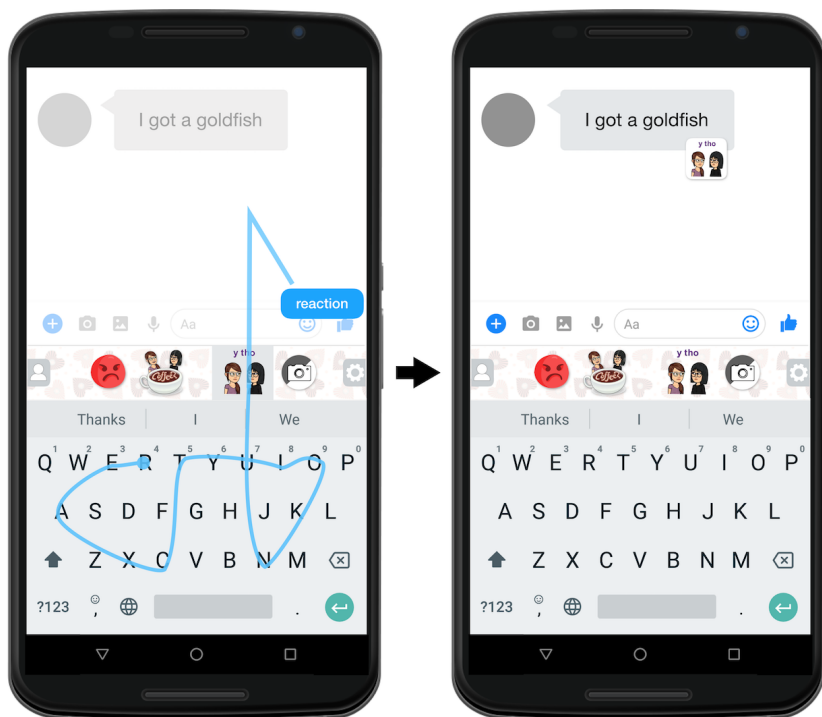


Figure 8.14: Gesture-typing the command “reaction” and then sliding through “y tho” sends it as a reaction to the last received message.

The Shared Emoji Toolbox

The Shared Emoji Toolbox features a simple keyboard with a shared communication toolbox of emojis (Figure 8.15). My goal was to a) demonstrate the feasibility of extending soft keyboards with a simple communication toolbox; b) demonstrate the feasibility of sharing keyboard customizations; and c) facing implementation challenges first-hand, to identify how mobile operating systems and app vendors could contribute to enabling my proposed design. As a first approach, I scoped the shared communication toolbox to a list of emojis, and the shared customizations to the background color of the character keys and the communication toolbox.

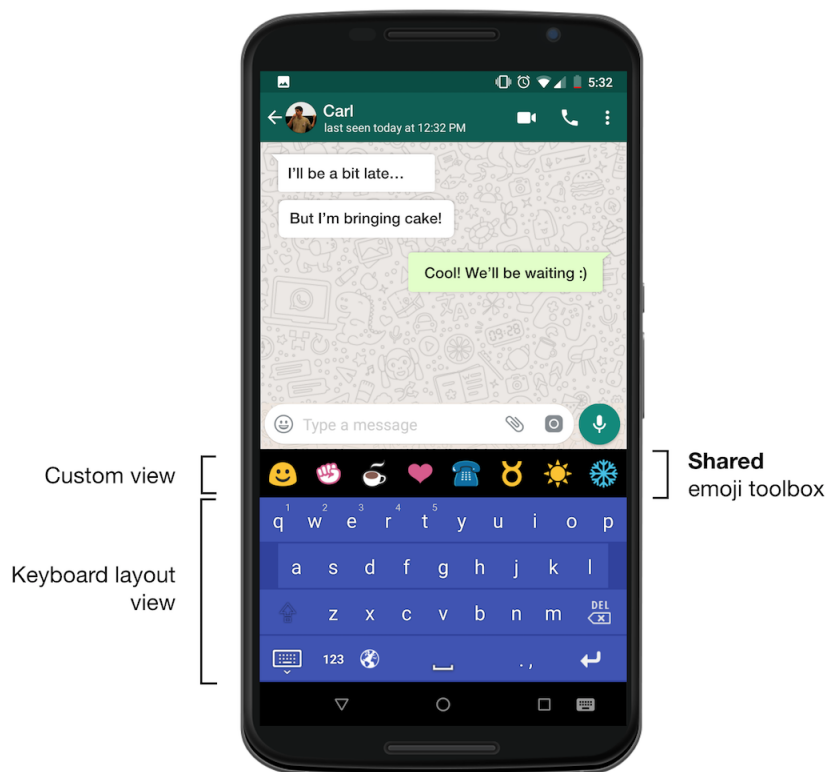


Figure 8.15: *The Shared Emoji Toolbox*. The black banner is a toolbox of emoji shared between the sender and receiver. The toolbox gets the list of shared emojis from a cloud service. Changes to the toolbox on the keyboard update the list of emojis in the cloud service. Thus, anyone that installs the prototype sees the shared toolbox and can update it.

The prototype extends the open-source “Simple Keyboard” for Android². To add *The Shared Emoji Toolbox*, I injected new code in the method that renders the layout of the keyboard. The injected code draws a custom view on top of the keyboard layout, opening the possibility to populate it with any widgets, views and behaviour. This custom view is independent of the layout that structures the keys, so when switching between keyboard layouts (e.g. to show the layout for symbols and numbers), the communication toolbox remains visible.

² “Simple Keyboard” for Android: <https://github.com/rkkr/simple-keyboard>

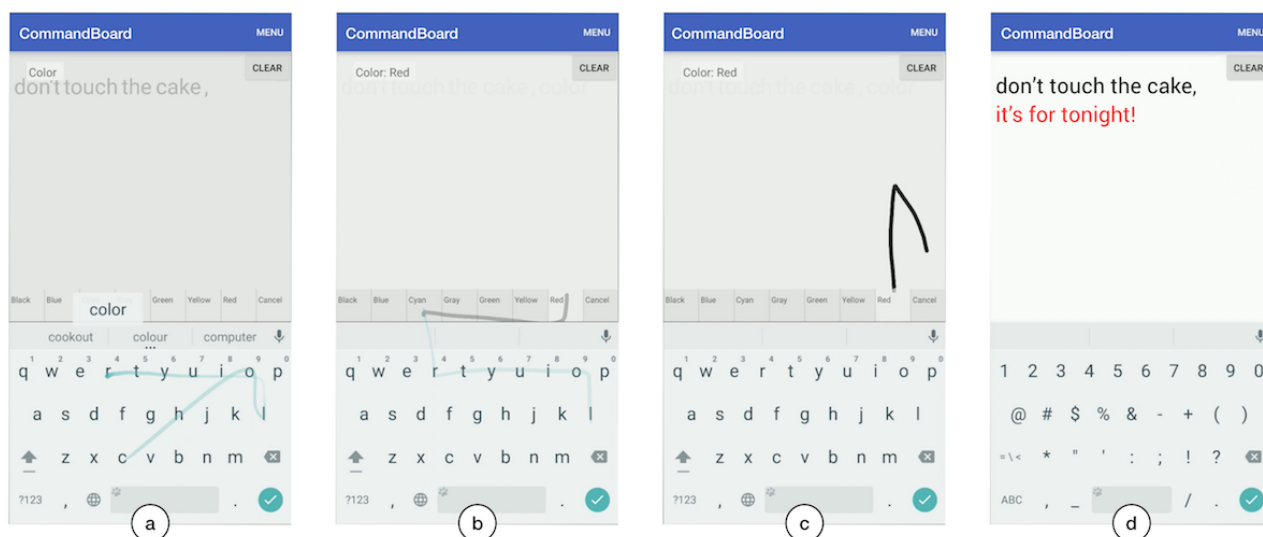
When the keyboard opens (e.g. after tapping on an app’s text field), it makes an HTTP request to a simple web service, which responds a JSON object with a list of emojis in their Unicode form³ and two colors (one for the toolbox background, the other for the character keys). This updates the local configuration of the keyboard, showing the shared emojis and customizations when the keyboard appears.

³Figure 8.15 shows the emojis for `\u263A`, `\u270a`, `\u2615`, `\u2764`, `\u260E`, `\u2649`, `\u2600` and `\u2744`

CommandBoard

CommandBoard prototypes the execution of gesture commands with a gesture keyboard, as envisioned in Figures 8.12, 8.13 and 8.14. We created a simple note-taking app that interacts with a gesture-keyboard provided by Xiaojun Bi⁴, which is based on the SHARK2 gesture keyboard algorithm (Kristensson and Zhai, 2004). Unlike commercial gesture-typing keyboards (e.g. GBoard, SwiftKey), this one and offers an open API for obtaining, for each gesture-typed word, the list of word candidates from the recognition engine, and the position of each key in the keyboard. The main goal of *CommandBoard* was to turn gesture-typing keyboards into artifacts capable of issuing commands (Alvina et al., 2017). Thus, we focused the prototype on text formatting rather than reproducing the full functionality of common communication or text-authoring apps.

⁴Xiaojun Bi’s webpage: <http://www.xiaojunbi.com>



The *CommandBoard* app features a list of commands for styling text (e.g. setting type-face, size, color, and style). When one of the gesture-typed words corresponds to a command, e.g. “color” (Figure 8.16a), it

Figure 8.16: *CommandBoard* prototype. a) The user gesture-types the command “color”. b) Without releasing their finger, the user slides through the parameter “red”. c) The user continues to perform the “execute” \downarrow gesture. d) The user can now type red text.

shows the detected command name and a bar with parameters. Without releasing the finger, the user can continue the gesture by sliding through the desired parameter, e.g. “red” (Figure 8.16b), and perform the “execute” gesture \downarrow (Figure 8.16c) to trigger the command. Then, new text is produced with the set format, e.g. in color red (Figure 8.16d). We call this the *type-and-execute* technique.

Moreover, we implemented an alternative technique for combining typed commands with gesture shortcuts, which we call *inline gesture shortcuts* (Figure 8.17). The user can type a word and, without releasing the finger, continue with a text-formatting gesture above the keyboard. The output is the typed word with an applied format. This way, users can generate words and apply formatting in a single gesture.

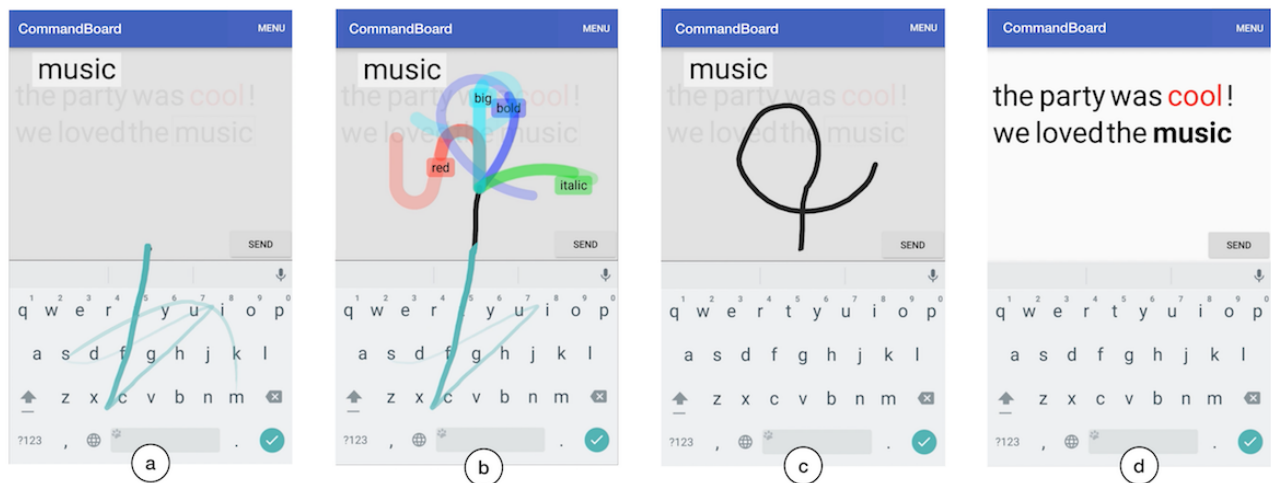


Figure 8.17: *CommandBoard* prototype (*inline gesture shortcuts* technique). a) The user gesture-types the word “music”. b) Without releasing their finger, the user goes above the keyboard and pauses; Octopus suggests four formatting commands. c) The user follows the “bold” gesture. d) The word “music” is printed in bold.

We compared *inline gesture shortcuts* with *markdown formatting* (e.g. typing `*cinema*` for producing **cinema**, or `_cinema_` for producing *cinema*, as featured in WhatsApp, Slack and other apps), and users significantly preferred formatting their text with gestures (see the study design and detailed results in Appendix B). While the study mostly focused on comparing the performance between the two techniques in a text formatting task, a third of the participants suggested using *CommandBoard* for accessing their favorite emojis in communication apps:

I have 5-10 smileys that I always use, so I think it'd be nice if I can use the gesture to get it. Because it's bothersome having to change to another keyboard view (emojis), so if I can do it with the gesture it'd be cool. (P3)

8.6 Barriers to Implementation

Current mobile operating systems prevent the implementation of some key aspects of the envisioned *Trojan Keyboard*. I identified the following barriers by taking diverse approaches: 1) reading the official documentation on custom input methods for Android⁵ and iOS⁶; 2) comparing the behaviour of third-party keyboards available in the market (GBoard, Bitmoji, SwiftKey) across different apps; 3) searching for open APIs for popular communication apps (Messenger, WhatsApp, Telegram, WeChat, iMessage, Snapchat); and 4) the implementation of *The Shared Emoji Toolbox* and *CommandBoard* prototypes, as described in the section above.

⁵Create an input method (Android): <https://developer.android.com/guide/topics/text/creating-input-method>

⁶Create a custom keyboard (iOS): <https://developer.apple.com/design/human-interface-guidelines/ios/extensions/custom-keyboards/>

Lack of soft-keyboard extensions

By prototyping *The Shared Emoji Toolbox*, I demonstrated that mobile operating systems support the creation of soft keyboards with custom layouts, making it feasible to implement a (limited) communication toolbox along with all accessory features for adding, removing, reordering and paginating communication tools⁷. However, it is not possible to extend closed, third-party keyboards with communication toolboxes. We faced this issue while implementing *CommandBoard*: the command-detection and command-execution functionality belongs to the *CommandBoard* app, not the keyboard, because it is not possible to add a gesture-input space to a gesture keyboard without access to its source code.

⁷The creation of a “communication tools marketplace” is independent of the challenge of implementing communication toolboxes within soft keyboards, and outside the scope of this thesis

By being unable to extend third-party keyboards in *runtime*, i.e. without compiling and installing a new keyboard, we shift the problem of distributing functionality among isolated apps to distributing functionality among isolated keyboards. Users would be forced to choose between their favorite keyboard for text input and *The Trojan Keyboard*. Mobile operating systems should either enable (communication) toolboxes as services separate from keyboards, so that users can access their communication tools independent of their soft-keyboard choice. Alternatively, mobile operating systems could enable generic soft-keyboard extensions, so that developers can extend keyboards for various purposes, including communication toolboxes.

Closed APIs

While keyboards can send input to apps, they cannot access the content of a conversation nor interact with the diverse functionality of each app. Communication apps used to offer open APIs to create plugins and alternative clients, but the current trend is to close all APIs and only offer specialized APIs for businesses^{8,9}. Some of my envisioned use cases would require access to the messages in conversations, e.g. for copying a sticker from a conversation to the user's communication toolbox (Figure 8.7), as well as joint development efforts with each app, e.g. for reacting to messages with custom expressive media or decorating pictures (Figure 8.8).

Similarly, apps offer no means to identify the contacts in a conversation. Shared communication toolboxes require identifying the conversation partner in the conversation that is currently open, so that the corresponding toolbox shows up. Nevertheless, this information is not accessible, preventing *The Trojan Keyboard* from supporting relationships across apps.

These are technological barriers that stem from the business models behind mainstream communication apps, which encourage exclusive features as competitive advantages. Mobile operating systems could facilitate standard APIs to enrich the interaction of third-party apps and widgets (such as keyboards) with conversations in apps. Nevertheless, major entrepreneurial efforts would still be required to tempt mainstream apps into shifting from isolated, feature-rich apps to platforms that integrate with user-owned communication tools.

Diverse implementations of the same functionality

Android communication apps share a basic, common API that allows keyboards to send media in an homogeneous way. For example, keyboards like GBoard can send GIFs and stickers to any app by implementing the *Commit Content API*, which allows developers to “*build messaging apps that accept rich content from any keyboard, as well as, keyboards that can send rich content to any app*”¹⁰. Nevertheless, each app is free to implement their own response to the sent content. For example, after sending a sticker from GBoard to Messenger, it directly appears in the conversation. However, when sending it to Telegram, the app opens a “send image” screen, prompting the user for a receiver and an image caption. Telegram does support stickers internally, but reacts

⁸ WhatsApp API for Businesses:
<https://developers.facebook.com/docs/whatsapp/api/reference>

⁹ Messenger API for Businesses:
<https://developers.facebook.com/docs/messenger-platform/>

¹⁰ Commit Content API for Android:
<https://developer.android.com/guide/topics/text/image-keyboard>

differently when receiving a sticker from an external source.

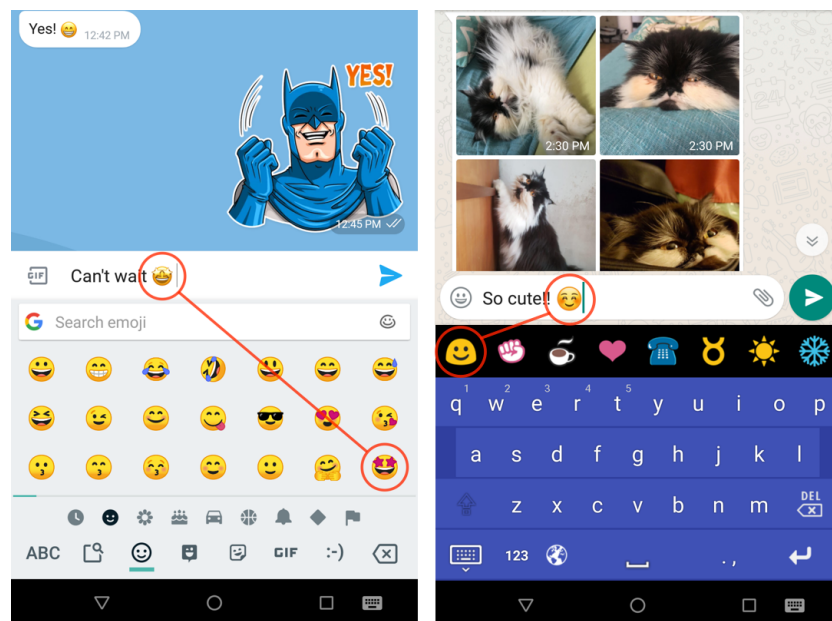


Figure 8.18: Left: the emojis offered by GBoard in Android change when sent to Telegram. Right: the emojis from *The Shared Emoji Toolbox* prototype change when sent to WhatsApp. In both examples, the keyboard and the app use different emoji representations.

Similarly, *The Trojan Keyboard* could allow users to choose their preferred representation of emoji, but once an emoji is sent, each app decides how to render it. Figure 8.18 shows how emojis change when sending them from GBoard to Telegram, and from *The Shared Emoji Toolbox* to WhatsApp, in Android. This issue is rare for iOS users, because communication apps normally use Apple’s emojis instead of their own; nevertheless, this implies that when an iOS user talks to an Android user, they might see different emojis. Apps *have* the means to choose how to render emojis, and should consider involving the user in that choice. Similar to how some delegate the representation of emojis to the mobile operating system’s font, they could also be delegated to the user’s preferred representation. Moreover, similar to how some have their own representation of emojis regardless the mobile operating system (e.g. WhatsApp¹¹ and Slack¹² until late 2017 and early 2018 respectively), they could use different representations of emoji per user or conversation. Once again, these barriers come from design decisions from each app vendor, and solutions are feasible from a technological point of view.

¹¹ <https://blog.emojipedia.org/whatsapp-releases-its-own-emoji-set/>

¹² <https://blog.emojipedia.org/slack-overhauls-emoji-support/>

8.7 Conclusions

In this chapter, I suggested repurposing soft keyboards for **enabling user-owned communication tools, supporting relationships across apps and providing access to functionality from other apps.**

I presented a design envisionment for *The Trojan Keyboard*, a soft keyboard extended with *communication toolboxes* that allow users to carry their favorite communication tools into any of their existing apps. Communication toolboxes may include any expressive media and customizable functionality, accessible through buttons or gesture shortcuts. *The Trojan Keyboard* helps users consistently express their identities across apps with *private* communication toolboxes, as well as expressing intimate bonds with special friends through shared *shared* communication toolboxes. Moreover, by extending *gesture-typing* keyboards, users can refine and extend their communication tools by combining them with typed commands.

While extended soft keyboards present an interesting opportunity to blur the walls between apps and support consistent expression in app ecosystems, there are technological and commercial barriers to the full realization of personal and shared communication toolboxes: lack of support for soft-keyboard extensions, closed APIs and diverse implementations of the same functionality across apps. However, through the implementation of *The Shared Emoji Toolbox* and *CommandBoard*, I demonstrated that current mobile operating systems allow the construction of technology probes that can help us start studying how users perceive and appropriate communication tools that they can carry with them into any of their existing apps.

I hope the envisionment of *The Trojan Keyboard* inspires new ways of designing mobile communication technology, where users own more control over their expression across their entire app ecosystem, and apps become a hybrid space of app- and user-owned communication tools.

9

Conclusions

In this thesis, I argue that we need a better understanding of how *ecosystems of communication apps* affect communication beyond the particularities of each app. I studied how people *adapt* and *adapt to* their communication apps with the goal of better understanding the role of their app ecosystems in how they mediate their relationships and express their identities. I proposed and explored new design directions to shift from building isolated apps to designing ecosystems of complementary app and user-owned communication tools.

Next, I summarize my contributions and reflect on the theoretical foundations that guided this thesis, contextualizing my findings in broader HCI perspectives. Last, I point to future work for addressing the limitations of this thesis and exploring new research directions.

9.1 Contributions

I studied how users manage their contacts across multiple applications and contributed the notion of *communication places*: personal constructs on top of an app with their own membership rules, perceived purpose and emotional connotations which affect the meaning of messages, the appropriateness of behavior, and the relationships with their contacts. Users create communication places to isolate contacts and gain control

over how communication happens in each app, but in doing so, they also restrict the available functionality for communicating with each group of contacts. I provide empirical evidence of how users break their communication places by “bringing outsiders” only to use specific functionality with a particular contact, sometimes causing tension and damaging their relationships.

I also studied how users customize their communication apps and found that they rely on diverse expressive media (e.g. emojis, stickers, GIFs), functionality and customizations to express their identities, their cultures and intimate bonds with others. I argued that users acquire expression habits from each app and try to transfer them to others, but the distinct features that enrich their expression also prevent a consistent communication style across their app ecosystems. I contribute a list of breakdowns, i.e. situations where apps prevented users from expressing themselves as intended, and workarounds, i.e. trade-offs and inventive solutions for expressing themselves consistently across their app ecosystems despite the limitations of each app.

To help users preserve their communication places and express themselves consistently across apps, I propose four directions for shifting from designing isolated apps, to designing for *ecosystems* of communication apps: 1) Allow multiple communication places with the same app by allowing users to separate their contacts in multiple instances of an app; 2) Support relationships across apps by associating functionality to people rather than apps; 3) Provide access to functionality from other apps by exposing apps’ expressive media and functionality system-wide; 4) Enable user-owned communication tools by allowing users to manage communication tools (i.e. expressive media and functionality) that belong to no app in particular, but can be transferred and used between apps.

I explored three of those design directions by repurposing three mechanisms currently available in mobile operating systems: *notifications*, which can be accessed system-wide and can be overlaid on top of the app in use; *gesture commands*, which could execute functionality consistently across apps; and *soft keyboards*, which can carry diverse input functionality from app to app.

First, I demonstrated how notifications can support relationships across apps and enable user-owned communication tools through the design and deployment of *Lifelines*, a communication channel for couples that repurposes notifications as peripheral displays of contextual information. Unlike mainstream communication apps, *Lifelines* con-

nects *two* people and mediates information in a way only they can interpret. Moreover, by treating *Lifelines* as a notification, couples access their contextual information from anywhere in their app ecosystems, perceiving it as a user-owned (or rather, couple-owned) communication tool. The results from a longitudinal study with nine couples contribute evidence of how *Lifelines* provided opportunities for starting conversations, coordinating tasks and finding reassurance without direct contact, and being more understanding with each other.

Second, I demonstrated how gesture commands can support relationships across apps, enable user-owned communication tools and provide access to functionality from other apps through a design envisionment around *personal gesture shortcuts*. I illustrated how users could associate gestures to functionality as well as contacts, supporting the expression of intimate bonds with others across apps. I also illustrated how they could associate gestures to functionality from their existing apps and from user-owned communication tools, supporting their identity expression throughout their app ecosystems. I identified requirements for apps and mobile operating systems to enable this envisionment, and presented *Fieldward* and *Pathward*, two interaction techniques to help users create their own gesture shortcuts. A comparative study of the techniques contributed evidence of extensive individual differences in users' gesture-creation strategies, resulting in highly personal gesture sets that reflected their culture, their past use of technology and emotional links with special contacts. These results further encourage using gestures as a means to help users express their individuality and intimate bonds with others as they execute commands.

Third, I demonstrated how soft keyboards can support relationships across apps, enable user-owned communication tools and access functionality from other apps through a design envisionment presenting *The Trojan Keyboard*. I illustrated how *The Trojan Keyboard* could feature *communication toolboxes* above the text-input area, allowing users to carry their own communication tools from app to app. Users could import communication tools into their toolboxes from existing apps that expose their functionality, from "communication tool marketplaces" or from conversations with their contacts. Communication toolboxes could be private, to support users' identity expression, or shared, to support intimate bonds with others. Moreover, by extending *gesture-typing* keyboards, users could refine and extend their communication tools by combining them with typed commands. I built two prototypes, *The Shared Emoji Toolbox* and *CommandBoard*, and identified the current barriers to implementing this envisionment as well as opportu-

nities for building technology probes for early studies on how mixing user-owned communication tools with existing apps affects mediated communication.

Last, I introduced *story questionnaires*, a new method for collecting lived experiences with online questionnaires by prompting users with stories based on data from previous participants. Story questionnaires helped collecting extra data based on examples from interviews, providing nuances to the previously identified themes.

9.2 Co-adaptation in communication app ecosystems

As introduced in Chapter 2, co-adaptation is a phenomenon in which users *adapt to* the interactive system by learning its capabilities and constraints at the same time that they *adapt* it (or appropriate it) to their own needs (Mackay, 1990b, 2000). The studies in the first part of this thesis show the implications of co-adaptation in mediated communication. Users appropriate the *siloed* nature of apps to separate contacts in *communication places*, but only because they cannot isolate their contacts otherwise. They define each communication place with membership rules, purposes and emotional connotations, which at the same time are influenced by the functionality, contacts and communication patterns associated with the app. Moreover, users adapt to the features of each app which shape their expression habits, but also customize them to better reflect their identities and intimate bonds with others. In summary, users shape their mediated communication practices by adapting to the constraints and capabilities of their apps, but at the same time appropriate them to their own needs.

However, my studies show that users *adapt to* their apps more than they *adapt* them. Even when they customize their expressive media and functionality, they can only do so in terms of what each app allows and restricts, which prevents a consistent expression across their app ecosystems. The four design directions I propose for supporting communication via app ecosystems guide designers towards thinking of more appropriable communication technology, so that users have more control over how they mediate their communication.

In this context, I believe that making technology more appropriable implies offering highly flexible software that users can adapt to

their self-presentation and relational-maintenance needs and desires. Each person and relationship is unique, and technology should help users highlight and nurture their singular communication practices rather than encouraging everybody to communicate the same way—according to which app they use at the moment. In particular, enabling user-owned communication tools opens a wide range of opportunities for making mobile communication highly personal, allowing users to bring their own functionality into existing apps and breaking the constraints that each app imposes today. My design envisionments for *personal gesture shortcuts* and *The Trojan Keyboard* illustrate mechanisms for blurring the walls between apps and helping users define how they want to communicate across their entire app ecosystems. *Lifelines* also illustrates a couple-owned tool that, as a notification, helps users acquire new functionality without having to switch to a new app and sacrifice their communication places. To make mediated communication technology truly appropriable, communication tools must belong to users rather than exclusively managed by apps, so they can adapt them to their own self-presentation and relationship maintenance practices and needs.

9.3 Blurring the walls between isolated communication apps

The first two studies provide evidence of users' struggles around the isolated nature of apps, supporting general critiques to how we design software today.

Apps encapsulate documents and content along with the tools that can operate on them, and offer limited input/output interfaces to interact with the outside world. Tchernavskij (2017) argues that encapsulation creates fewer opportunities for users to cause errors, and allows designers to keep control over how their software is used, but at the same time limits how apps can be combined, encouraging feature-rich, one-size-fits-all designs over complementary tools. Beaudouin-Lafon (2004) advocated for "*designing interaction, not interfaces*" by defining *reinterpretable* interaction architectures that give more control to end users, i.e. that enable systems to be used in ways they were not designed for. However, fourteen years later, the model of rigid, isolated apps still prevails. In the domain of mediated communication, app silos force users to break their communication places when they need

functionality from one app for a contact in another, and they prevent users from expressing their identities and intimate bonds with others in a consistent way across their app ecosystems. Beyond agreeing with previous critiques to the siloed model of apps, this thesis provides empirical evidence of its social implications.

Previous work proposed and implemented alternative interaction architectures to decouple tools from the content they operate with, a key step toward decoupling tools from apps. Beaudouin-Lafon (2000) describes interaction in terms of *instruments* that mediate our interaction with *domain objects*, similar to how we use tools in the real world to interact with physical objects. VIGO (Klokrose and Beaudouin-Lafon, 2009) enables distributed objects and instruments in multi-surface environments. Wulf et al. (2008) propose a component-based software architecture that not only allows development flexibility at *design time* but also at *runtime*, allowing users to tailor their apps by recomposing components on the fly. Webstrates (Klokrose et al., 2015) also shifts flexibility to users, allowing them to recombine and extend shared web interfaces by remixing the links between documents and *instruments* at runtime. Gjerlufsen et al. (2011) proposes *Shared Substance*, a middleware and run-time environment that decouples data from functionality for enabling flexible multi-surface applications. While these examples provide inspiring alternatives to the dominant model of isolated apps, their large-scale implementation implies that we re-build all software from scratch and that thousands of app vendors progressively agree on transitioning to a more flexible and open software model.

While such approaches seek new software architectures to build more reinterpretable software from scratch, my position is to find ways of bridging existing apps with more appropriable tools. By building hybrid ecosystems of apps and user-owned tools, users can gain more control over their interaction *today*, while using the apps they already adopted. This is especially important in the context of communication, where people's communication practices are tightly coupled with the technology that currently mediates their expression. While the gained control and possibilities for appropriation are more limited than what fully reinterpretable architectures would allow, cross-app technology probes based on notifications, gesture commands, and soft keyboards can help us gain early knowledge on the implications of app-independent tools, e.g. how they enrich or hinder self-presentation and relational maintenance practices.

9.4 Communication substrates

Substrates are sets of rules and constraints that structure data and determine how tools may interact with it. For example, Webstrates (Klok-mose et al., 2015) define substrates as “*artifacts that embody content, computation and interaction*” and applies the notion for enabling *instruments* and *documents* that are loosely coupled and can change their behaviour, aesthetics and capabilities depending on how they are combined. For example, while writing a conference paper, one user can embed the paper document in a WYSIWIG-editor instrument, showing the paper as the final print layout and making it capable of handling formatting commands from a Word-like menu. Another user embeds the same paper document in a LaTeX-editor instrument, showing the paper in its LaTeX syntax and making it capable of interpreting LaTeX commands.

Graphical Substrates (Maudet et al., 2017) and Paper Substrates (Garcia et al., 2012) explore the concept by first studying creative practices among graphic designers and music composers, and then building software with domain-specific substrates. For example, Maudet et al. observed that designers establish relationships among the concept, context, content, and layout of their designs to guide their creative work during a project. Inspired by those relationships, they built technology probes that allowed designers to set rules and constraints to their creative tools, such as Linkify, which links content to graphical properties that alter the layout (e.g. the length of titles affects the position of images and subtitles).

This thesis provides an empirical foundation for considering *communication substrates*. Both the creation of communication places and users’ efforts in maintaining a consistent expression across apps point to personal rules and constraints that determine how users want to manage their communication. Beaudouin-Lafon (2017) proposes rethinking apps as a particular type of substrate he calls *environments*, which “*impose a specific organization on their content*”.

Communication apps could be modeled as *communication environments*, substrates where communication tools and conversation objects can be embedded—by designers at design time, and by users at runtime. Environments could feature customizable rules on core aspects of the communication medium, e.g. an “ephemeral environment” à la *Snapchat* would automatically destroy messages after a period of time, unlike a “persistent environment”. Conversations could be manipula-

ble objects that persist the communication between people, and users could move them between environments. For example, two friends may temporally move a conversation from their casual environment to an “encrypted environment” when exchanging sensitive information. Users may also import private and shared communication tools (similar to the private and shared communication toolboxes from Chapter 8) to set the communication style of each environment. Treating conversations as manipulable, app-independent objects would allow users to communicate from different environments, eliminating the required symmetry of use imposed by most apps. Moreover, users could separate their contacts without sacrificing functionality, as well as move them around without causing frictions in their relationships.

Future work could further explore the concept by implementing ecosystems of communication substrates and deploying them among networks of users. In the shorter term, implementing *The Trojan Keyboard*'s private and shared communication toolboxes as communication substrates can provide early insights about how users define and evolve relationships between communication tools and contacts across their app ecosystems.

9.5 Limitations and Future Work

This thesis stands as a first step towards supporting communication via app ecosystems, inviting researchers and practitioners to keep exploring how to grant users greater control over the technology that mediates their expression and relationships.

The design envisionments I proposed in Chapters 7 and 8 require close collaboration with app vendors and mobile operating systems to be fully implemented. Despite current limitations, the prototypes I presented open opportunities for new studies about how people leverage more integrated ecosystems of communication apps. For example, *The Shared Emoji Toolbox* could be extended with other user-owned communication tools (e.g. a custom GIF search box, or photo filters and stickers to decorate pictures); a longitudinal study with pairs of participants in close relationships could then observe how they develop shared communication styles over time. *Fieldward*, *Pathward* and *CommandBoard* could be implemented within an experimental messaging app, where users can link gestures to a set of user-owned communi-

cation tools; a longitudinal study could then investigate how pairs of users evolve the selection of communication tools and gestures over time, as well as what new communication practices they would like to import from or transfer to their existing apps. Hopefully, such insights would encourage app vendors to disable the barriers that prevent a full implementation of the design envisions I present around *The Trojan Keyboard* and *personal gesture shortcuts*.

This thesis describes five technological artifacts: *Lifelines*, *Fieldward*, *Pathward*, *The Shared Emoji Toolbox* and *CommandBoard*, which were built separately but may offer new insights and value when combined. For example, combining *Fieldward* with Octopocus (Bau and Mackay, 2008) and *The Shared Emoji Toolbox* could allow users to create gesture shortcuts to their favorite communication tools and share them with each other (Figure 9.1). Combining *Lifelines* with *CommandBoard* could offer couples the possibility of leaving messages on their lifelines (Figure 9.2), or conversely, sending pieces of their lifelines as messages (Figure 9.3). Studies on these combinations could provide richer insights into how users define and evolve shared communication practices across the apps they use with their closest contacts, and contrast them with how they use feature-rich keyboards (e.g. GBoard, SwiftKey) that support individual expression only.

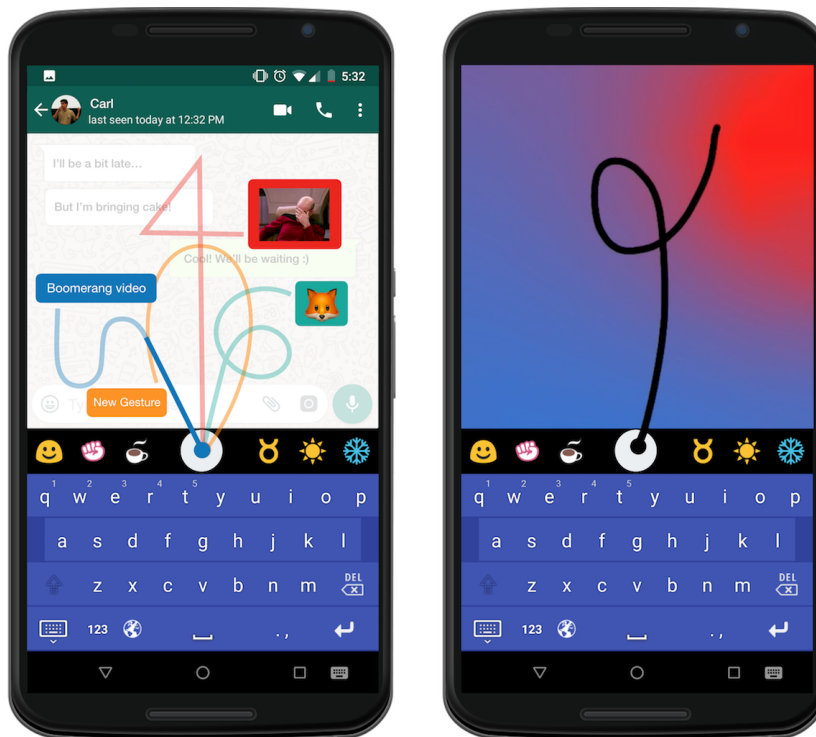


Figure 9.1: Combining *The Shared Emoji Toolbox* with Octopocus and *Fieldward*. The button of the shared gestures tool provides a starting point to perform gestures. Long-pressing it triggers Octopocus (left). The example shows four gestures: the blue one opens a “Boomerang video” tool for recording loop videos that play back and forth; the red one sends a “Facepalm GIF”; the green one sends a Fox emoji, and the orange one opens *Fieldward* (right) for creating a new shared gesture shortcut.

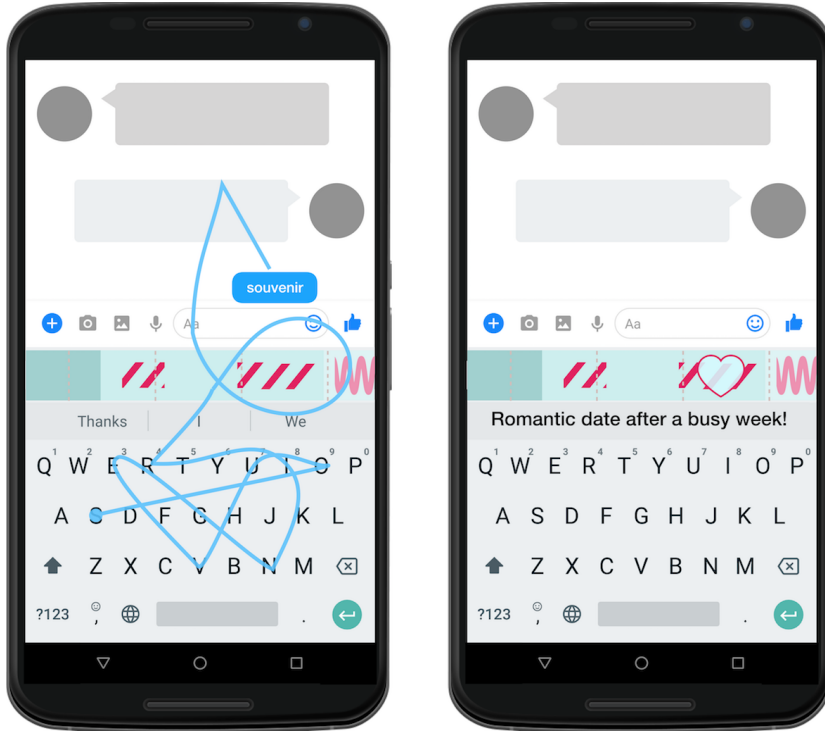


Figure 9.2: Combining *Lifelines* with *CommandBoard* for leaving messages on a lifeline. The keyboard shows the lifeline of the user’s partner. The user gesture-types “souvenir”, then circles a fragment of the lifeline, and executes the command with the \downarrow gesture. A heart mark appears on the lifeline, which the user annotates as “Romantic date after a busy week!”. Touching on the heart reveals the message below the lifeline.

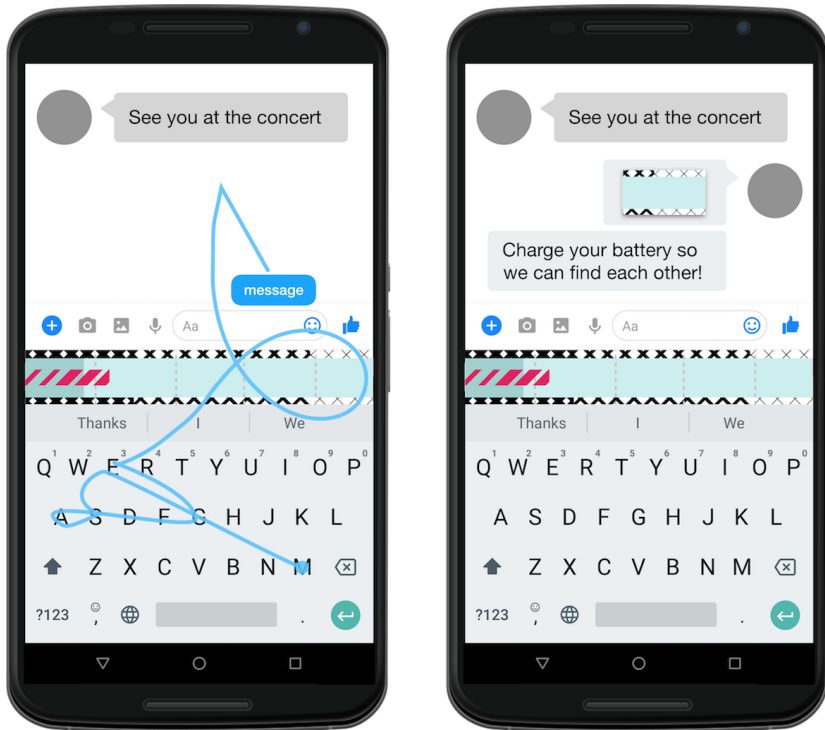


Figure 9.3: Combining *Lifelines* with *CommandBoard* for sending a piece of lifeline as a message. The user gesture-types “message”, then circles a fragment of the lifeline, and executes the command with the \downarrow gesture. The circled fragment is sent as a message.

While this thesis focuses on communication apps that mediate one-on-one or closed group conversations, a great part of mediated communication goes through social media platforms, where users *broadcast* content to larger, sometimes ill-defined groups. Future work could study whether users' efforts to maintain a consistent expression style across apps extend to social media sites. Moreover, my studies, envisionments and artifacts mainly focus on mobile devices, but mediated communication is increasingly ubiquitous and should be considered within users' *artifact ecologies* (Bødker and Klokmoose, 2011). Future work could extend my proposed design directions to support communication beyond ecosystems of *apps*: How could users carry their own communication tools across different *devices*?

This thesis contributes an empirical foundation, design envisionments and technological artifacts that open doors to a diverse range of future research: How does communication change when mixing functionality from different apps? How do users select and customize their own communication tools, and how do these affect their communication across mainstream apps? What kind of communication tools would best leverage the possibility of carrying them from app to app? How can we help users *craft* new communication tools and augment their entire app ecosystems with their own ideas? What are the broader design, technological, commercial and privacy-related implications behind enabling user-owned communication tools that can tightly interact with other apps?

Last, this thesis is centered on mediated communication, but its insights inform the design of software in other domains. Researchers and tool makers may consider how personal collections of isolated apps impact other software-mediated practices (e.g. graphic design, software development, video production). For example: how does the use of one app in an ecosystem influence the use of others? Do users struggle due to app-exclusive functionality? How can we design software so that users can better integrate it with their app ecosystems?

I hope this thesis inspires new research questions about how ecosystems of apps affect communication and software-mediated practices, and that researchers and tool makers continue exploring this new exciting design space.

A

Fieldward vs. Pathward: Quantitative Results

We ran an experiment to compare the *Pathward* and *Fieldward* techniques, with a *No Feedforward* control condition. Although these techniques can work on any touch display, we focus on mobile devices, which offer compelling applications for personalized gestures. Our research questions include:

- *Can dynamic guides help participants create memorable gestures that are also easy for the system to recognize?*
- *Do participants prefer Fieldward or Pathward?*
- *Which strategies do participants use to define personal gestures, and are they affected by the choice of technique?*

This Appendix details the methodological challenges, measures and analysis related to the first two questions. For the results concerning the third question, see Chapter 7.

A.1 Methodological Challenges

The experimental design was challenging, causing us to run a series of pilot studies before settling on the two-part approach described in the Method section below. We ran the first pilot on a tablet, but found that participants used the large screen space to create very long gestures, making them more likely to be unique. We decided to switch to smartphones, since the smaller screen size significantly increased the difficulty of the task, which also increased the need for a dynamic guide. Inspired by situations that require one-handed interaction, such as standing on a crowded bus or carrying groceries, we asked participants to draw each gesture with the thumb of one hand.

Another problem stemmed from our initial use of “invocable” guides, in which the feedforward only appears if the user hesitates. Some pilot test users never invoked feedforward, making it impossible to compare techniques. For the experiment, we decided to turn on the dynamic guide at touch down, to ensure consistency across conditions; in a real application, we would restore the delay so that experts could proceed with no guide.

We designed the dynamic guides so users could choose the desired level of “recognizability” of their gestures, that is, how sloppily the gesture could be performed and still be recognized correctly. In the first pilot study, participants applied different criteria to determine if recognition was “good enough”, and many did not understand what “recognizable” actually meant; for the experiment, we chose a specific recognition threshold.

Early pilot tests were excessively long, up to two hours, and exhausted the participants. So we sought alternatives that would reduce the time to an hour while ensuring sufficient task difficulty. We explored starting with an existing gesture set, to which participants added new gestures. The task was immediately difficult, since each stored gesture increases the likelihood of a collision. Unfortunately, providing users with existing gestures influences their strategies for creating new gestures in subtle and unmeasurable ways. In the experiment, we asked participants to create the gesture set from scratch, starting with six gestures in the initial practice block.

Early pilot studies used a within-participants design, in which users were shown a command name and asked to generate an “easy-to-

remember” gesture, register it and later recall it. We counter-balanced the feedforward techniques for order, and tried various block sizes, ranging from 1 to 12, rotating through each of the three techniques: *Fieldward*, *Pathward* and *No Feedforward*. Unfortunately, we found that the two feedforward techniques were highly confounded: a technique from a previous block sometimes revealed subtle information about gestures already present in the gesture set, and influenced participants in unmeasurable ways. For example, a participant in the *No Feedforward* condition might use a gesture suggestion from the previous *Pathward* condition. This carry-over meant that a within-participant design could not untangle the effect of the current condition from previous conditions.

Another problem was that participants learned at different rates, and faced problems generating gesture ideas at different points in the study. Many participants avoided using dynamic guides when the gesture set was almost empty, since gestures were recognized reliably anyway. They started to use guides only after they “got stuck”, usually after about a dozen gestures. Other participants developed strategies that worked for a few trials, after which they had to come up with a new strategy.

The high variability in learning and gesture-creation strategies, independent of the techniques, led us to consider a between-participants experimental design. Exposing three groups of participants, each to a different technique (*Fieldward*, *Pathward* and *No Feedforward*), provides a “clean” comparison of the characteristics of each technique. However, it does not allow us to determine which technique users might prefer had they been exposed to all of the techniques.

Finally, we considered alternative control conditions, for example displaying the existing gestures. However, providing information about registered gestures only in the control condition would strongly bias the participants; and providing it in all conditions would weaken our ability to measure the effects of the feedforward techniques. Further, since we deal with multiple examples of each user-defined gesture, there is also no single “correct” version to display.

A.2 Experiment Design

Based on the above considerations, we decided on a two-part experimental design. Part One treats the feedforward technique – *Pathward*, *Fieldward*, or *No Feedforward* – as a between-participants factor, whereas Part Two treats them as a within-participants factor. We believe this offers a reasonable compromise: The between-participant blocks let us compare performance measures across techniques without confounding or learning effects; the within-participant blocks let participants try all three techniques when the size of the gesture set is large enough that the task consistently benefits from a dynamic guide. We collect all the same data for both Parts One and Two, but the analysis for Part Two focuses on the qualitative data, where participants compare the techniques to each other.

Part One: Between-Participants Design

Part One uses a between-participants design with one factor: TECHNIQUE (*Path*, *Field*, *None*). Participants are randomly assigned to one of three groups, each with a different feedforward technique.

All participants begin with practice block *a*, which serves as the initial practice session. The experimenter first explains the general goals of the study and then demonstrates how to generate and record a gesture that is recognizable. Participants are asked to hold the phone with only one hand and perform all gestures with their thumb; they are invited to keep their free hand busy (e.g. by holding a pen) to ensure they only use one hand to perform the gestures. The experimenter asks the participant to copy and register four new gestures, the first two with *No Feedforward* and the others with the assigned technique – *Path*, *Field*, *None* – explaining how the technique works. Participants are then asked to create and register two of their own gestures using the assigned feedforward technique. The practice block poses the easiest gesture-creation task, since the gesture set is initially empty, making the first few gestures easily distinguishable. The gesture set in subsequent experimental blocks includes these six preliminary gestures.

Next, each group of participants receives three blocks of the same technique: A is assigned *Fieldward*, B is assigned *Pathward*, and C is the control *No Feedforward* technique. Each participant performs three ex-

perimental blocks of six trials, to create a total of 18 new gestures. At the end of each block, participants are tested on their ability to remember the last six gestures they created in order to prevent them from simply creating random scribbles. Commands are presented in random order and the participant is asked to draw the associated gesture, with no feedforward. The message “OK!” appears if the recognizer matches the gesture with the right command; otherwise “wrong gesture” appears.

At the end of Part One, each participant has added a total of 24 gestures to their gesture set: six from the practice block and 18 from the three experimental blocks. After finishing block 3, participants answer a questionnaire, and may take a break before beginning Part Two.

Part Two: Within-Participants Design

Participants are assigned the two remaining feedforward techniques from Part One in blocks 4 and 5, counter-balanced for order across participants according to a Latin square. Block 6 presents the same technique as in blocks 1-3. This produces an ABCA design that lets participants compare the three techniques at a point when the task has become reliably difficult.

All participants begin with practice block *b*, which introduces the two remaining feedforward techniques. This practice block uses a different gesture set, to prevent confounding with the participant’s existing gesture set. The first three trials of the practice block show one new technique, the last three show the other. The experimenter explains each new technique and asks the participant to use it to copy and register a new gesture. Participants then create two additional gestures, using the same technique. When practice block *b* is finished, the six gestures are deleted and the 24 gestures from Part One are restored.

Each participant performs three experimental blocks of six trials to create a total of 18 additional gestures. At the end of each block, participants are tested on their ability to remember the last six gestures they created. At the end of the experiment, each participant has created a total of 36 gestures under experimental conditions, plus the six created during practice block *a*. After finishing block 6, participants answer a questionnaire.

A.3 Quantitative Performance Measures

Each time the user tries a gesture, we log:

- **GESTURE LENGTH:** The gesture length in pixels.
- **GESTURE DURATION:** The time it took the user to draw the gesture, in milliseconds.
- **RECOGNITION SCORES:** The list of inter-gesture distance scores from the recognizer.

For each stage of each trial (Generate, Register 2, Register 3), we aggregate:

- **SUCCESSFUL ATTEMPTS:** The number of *recognizable* gestures performed.
- **FAILED ATTEMPTS:** The number of *non-recognizable* gestures performed.
- **USER-REJECTED GESTURES:** The number of *recognizable* gestures that users performed but decided not to register.

Results

We collected a total of 2916 experimental trials (27 PARTICIPANT \times 3 TECHNIQUE \times 36 TRIALS). As described earlier, we restrict the statistical analyses to the 1458 trials of between-participants' data from Part One. Similarly, we analyse only the questionnaire responses from Part Two.

Using ANOVA, we first determined that there were no unwanted significant effects from the control variables (command category, command examples). We then ran a 3 \times 3 mixed analysis of variance with factors TECHNIQUE \times BLOCK, followed with Tukey HSD tests for post-hoc comparisons when warranted.

FAILED ATTEMPTS: We use FAILED ATTEMPTS to examine gesture creation over time (see Fig. A.1). We found a main effect of BLOCK ($F_{2,48} = 8.617$, $p = .0006$). Block 1 (mean=2.69) produced signifi-

cantly fewer failed attempts of creating a new recognizable gesture than Block 2 (mean=4.36) and than Block 3 (mean=5.019) (both $p < .05$). In terms of *TECHNIQUE*, across all blocks, participants using the *Field* technique (mean=3.05) failed fewer trials than those using *Path* (mean=3.92) and *None* (mean=5.10). However, the differences were not significant ($F_{2,24} = 1.504, p = .242$).

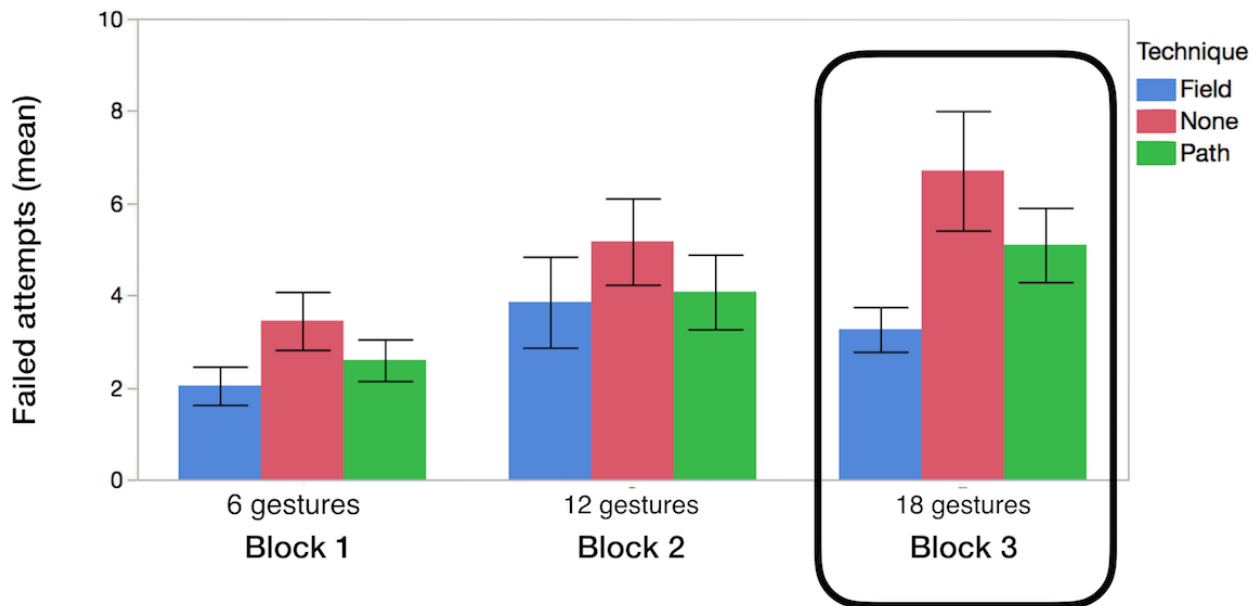


Figure A.1: FAILED ATTEMPTS represents a proxy measure task difficulty. The more gestures the user registers, the harder it is to find a new one. *Fieldward* gets significantly less failed attempts in the last block where the task gets the hardest. Bars indicate 95% confidence intervals.

GESTURE LENGTH: We found a main effect of *BLOCK* ($F_{2,48} = 15.27, p < 0.0001$) revealing that gestures get longer over time. Those created in Block 1 (mean=2219 pixels) are significantly shorter than those in both Blocks 2 (mean=2587 pixels) and 3 (mean=2766 pixels) (both comparisons $p < .05$). A main effect of *TECHNIQUE* ($F_{2,24} = 5.53, p = .01$) reveals that gestures created with *Field* (mean=2949 pixels) are significantly longer than those created with *None* (mean=2338 pixels) and *Path* (mean=2285 pixels) (both $p < .05$). However, an interaction effect between *BLOCK* and *TECHNIQUE* ($F_{4,48} = 4.14, p = .0058$) clarifies that gestures created with *Field* only begin to get significantly longer than those with *None* and *Path* starting in Block 2 (both $p < .05$).

USER-REJECTED GESTURES: Participants using the *Field* technique (mean=1.2) rejected more gestures compared to the *None* (mean=.7) technique. Participants in the *Path* condition fell in between (mean=.9). However, these differences were only at trend level ($F_{2,24} = 3.6, p = .07$). This requires follow-up research.

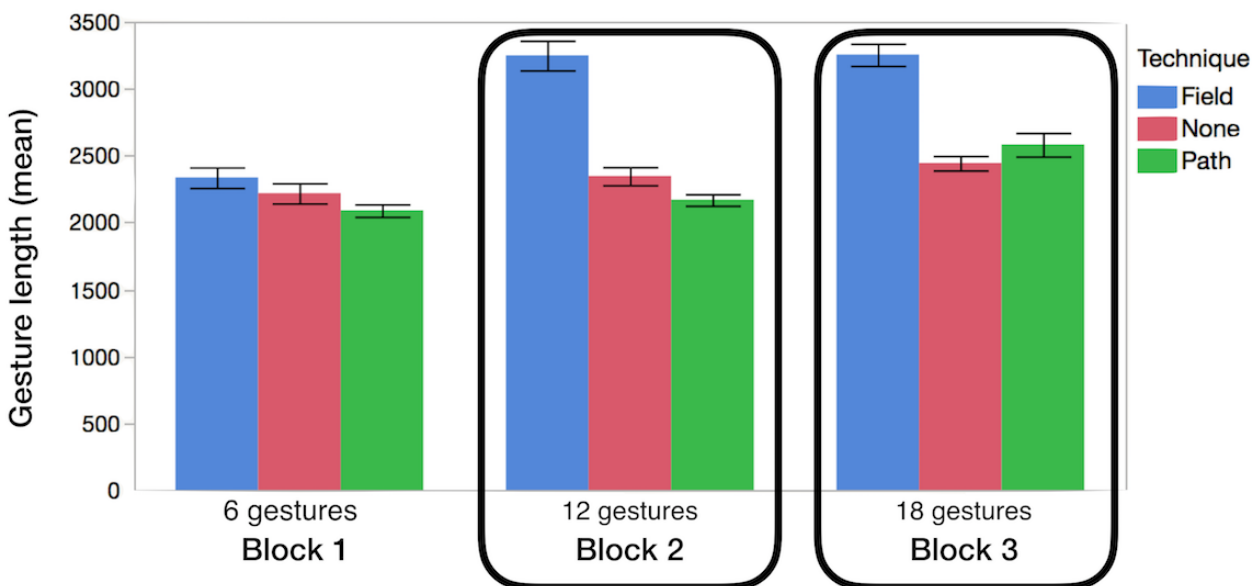


Figure A.2: Gestures get significantly longer block after block. Gestures created with *Fieldward* are significantly longer to those created with *Pathward* or *No Feedforward*. However, an interaction effect between BLOCK and TECHNIQUE clarifies that gestures created with *Fieldward* only begin to get significantly longer than the rest starting in Block 2.

RECALL ACCURACY: We found no significant differences ($F_{2,24} = .26$, $p = .77$) in recall accuracy among *Field* (mean=1.97), *None* (mean=2.08) and *Path* (mean=1.86).

A.4 Quantitative Self-Reported Measures

In Part Two, participants were asked to rate four statements on a 5-point Likert scale, from strongly disagree to strongly agree. The statements asked whether the current technique helped them to: A) think of new gestures, B) discover recognizable gestures, C) discover memorable gestures, and D) adapt my memorable gestures to make them recognisable.

Results

Three questions resulted in significant differences across technique, based on analysis using a Friedman test:

Question/Technique	<i>Field</i>	<i>Path</i>	<i>None</i>
A) Think of new gestures	4	3	3
B) Discover recognizable gestures	4	3	2
C) Discover memorable gestures	3	2	3
D) Adapt my memorable gestures to make them recognizable	4	3	3

Table A.1: Medians of the answers from the 5-point Likert-scale questions about how much each technique helped on the listed aspects.

B. DISCOVER RECOGNIZABLE GESTURES: Participants showed significantly stronger agreement ($p = .019$) for *Field* compared to *None*; *Path* was in between but was not significantly different from the other two: $\chi^2(2) = 9.16, p = .01$.

C. DISCOVER MEMORABLE GESTURES: Participants showed significantly stronger agreement ($p = .016$) for *None* compared to *Path*; *Field* was in between but not significantly different from the other two: $\chi^2(2) = 10.54, p = .005$.

D. ADAPT MEMORABLE GESTURES TO MAKE THEM RECOGNIZABLE: Participants showed significantly stronger agreement for *Field* compared to *Path* ($p = .008$) and to *None* ($p = .005$): $\chi^2(2) = 15.02, p = .001$.

Finally, participants picked the technique they most preferred for creating memorable gestures. *Field* was most popular (15), although many liked *None* (9). Very few preferred *Path* (3).

B

CommandBoard vs. Markdown

B.1 Inline Gesture Shortcuts

Standard mobile devices use icons, buttons and menus to access functionality, because these are easy for novice users to recognize and use. However, many experts prefer the efficiency of command-line interfaces, even though they require learning and subsequent recall of command names and syntax. One of the goals of *CommandBoard* is to bridge the gap between these two approaches, by supporting both recognition- *and* recall-based interaction, with a smooth transition between novice and expert use.

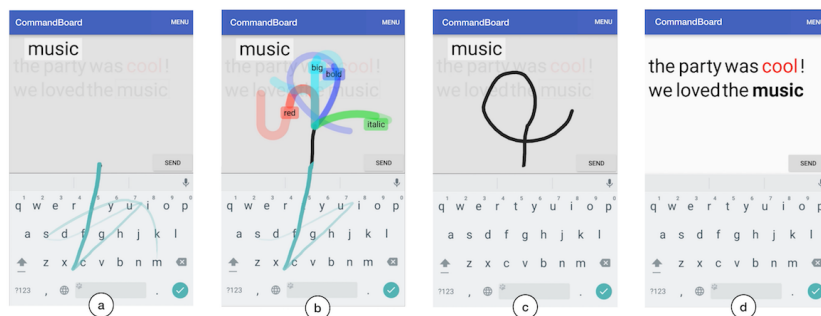


Figure B.1: *CommandBoard's* inline gesture shortcuts. a) The user gesture-types the word “music”. b) Without releasing their finger, the user goes above the keyboard and pauses; Octopocus suggests four formatting commands. c) The user follows the “bold” gesture. d) The word “music” is printed in bold.

CommandBoard's inline gesture shortcuts let users invoke gesture shortcuts to commands that directly affect the typed text. This can be used, for example, to allow users to style their text *as they type*. After typing

the text to be styled, the user slides the finger above the keyboard and executes one of the possible styling commands. Users can benefit from motor memory to recall these gestures. If the user cannot remember a gesture, dwelling above the keyboard brings up the Octopocus dynamic guide, (Figure B.1), and users can follow one of the available options (recognition). As they become experts, they can perform the command gesture directly, without pausing for the guide (recall).

B.2 User Study

We sought an ecologically valid domain for testing *CommandBoard's* ability to support both recognition and recall. We chose the markdown commands available in chat applications such as *WhatsApp* and *Slack*, since users can style their text by typing markdown symbols before and after the text (recall, Figure B.2), or by choosing a formatting option from a context menu (recognition, Figure B.3).

Our research questions include:

1. Are *inline gesture shortcuts* faster and more accurate than text-based *markdown symbols*?
2. Do users prefer *CommandBoard's inline gesture shortcuts*?

We conducted a two-part study, using a within-participants design, to compare *CommandBoard's inline gesture shortcuts* technique to *markdown symbols* (see Figure B.5). Part A is a one-factor experiment that compares speed and accuracy of expert users using these two techniques. Part B is a qualitative study designed to assess participants' preferences as well as incidental learning with respect to each technique. Part B follows Part A, with the same participants, hardware and software.

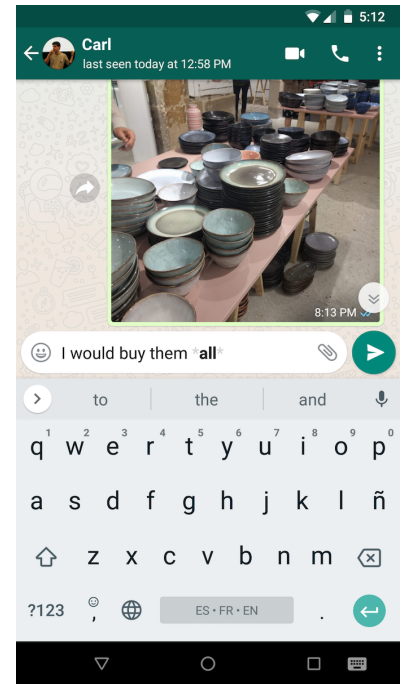


Figure B.2: Markdown in WhatsApp. Users can type symbols around their text to make it bold (e.g. `*all*`), if they recall which symbols produce the desired text styling.

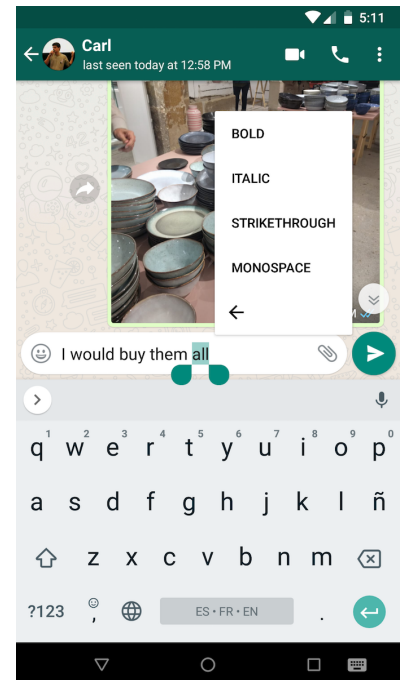


Figure B.3: WhatsApp's text styling options in the contextual menu supports recognition of commands.

B.3 Participants

We recruited 12 right handed participants (4 women, 8 men), aged 23-41. All use mobile phones daily. Two gesture-type daily; the others are non-users. Three sometimes use markdown symbols in existing chat applications; the rest do not.

B.4 Hardware and Software

We used two LG Nexus 5X (5.2" display) smartphones, running Android 7.1.

We implemented *CommandBoard* as an Android application that lets users issue text-styling commands with *inline gesture shortcuts*, using the native Android gesture recognizer. The *inline gesture shortcuts* technique requires the user to draw through the letters of the indicated word on the keyboard. *CommandBoard* recognizes the word, and renders it on the screen. If the user continues the stroke above the keyboard, a semi-transparent overlay appears and the stroke is interpreted as a command gesture. The overlay displays an OctoPocus-like (Bau and Mackay, 2008) dynamic guide indicating the gestures associated with possible styling commands. Lifting the finger applies the recognized gesture-command to the word output and the overlay disappears. Note: We removed OctoPocus' dwell delay in the experiment to avoid confounding time measures. We also implemented the *markdown symbols* technique, which requires the user to type a specified symbol before and after the word to be styled.

Command-Set Design

We created a command set consisting of six text-styling commands: *underline*, *monospace*, *big*, *small*, *outline*, and *gradient color* and mapped them to *inline gesture shortcuts* and *markdown symbols*. The *inline gesture shortcuts* set consists of six gestures chosen from Appert and Zhai (2009), listed in Figure B.4). The *markdown symbols* set consists of six characters chosen from the second row of the symbol keyboard: @, #, \$, %, &, and +. We ensured that none overlap with existing chat sym-

bols from, e.g., *WhatsApp* and *Slack*. Mappings between gestures and markdown symbols are counter-balanced across participants using a Latin square.

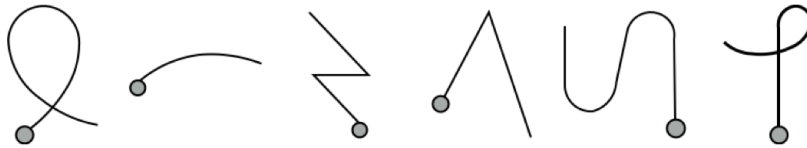


Figure B.4: Gesture set: Grey circles indicate where to begin drawing.

Phrase Set Design

We constructed two sets of 24 three-word phrases drawn from the Oxford Dictionary¹. The middle words are each four-five letters long, and end in 24 different letters of the alphabet (we exclude 'j' and 'q'), to ensure gesture starting points are distributed evenly across the keyboard. We also balanced angles between stroke segments across the sets, to avoid unwanted performance effects (Alvina et al., 2016; Pastel, 2006). Eight words include acute angles, e.g. "menu"; eight include at least one obtuse angle, e.g. the 'agi' in "magic"; and eight include only 0° or 180° angles, e.g. "power".

¹ <https://en.oxforddictionaries.com/>

We used the 24 middle words to create two sets of 24 three-word phrases. We created two phrases around each middle word, using three-to-six letter surrounding words that make sense when read together as a phrase. For example, the first set includes 'play *video* games', and the second set includes 'some *video* clips'. We distributed the first set of 24 phrases across the practice and experimental conditions of the experiment, and distributed the second set across the pre- and post-test conditions of the study. We counter-balanced for order within and across participants using a Latin square.

B.5 Procedure

Figure B.5 shows the study design. Part A consists of four conditions, each comprised of two blocks of six trials, grouped by `TECHNIQUE`. Part B consists of a single recomposition task where users can freely choose the desired technique.

Condition	Part A: Experiment				Part B: Study
	Pre-test	Practice	Experiment	Post-test	Recomposition task
Feedback?	no	yes	yes	no	yes
Gesture	6 trials	6 trials	6 trials x 3	6 trials	12 trials
Markdown	6 trials	6 trials	6 trials x 3	6 trials	(user chooses technique)

Figure B.5: Part A (Experiment): Each condition (Practice, Experiment, Pre-test, and Post-test) includes two blocks of six trials, one per technique, with three replications in the experimental condition. Part B (Study): Participants re-compose 12 of their own messages, with free choice of technique.

Part A: Trial Description

Each trial begins by displaying a three-word phrase, with a styled middle word, e.g. play *video* games. The participant presses START, then retypes the phrase, using the indicated technique to style the middle word. This simulates the process of issuing styling commands during the flow of writing. To simulate “expert” behavior, each trial includes explicit instructions as to how to execute the command, removing the need for recall memory. Participants may preview styling results.

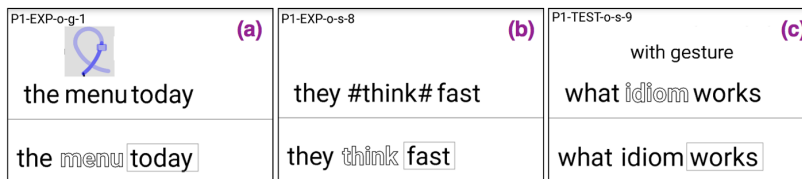


Figure B.6: Each trial presents instructions above the line, and the result below the line. Practice and Experimental conditions present a) *inline gesture shortcuts* to draw, or b) *markdown symbols* to type, to issue the specified styling command. Pre- and Post-Test conditions present c) the styled text to reproduce with the specified technique, with no feedback.

Practice and experimental trials display the correct styling command, either the gesture to draw (Figure B.6a, *inline gesture shortcuts* condition) or the symbols to type (Figure B.6b, *markdown symbols* condition). This simulates expert performance by eliminating errors due to forgetting a gesture shape or markdown symbol. Conditions are separated by short breaks.

Practice Condition

Participants are exposed to two practice blocks, one per TECHNIQUE (*inline gesture shortcuts* and *markdown symbols*). Each block involves typing six three-word phrases, and styling the middle word. Each trial shows which *inline gesture shortcuts* or *markdown symbols* to use. In the *inline gesture shortcuts* condition, the gesture template appears as soon as the participant’s finger leaves the keyboard. Participants can retype phrases as often as they like, until they are comfortable performing the task quickly and reliably. An error message appears if they forget to apply the style or make a typing or styling error. Pressing CLEAN restarts the trial; DONE moves to the next trial.

Experimental Condition

Participants are exposed to two six-trial blocks, one per TECHNIQUE (*inline gesture shortcuts* and *markdown symbols*), for a total of 12 trials. Experimental trials are identical to practice trials, except that participants retype and style each three-word phrase three times (three replications), to provide a stable performance measure.

Pre- and Post-test Conditions

Participants begin with two blocks of six trials, one for each TECHNIQUE (*inline gesture shortcuts* and *markdown symbols*), counter-balanced for order within and across participants. Each trial displays the phrase to be typed including the styled the middle word (see Figure B.6c). Participants reproduce the styled phrase with each technique, with no feedback. This serves as a baseline measure of styling command recall.

The pre- and post-test conditions are identical, but use phrases from the alternate phrase set. The pre-test offers an initial assessment of learning, how much they remember immediately after their first exposure to each technique. The post-test offers a second assessment, based on more extensive practice during the recomposition task.

Part B: Recomposition Task

After completing the Pre-Test condition in Part A, participants are asked to perform a more open-ended set of tasks, in order to assess their overall preferences for each technique. For greater ecological validity, we asked participants to check their smart phones and choose 12 recent messages to retype, avoiding ones they felt were too personal. Participants were free to change the text as they liked. We then asked them to recompose these 12 messages, using either technique to style at least one word. We provided a 'cheat sheet' with the relevant markdown symbols for the *markdown symbols* technique, and displayed a dynamic guide with the relevant gestures for the *inline gesture shortcuts* technique.

B.6 Quantitative Performance Measures

INPUT TIME We measure *INPUT TIME* in seconds for the phrase and each word-output, referred to as: WO_1 , WO_2 , and WO_3 . Note that WO_2 includes inserting the two markdown symbols. This measure allows us to assess the gesture-typing time for both *inline gesture shortcuts* and *markdown symbols*.

GESTURE-TYPING AND COMMAND SELECTION TIME The participant must gesture-type the middle word and style it using *inline gesture shortcuts* or *markdown symbols* (i.e. WO_2). We capture the times spent in each sub-activity. We measure Command Selection Time (*COMMAND TIME*) and Gesture-Typing Time (*TYPING TIME*).

Under the *inline gesture shortcuts* condition, we measure the time spent leaving the keyboard and drawing the gesture (*COMMAND TIME*). If a participant crosses the top border of the keyboard, below the suggestion bar, at $event_k$, then *COMMAND TIME* and *TYPING TIME* are as follows:

$$COMMAND TIME = t(event_N) - t(event_k)$$

$$TYPING TIME = t(event_k) - t(event_0)$$

Under the *markdown symbols*, we measure the time spent writing the symbols before and after the word (*COMMAND TIME*) for WO_2 . Given an input I is a sequence of touch *events*, where $I = \langle event(x, y, t, action)_{0...N} \rangle$, if a participant starts gesture-typing the word at $event_i$ (tagged as down) and lifts her finger at $event_j$ (tagged as up) in WO_2 , then *COMMAND TIME* and *TYPING TIME* are as follows:

$$COMMAND TIME = t(event_i) - t(event_0) + t(event_N) - t(event_j)$$

$$TYPING TIME = t(event_j) - t(event_i)$$

GAP TIME We assess how long participants spend switching from writing a regular word (WO_1) to a styled word (WO_2) and back again (WO_3). Given that an input I is a sequence of touch *events*, $I = \langle event(x, y, t, action)_{0...N} \rangle$ where t is the timestamp, we measure *gap* time between each word-output as follows: $gap(WO_i, WO_{i+1}) = t(WO_{i+1}.event_0) - t(WO_i.event_N)$

ERRORS We count three types of error: typographical errors (*TYPING ERRORS*), incorrect symbols or gestures (*STYLING ERRORS*), or forgetting to style the middle word (*MISSING ERRORS*). Note that *TYPING ERRORS* and *STYLING ERRORS* can occur in the same trial. A trial is considered correct when it has no errors.

B.7 Data Collection

We log all touch events and the recognized word output for each trial. We tag each touch event with one of five actions: *shift*, *tap*, *down*, *move*, and *up*. *tap* involves pressing a key and *shift* involves holding down the keyboard shift key. The remaining actions identify the start (*down*), drawing phase (*move*) and completion (*up*) of a gesture. These measures allow us to compute speed, movement time and errors for each technique.

Participants answer a five-point Likert-style questionnaire to assess their perceived accuracy, speed, ease-of-use, confidence, comfort, and enjoyment of each technique. We also take observational notes and debrief participants, with a particular focus on what the participants liked and disliked about the techniques and their strategies for styling their text.

B.8 Results

We collected a total of 432 experimental trials ($12 \text{ PARTICIPANT} \times 2 \text{ TECHNIQUE} \times 6 \text{ trials} \times 3 \text{ replications}$). We removed one trial (P4) who gave up after repeated typing errors on the third word of one phrase. After determining we had no unwanted significant effects from the word sets, we ran a one-way repeated-measures analysis of variance for factor *TECHNIQUE*, followed by Tukey HSD tests for post-hoc comparisons.

Input Time

The overall *INPUT TIME* (trial completion time) is significantly affected by *TECHNIQUE* ($F_{1,11} = 86.9, p < 0.0001$). This is due primarily to styling the middle word (WO_2), as shown in Figure B.7.

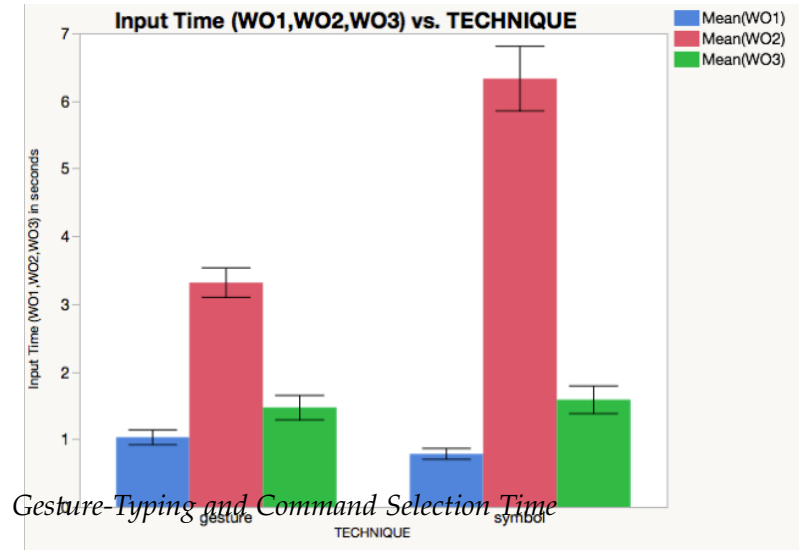


Figure B.7: Average time spent entering each word. WO_2 is the styled word. commands are significantly faster: almost double.

On average, participants spent significantly more time styling words with *markdown symbols* (mean 6.3s) than with *inline gesture shortcuts* (3.3 seconds), with $F_{1,11} = 71.1$, $p < 0.0001$. When we break apart *INPUT TIME* for WO_2 into time to select the command (*COMMAND TIME*) and time to gesture-type it (*TYPING TIME*), we find that participants spend significantly longer writing symbols (mean *COMMAND TIME* =5.8s) than drawing gestures (mean *COMMAND TIME* =1.5s) [$F_{1,11} = 177.6$, $p < 0.0001$] (Figure B.8).

However, they spend more time gesture-typing the styled word when using *inline gesture shortcuts* (mean *TYPING TIME* =1.8s) than *markdown symbols* (mean *TYPING TIME* =0.6s) [$F_{1,11} = 68.3$, $p < 0.0001$]. This may be an artifact of the experimental design, since participants slowed down to check that they had gesture-typed the correct word, before drawing the styling gesture. In the long run, this may actually benefit the *inline gesture shortcuts* technique, because slowing down improves the recognition process with gesture keyboards (Kristensson and Zhai, 2004). Recognized words are less likely to change when users slide into the command gesture input space.

Gap Time

When the participants switch from writing the first word to applying a styling command to the second word, the gap duration (*GAP* (WO_1, WO_2)) is significantly longer for *markdown symbols* (mean=1.9s) than for *inline gesture shortcuts* (mean=1.2s) [$F_{1,11} = 49.7$, $p < 0.0001$]. This suggests that participants needed more time to consider which

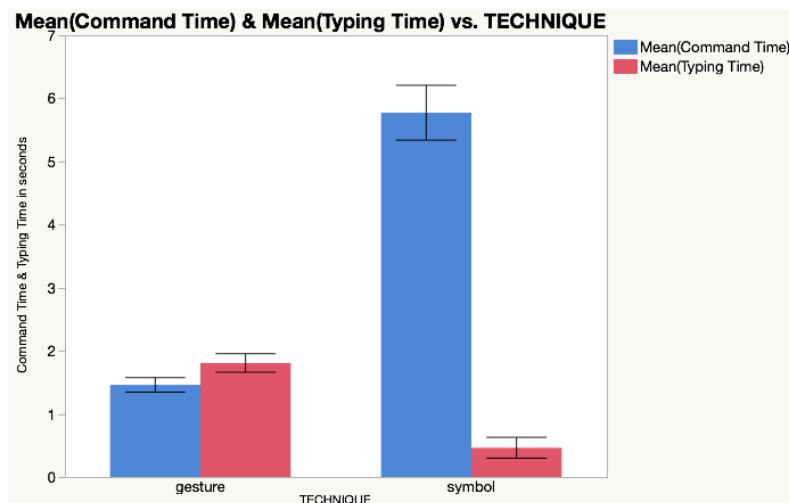


Figure B.8: Average time spent gesture-typing (*TYPING TIME*) and issuing the command (*COMMAND TIME*). Participants drew quickly with *inline gesture shortcuts*, but took significantly longer inserting *markdown symbols*.

key to press when selecting markdown symbols, i.e. searching and pre-planning. However, when participants finish applying the styling command to the middle word, they spend significantly less time writing the third word when using *markdown symbols* (mean *GAP* (WO_2, WO_3) 0.9s) than when using *inline gesture shortcuts* (mean *GAP* (WO_2, WO_3) 1.5s) [$F_{1,11} = 128.4, p < 0.0001$]. In the *markdown symbols* condition, they can already see if they have applied the correct command as they press the SPACE bar, whereas with *inline gesture shortcuts*, they must check again after releasing their finger. This would be improved by displaying a progressive preview at the end of the dynamic guide, but was not made available during the experiment.

Errors

Participants made significantly fewer styling errors with *inline gesture shortcuts* (mean *STYLING ERRORS* = 0.09) than with *markdown symbols* (mean *STYLING ERRORS* = 0.36), [$F_{1,11} = 13.7, p = 0.0035$]. However participants using *inline gesture shortcuts* were somewhat more likely to forget to actually style the word – *inline gesture shortcuts* (mean *MISSING ERRORS* = 0.3) versus *markdown symbols* (mean *MISSING ERRORS* = 0.04), [$F_{1,11} = 26.7, p = 0.0003$]. This is probably an artifact of the experimental setting, since in actual use, users would not ‘forget’ to style a word if they wanted to. We did not find a significant effect of *TECHNIQUE* on accuracy [$F_{1,11} = 49.7, p = 0.47$], which suggests that using gestures to style text does not interfere with typing accuracy.

B.9 Preferences Study

Pre- and Post-test Results

We ran a one-way repeated measures analysis of variance for factor TECHNIQUE to compare *STYLING ERRORS* during the Pre- and Post-test conditions. We found a significant interaction effect [$F_{1,11} = 4.4$, $p = 0.0375$] for *STYLING ERRORS*. In the Pre-test, the average *STYLING ERRORS* for *inline gesture shortcuts* and *markdown symbols* are 0.52 and 0.32, respectively. In the Post-test, the average *STYLING ERRORS* for *inline gesture shortcuts* and *markdown symbols* are 0.35 and 0.38, respectively.

Prior to the pre-test condition, participants had practiced both techniques, but always with a direct indication of how to perform the gesture or what symbols to type. The pre-test was the first time that participants had ever tried executing the commands without help. Participants remembered half the gestures and two thirds of the symbols from the previous practice and experiment condition. The post-test was given after participants had experimented with their choice of technique to recompose their own text, and participants remembered almost two thirds of the gestures. This suggests that we should study longer term use of *CommandBoard's inline gesture shortcuts* technique, to see how well it supports incremental learning over time.

Recomposition Task Results

Although given a choice between using *markdown symbols* or *inline gesture shortcuts*, all participants chose gestures. They ignored the cheat-sheet showing all markdown symbols and their resulting styles. P11 was the exception, but he only looked at the cheat-sheet to get inspiration from the style examples. We observed three strategies when styling words with gestures: thinking of a style first, and then using OctoPocus to follow the corresponding gesture; activating OctoPocus first, and then deciding on a style from the options; and performing a learned gesture to apply a style with no hesitation.

A few participants explained the rationale behind their styling. P2 recomposed a text message to his wife with a shopping list, and he used all available styles to highlight the ingredients they had to buy

for a salad. P8 associated word categories to styles: big meant positive or a lot, small meant negative or uncertain, underline was important or certain, outline and gradient were for special words. P12 also assigned meanings to different styles: gradient for opinions, outline for time-related words, underline for important words, and big for emphasis in general: “*Big is the most useful.*” P11 on the other hand cared less about the different styling options, and mostly focused on emphasizing important words: “*I think I didn’t really want to choose a specific [style], I just wanted to add an effect on it so it looks different from other words.*”

Self-reported Quantitative Measures

Participants were asked to rate six statements on a 5-point Likert scale, from strongly disagree (1) to strongly agree (5). The statements asked whether the current technique helped them to style text: a) accurately, b) quickly, c) easily, d) confidently, e) comfortably, and f) enjoyably. Table B.1 lists the medians of each question for both techniques. An analysis using a Friedman test showed that participants reported significantly stronger agreement for *inline gesture shortcuts* compared to *markdown symbols* on five statements: ACCURATELY ($p = .34, \chi^2(1) = 4.5$), QUICKLY ($p = .007, \chi^2(1) = 7.36$), EASILY ($p = .11, \chi^2(1) = 6.4$), COMFORTABLY ($p = .002, \chi^2(1) = 10$) and ENJOYABLY ($p = .001, \chi^2(1) = 11$).

Statement	Symbols	Gestures
ACCURATELY*	2.5	4.0
QUICKLY*	2.0	4.0
EASILY*	2.0	4.0
CONFIDENTLY	2.5	4.0
COMFORTABLY*	2.0	4.0
ENJOYABLY*	2.0	4.5

Table B.1: Participant ratings of how each technique helped them to style text (median values; * indicates a significant difference). Participants significantly preferred gestures in all categories except ‘confidently’.

User Preferences and Debriefing

The final questionnaire asked participants to rate their preference between the two techniques on a 5-point scale (from strong preference for *markdown symbols* to strong preference for *inline gesture shortcuts*). All participants preferred *gestures*: 10 indicated a strong preference, 2 indicated some preference.

Six participants expressed their preference in terms of *typing flow*, ex-

plaining that *inline gesture shortcuts* best supported styling without interrupting their text composition process. P2 commented “*I didn’t use the symbols at all in the chat. It’s troublesome to have to switch the keyboard, doing it in the beginning and at the end. It really breaks the flow of the writing. While with the gesture, it’s always there, I can pick what I want on the go.*” P9 wrote “*It’s enjoyable to use and in coherent with using gestures to type words.*”

Participants differed with respect to recognition and recall. Four participants found *inline gesture shortcuts* easier to recall than *markdown symbols*: “*I used big, small, underline in the recomposition task, so I remember them*” (P1). However, four participants had difficulty recalling the *inline gesture shortcuts* mappings: P9 said “*the paths of gestures are difficult to link with their meanings*”, and P6 said “*If the gestures are well designed or designed by the user himself, it could be quite natural.*” Two participants felt more comfortable creating mnemonics for *markdown symbols* rather than *inline gesture shortcuts*, despite their overall preference for *inline gesture shortcuts*: “*It’s easier to remember the symbols for each type (+ for big; \$ for the underlined because of the line in the S).*” Three participants also appreciated the convenience of recognizing gestures with OctoPocus rather than always having to recall them: “*this is nice, I don’t have to remember and just follow [the OctoPocus guideline].*”

Finally, we asked participants to suggest other applications for *inline gesture shortcuts*. Four participants suggested using gestures to add emojis: “*I have 5-10 smileys that I always use, so I think it’d be nice if I can use the gesture to get it. Because it’s bothersome having to change to another keyboard view (emojis), so if I can do it with the gesture it’d be cool.*” (P3). Two thought of command shortcuts: “*If you like a webpage, you could do a special gesture to bookmark it. To refresh the page, you could use a circular gesture, etc.*” (P2). Other suggested applications were changing lines, replacing the enter key, taking notes and changing fonts.

B.10 Discussion & Conclusions

CommandBoard repurposes the unused output space above the keyboard to accept gestures that invoke commands; extends gesture keyboards with command gestures, without disrupting existing command invocation techniques; and makes it easy for users to discover gesture-command mappings.

We ran an experiment to compare *CommandBoard*'s command invocation to a conventional markdown language for styling text. We found that participants are not only significantly faster with *CommandBoard* (almost double), but also participants significantly preferred *CommandBoard* (unanimously).

Although we expected that *CommandBoard* would perform better than current markdown commands, we were surprised by the size of the effect (approximately twice as fast) and by how much the participants preferred it over standard markdown commands. We believe this is because users can fluidly style their text without interrupting the flow of their typing. Users not only avoid switching modes, but also avoid selecting text, the most time-consuming aspect of text editing (Fuccella et al., 2013).

The pre- and post-test results from the experiment indicate that users can easily learn gesture commands simply through the process of using them. We expected relatively low post-test scores, since users had only limited experience with the gestures during the practice and experimental conditions. Even so, users clearly made fewer errors in the post-test, which suggests that even limited experience can improve gesture recall.

The experiment restricted *CommandBoard*'s *inline gesture shortcuts* to styling one word at a time. For example, the 'happy'+pigtail gesture generates **happy**. However, sometimes users want to apply a style to multiple words. One option would be to combine *CommandBoard* with other advanced text selection techniques, such as selecting a phrase with a two-finger gesture on top of the keyboard (Fuccella et al., 2013). Gesture grammars can also combine command gestures with selection-scope gestures. For example, in *type-and-execute*, after sliding her finger to the input space above the keyboard, the user could specify the scope of the selection with a marking menu that includes last word, last sentence, last paragraph, and select all.

Future work will focus on enabling the integration of *CommandBoard* with existing communication apps and gesture-typing keyboards, so that users can execute commands from their keyboards across their entire app ecosystem.

References

- (2016). @stickerdownloadbot. <https://storebot.me/bot/stickerdownloadbot>.
- (2016). @storebot. <https://storebot.me/>.
- Altman, I., Vinsel, A., and Brown, B. B. (1981). Dialectic conceptions in social psychology: An application to social penetration and privacy regulation. In *Advances in experimental social psychology*, volume 14, pages 107–160. Elsevier.
- Alvina, J., Griggio, C. F., Bi, X., and Mackay, W. E. (2017). Command-board: Creating a general-purpose command gesture input space for soft keyboard. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, UIST '17*, pages 17–28, New York, NY, USA. ACM.
- Alvina, J., Malloch, J., and Mackay, W. E. (2016). Expressive keyboards: Enriching gesture-typing on mobile devices. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology, UIST '16*, pages 583–593, New York, NY, USA. ACM.
- Andalibi, N., Bentley, F., and Quehl, K. (2017). Multi-channel topic-based mobile messaging in romantic relationships. *Proc. ACM Hum.-Comput. Interact.*, 1(CSCW):20:1–20:18.
- Appert, C. and Zhai, S. (2009). Using strokes as command shortcuts: Cognitive benefits and toolkit support. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09*, pages 2289–2298, New York, NY, USA. ACM.
- Autodesk Inc. (2016). Autodesk Maya 2016 documentation. Available: <http://help.autodesk.com/view/MAYAUL/2016/ENU/?guid=GUID-9C43FD30-D92F-4035-9820-4C3FFDD8E211>, Accessed April 12, 2016.
- Bailly, G., Lecolinet, E., and Nigay, L. (2007). Wave menus: Improving the novice mode of hierarchical marking menus. In Baranauskas, C., Palanque, P., Abascal, J., and Barbosa, S. D. J., editors, *Human-Computer Interaction - INTERACT 2007*, volume 4662 of LNCS,

- pages 475–488, Berlin / Heidelberg. Springer-Verlag.
- Bailly, G., Lecolinet, E., and Nigay, L. (2008). Flower menus: A new type of marking menu with large menu breadth, within groups and efficient expert mode memorization. In *Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '08*, pages 15–22, New York, NY, USA. ACM.
- Bales, E., Li, K. A., and Griwsold, W. (2011). Couplevibe: Mobile implicit communication to improve awareness for (long-distance) couples. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work, CSCW '11*, pages 65–74, New York, NY, USA. ACM.
- Barkhuus, L. (2012). The mismeasurement of privacy: using contextual integrity to reconsider privacy in hci. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 367–376. ACM.
- Bau, O. and Mackay, W. E. (2008). Octopocus: A dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology, UIST '08*, pages 37–46, New York, NY, USA. ACM.
- Baxter, L. A. (1988). A dialectical perspective on communication strategies in relationship development.
- Baxter, L. A. (1990). Dialectical contradictions in relationship development. *Journal of Social and Personal relationships*, 7(1):69–88.
- Bayer, J. B., Ellison, N. B., Schoenebeck, S. Y., and Falk, E. B. (2016). Sharing the small moments: ephemeral social interaction on Snapchat. *Information, Communication & Society*, 19(7):956–977.
- Beaudouin-Lafon, M. (2000). Instrumental interaction: An interaction model for designing post-wimp user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '00*, pages 446–453, New York, NY, USA. ACM.
- Beaudouin-Lafon, M. (2004). Designing interaction, not interfaces. In *Proceedings of the working conference on Advanced visual interfaces*, pages 15–22. ACM.
- Beaudouin-Lafon, M. (2017). Towards Unified Principles of Interaction. In *Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter - CHIItaly '17*, pages 1–2, Cagliari, Italy. ACM Press.
- Beaudouin-Lafon, M. and Mackay, W. E. (2018). Rethinking interaction: From instrumental interaction to human-computer partnerships. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, CHI EA '18*, pages W34:1–W34:5, New York, NY, USA. ACM.
- Becker, J. A. H. and Stamp, G. H. (2005). Impression Management in Chat Rooms: A Grounded Theory Model. *Communication Studies*, 56(3):243–260.

- Bentley, F. and Metcalf, C. J. (2009). The use of mobile social presence. *IEEE Pervasive Computing*, 8(4).
- Bentley, F. R., Chen, Y.-Y., and Holz, C. (2015). Reducing the stress of coordination: Sharing travel time information between contacts on mobile phones. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15*, pages 967–970, New York, NY, USA. ACM.
- Bentley, F. R. and Metcalf, C. J. (2007). Sharing motion information with close family and friends. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '07*, pages 1361–1370, New York, NY, USA. ACM.
- Binder, J., Howes, A., and Sutcliffe, A. (2009). The problem of conflicting social spheres: Effects of network structure on experienced tension in social network sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09*, pages 965–974, New York, NY, USA. ACM.
- Blom, J. O. and Monk, A. F. (2003). Theory of personalization of appearance: Why users personalize their pcs and mobile phones. *Hum.-Comput. Interact.*, 18(3):193–228.
- Bødker, S. and Klokmose, C. N. (2011). The human–artifact model: An activity theoretical approach to artifact ecologies. *Human–Computer Interaction*, 26(4):315–371.
- Branham, S. M., Harrison, S. H., and Hirsch, T. (2012). Expanding the design space for intimacy: Supporting mutual reflection for local partners. In *Proceedings of the Designing Interactive Systems Conference, DIS '12*, pages 220–223, New York, NY, USA. ACM.
- Braun, V. and Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101.
- Brown, B., Taylor, A. S., Izadi, S., Sellen, A., Kaye, J. J., and Eardley, R. (2007). Locating family values: A field trial of the whereabouts clock. In *Proceedings of the 9th International Conference on Ubiquitous Computing, UbiComp '07*, pages 354–371, Berlin, Heidelberg. Springer-Verlag.
- Brown, N. and Stockman, T. (2013). Examining the use of thematic analysis as a tool for informing design of new family communication technologies. In *Proceedings of the 27th International BCS Human Computer Interaction Conference, BCS-HCI '13*, pages 21:1–21:6, Swinton, UK, UK. British Computer Society.
- Bunt, A., Conati, C., and McGrenere, J. (2007). Supporting interface customization using a mixed-initiative approach. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 92–101. ACM.
- Buschek, D., Hassib, M., and Alt, F. (2018). Personal mobile messaging in context: Chat augmentations for expressiveness and aware-

- ness. *ACM Trans. Comput.-Hum. Interact.*, 25(4):23:1–23:33.
- Byron, K. and Baldridge, D. C. (2007). E-mail recipients' impressions of senders' likability: The interactive effect of nonverbal cues and recipients' personality. *The Journal of Business Communication* (1973), 44(2):137–160.
- Capra, R., Khanova, J., and Ramdeen, S. (2013). Work and personal e-mail use by university employees: Pim practices across domain boundaries. *Journal of the American Society for Information Science and Technology*, 64(5):1029–1044.
- Cecchinato, M. E., Sellen, A., Shokouhi, M., and Smyth, G. (2016). Finding email in a multi-account, multi-device world. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 1200–1210, New York, NY, USA. ACM.
- Chalmers, M. and Galani, A. (2004). Seamful interweaving: Heterogeneity in the theory and design of interactive systems. In *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, DIS '04, pages 243–252, New York, NY, USA. ACM.
- Chen, Z., Lu, X., Ai, W., Li, H., Mei, Q., and Liu, X. (2018). Through a gender lens: Learning usage patterns of emojis from large-scale android users. In *WWW*.
- Chung, H., Lee, C.-H. J., and Selker, T. (2006). Lover's cups: Drinking interfaces as new communication channels. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 375–380, New York, NY, USA. ACM.
- Church, K. and de Oliveira, R. (2013). What's up with whatsapp?: Comparing mobile instant messaging behaviors with traditional sms. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '13, pages 352–361, New York, NY, USA. ACM.
- Clarke, V., Hayfield, N., Moller, N., and Tischner, I. (2017). Once upon a time: Story completion methods.
- Consolvo, S., Smith, I. E., Matthews, T., LaMarca, A., Tabert, J., and Powledge, P. (2005). Location disclosure to social relations: Why, when, & what people want to share. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 81–90, New York, NY, USA. ACM.
- Constantin, C. (2002). *To be Or Not to be Emotional: Impression Formation Effects of Emoticons in Moderated Chatrooms*.
- Cramer, H., de Juan, P., and Tetreault, J. (2016). Sender-intended functions of emojis in US messaging. pages 504–509. ACM Press.
- Cramer, H. and Jacobs, M. L. (2015). Couples' communication channels: What, when & why? In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages

- 709–712, New York, NY, USA. ACM.
- Davis, J. L. and Jurgenson, N. (2014). Context collapse: theorizing context collusions and collisions. *Information, Communication & Society*, 17(4):476–485.
- Draxler, S., Stevens, G., Stein, M., Boden, A., and Randall, D. (2012). Supporting the social context of technology appropriation: on a synthesis of sharing tools and tool knowledge. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2835–2844. ACM.
- Dresner, E. and Herring, S. C. (2010). Functions of the nonverbal in cmc: Emoticons and illocutionary force. *Communication theory*, 20(3):249–268.
- Duggan, M. and Ellison, N. B. (2015). Social media update 2014.
- Dyck, J., Pinelle, D., Brown, B. A., and Gutwin, C. (2003). Learning from Games: HCI Design Innovations in Entertainment Software. In *Graphics interface*, volume 2003, pages 237–246. Citeseer.
- Ellison, N. B., Steinfield, C., and Lampe, C. (2007). The benefits of facebook “friends”: social capital and college students’ use of online social network sites. *Journal of Computer-Mediated Communication*, 12(4):1143–1168.
- Ellison, N. B., Steinfield, C., and Lampe, C. (2011). Connection strategies: Social capital implications of facebook-enabled communication practices. *New media & society*, 13(6):873–892.
- Findlater, L. and McGrenere, J. (2004). A comparison of static, adaptive, and adaptable menus. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 89–96. ACM.
- Findlater, L., McGrenere, J., and Modjeska, D. (2008). Evaluation of a role-based approach for customizing a complex development environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1267–1270. ACM.
- Flanagan, J. C. (1954). The critical incident technique. *Psychological bulletin*, 51(4):327.
- Freed, D., Palmer, J., Minchala, D., Levy, K., Ristenpart, T., and Dell, N. (2018). “A stalker’s paradise”: How intimate partner abusers exploit technology. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 667. ACM.
- Freed, D., Palmer, J., Minchala, D. E., Levy, K., Ristenpart, T., and Dell, N. (2017). Digital technologies and intimate partner violence: A qualitative analysis with multiple stakeholders. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW):46.
- Fucella, V., Isokoski, P., and Martin, B. (2013). Gestures and widgets: Performance in text editing on multi-touch capable mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’13*, pages 2785–2794, New York, NY,

USA. ACM.

- Garcia, J., Tsandilas, T., Agon, C., and Mackay, W. (2012). Interactive paper substrates to support musical creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1825–1828. ACM.
- Gaver, W. W., Beaver, J., and Benford, S. (2003). Ambiguity as a resource for design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '03*, pages 233–240, New York, NY, USA. ACM.
- Gjerlufsen, T., Klokmoose, C. N., Eagan, J., Pillias, C., and Beaudouin-Lafon, M. (2011). Shared substance: developing flexible multi-surface applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3383–3392. ACM.
- Glikson, E., Cheshin, A., and van Kleef, G. A. (2017). The Dark Side of a Smiley: Effects of Smiling Emoticons on Virtual First Impressions. *Social Psychological and Personality Science*, page 194855061772026.
- Goffman, E. (1959). The presentation of self in. *Butler, Bodies that Matter*.
- Grinter, R. E. and Eldridge, M. A. (2001). Y do tngrs luv 2 txt msg? In *Proceedings of the Seventh Conference on European Conference on Computer Supported Cooperative Work, ECSCW'01*, pages 219–238, Norwell, MA, USA. Kluwer Academic Publishers.
- Gross, B. M. and Churchill, E. F. (2007). Addressing constraints: Multiple usernames task spillage and notions of identity. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems, CHI EA '07*, pages 2393–2398, New York, NY, USA. ACM.
- Guay, M. (2010). Superpower your touchpad computer with Scrybe. *How-To Geek*. Available: <http://www.howtogeek.com/howto/16587/superpower-your-touchpad-computer-with-scrybe/>, Accessed April 12, 2016.
- Hancock, J. T. and Dunham, P. J. (2001). Impression formation in computer-mediated communication revisited: An analysis of the breadth and intensity of impressions. *Communication research*, 28(3):325–347.
- Haraty, M., McGrenere, J., and Bunt, A. (2017). Online Customization Sharing Ecosystems: Components, Roles, and Motivations. pages 2359–2371. ACM Press.
- Harrison, S. and Dourish, P. (1996). Re-place-ing space: The roles of place and space in collaborative systems. In *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work, CSCW '96*, pages 67–76, New York, NY, USA. ACM.
- Hassenzahl, M., Heidecker, S., Eckoldt, K., Diefenbach, S., and Hillmann, U. (2012). All you need is love: Current strategies of me-

- diating intimate relationships through technology. *ACM Trans. Comput.-Hum. Interact.*, 19(4):30:1–30:19.
- Hassib, M., Buschek, D., Wozniak, P. W., and Alt, F. (2017). Heartchat: Heart rate augmented mobile chat to support empathy and awareness. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 2239–2251, New York, NY, USA. ACM.
- Hogan, B. (2010). The presentation of self in the age of social media: Distinguishing performances and exhibitions online. *Bulletin of Science, Technology & Society*, 30(6):377–386.
- Hollan, J. and Stornetta, S. (1992). Beyond being there. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 119–125. ACM.
- Horvitz, E. (1999). Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, pages 159–166, New York, NY, USA. ACM.
- Houser, M. L., Fleuriot, C., and Estrada, D. (2012). The Cyber Factor: An Analysis of Relational Maintenance Through the Use of Computer-Mediated Communication. *Communication Research Reports*, 29(1):34–43.
- Hsieh, S. H. and Tseng, T. H. (2017). Playfulness in mobile instant messaging: Examining the influence of emoticons and text messaging on social interaction. *Computers in Human Behavior*, 69:405–414.
- Hutchinson, H., Mackay, W., Westerlund, B., Bederson, B. B., Druin, A., Plaisant, C., Beaudouin-Lafon, M., Conversy, S., Evans, H., Hansen, H., Roussel, N., and Eiderbäck, B. (2003). Technology probes: Inspiring design for and with families. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 17–24, New York, NY, USA. ACM.
- Jacobs, M., Cramer, H., and Barkhuus, L. (2016). Caring about sharing: Couples' practices in single user device access. In *Proceedings of the 19th International Conference on Supporting Group Work*, GROUP '16, pages 235–243, New York, NY, USA. ACM.
- Jalal, G. (2016). *Réification des propriétés visuelles pour les tâches de composition*. PhD thesis, Paris Saclay.
- Jasso, G. and Opp, K.-D. (1997). Probing the character of norms: A factorial survey analysis of the norms of political action. *American Sociological Review*, pages 947–964.
- Jung, Y. and Lyytinen, K. (2014). Towards an ecological account of media choice: A case study on pluralistic reasoning while choosing email. *Info. Sys. Jour.*, 24(3):271–293.
- Kahler, H. (2001). More than words - collaborative tailoring of a word processor. *Journal of Universal Computer Science*, 7(8):826–847.

- Kaye, J. J. (2006). I just clicked to say i love you: Rich evaluations of minimal communication. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems, CHI EA '06*, pages 363–368, New York, NY, USA. ACM.
- Kelly, R., Gooch, D., Patil, B., and Watts, L. (2017). Demanding by design: Supporting effortful communication practices in close personal relationships. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 70–83. ACM.
- Kelly, R. and Watts, L. (2015). Characterising the inventive appropriation of emoji as relationally meaningful in mediated close personal relationships. *Experiences of Technology Appropriation: Unanticipated Users, Usage, Circumstances, and Design*.
- Kim, H., Kim, G. J., Park, H. W., and Rice, R. E. (2007). Configurations of Relationships in Different Media: FtF, Email, Instant Messenger, Mobile Phone, and SMS. *Journal of Computer-Mediated Communication*, 12(4):1183–1207.
- Klokmoose, C. N. and Beaudouin-Lafon, M. (2009). Vigo: Instrumental interaction in multi-surface environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09*, pages 869–878, New York, NY, USA. ACM.
- Klokmoose, C. N., Eagan, J. R., Baader, S., Mackay, W., and Beaudouin-Lafon, M. (2015). Webstrates: Shareable dynamic media. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, UIST '15*, pages 280–290, New York, NY, USA. ACM.
- Kristensson, P.-O. and Zhai, S. (2004). Shark2: A large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology, UIST '04*, pages 43–52, New York, NY, USA. ACM.
- Kumaraguru, P. and Cranor, L. F. (2005). *Privacy indexes: a survey of Westin's studies*. Carnegie Mellon University, School of Computer Science, Institute for
- Kurtenbach, G. and Buxton, W. (1991). Issues in combining marking and direct manipulation techniques. In *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology, UIST '91*, pages 137–144, New York, NY, USA. ACM.
- Kurtenbach, G. and Buxton, W. (1993). The limits of expert performance using hierarchic marking menus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems, CHI '93*, pages 482–487, New York, NY, USA. ACM.
- Lederer, S., Hong, J. I., Dey, A. K., and Landay, J. A. (2004). Personal privacy through understanding and action: Five pitfalls for de-

- signers. *Personal Ubiquitous Comput.*, 8(6):440–454.
- Lee, J. Y., Hong, N., Kim, S., Oh, J., and Lee, J. (2016). Smiley face: why we use emoticon stickers in mobile messaging. pages 760–766. ACM Press.
- Licoppe, C. (2004). ‘connected’ presence: The emergence of a new repertoire for managing social relationships in a changing communication technoscape. *Environment and planning D: Society and space*, 22(1):135–156.
- Ling, R. and Yttri, B. (2002). 10 hyper-coordination via mobile phones in norway. *Perpetual contact: Mobile communication, private talk, public performance*, 139.
- Long, Jr., A. C., Landay, J. A., and Rowe, L. A. (1999). Implications for a gesture design tool. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '99*, pages 40–47, New York, NY, USA. ACM.
- Lottridge, D., Masson, N., and Mackay, W. (2009). Sharing empty moments: Design for remote couples. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09*, pages 2329–2338, New York, NY, USA. ACM.
- Lu, X., Ai, W., Liu, X., Li, Q., Wang, N., Huang, G., and Mei, Q. (2016). Learning from the ubiquitous language: an empirical analysis of emoji usage of smartphone users. pages 770–780. ACM Press.
- Mackay, W. E. (1990a). Patterns of sharing customizable software. In *Proceedings of the 1990 ACM Conference on Computer-supported Cooperative Work, CSCW '90*, pages 209–221, New York, NY, USA. ACM.
- Mackay, W. E. (1990b). *Users and customizable software: A co-adaptive phenomenon*. PhD thesis, Citeseer.
- Mackay, W. E. (1991). Triggers and barriers to customizing software. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 153–160. ACM.
- Mackay, W. E. (2000). Responding to cognitive overload: Co-adaptation between users and technology. *Intellectica*, 30(1):177–193.
- Mackay, W. E. (2002). Using video to support interaction design. *DVD Tutorial, CHI'02*.
- Mackay, W. E. and Fayard, A.-L. (1997). Hci, natural science and design: a framework for triangulation across disciplines. In *Proceedings of the 2nd conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 223–234. ACM.
- MacLean, A., Carter, K., Lövstrand, L., and Moran, T. (1990). User-tailorable systems: pressing the issues with buttons. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 175–182. ACM.
- Mancini, C., Rogers, Y., Thomas, K., Joinson, A. N., Price, B. A., Ban-

- dara, A. K., Jedrzejczyk, L., and Nuseibeh, B. (2011). In the best families: Tracking and relationships. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11*, pages 2419–2428, New York, NY, USA. ACM.
- Marshall, T. C., Bejanyan, K., Di Castro, G., and Lee, R. A. (2013). Attachment styles as predictors of facebook-related jealousy and surveillance in romantic relationships. *Personal Relationships*, 20(1):1–22.
- 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.
- Matthews, T., O'Leary, K., Turner, A., Sleeper, M., Woelfer, J. P., Shelton, M., Manthorne, C., Churchill, E. F., and Consolvo, S. (2017). Stories from survivors: Privacy & security practices when coping with intimate partner abuse. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2189–2201. ACM.
- Maudet, N., Jalal, G., Tchernavskij, P., Beaudouin-Lafon, M., and Mackay, W. E. (2017). Beyond Grids: Interactive Graphical Substrates to Structure Digital Layout. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*, pages 5053–5064, Denver, Colorado, USA. ACM Press.
- Mauney, D., Howarth, J., Wirtanen, A., and Capra, M. (2010). Cultural similarities and differences in user-defined gestures for touch-screen user interfaces. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems, CHI EA '10*, pages 4015–4020, New York, NY, USA. ACM.
- McGrenere, J., Baecker, R. M., and Booth, K. S. (2007). A field evaluation of an adaptable two-interface design for feature-rich software. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 14(1):3.
- McRoberts, S., Ma, H., Hall, A., and Yarosh, S. (2017). Share First, Save Later: Performance of Self through Snapchat Stories. pages 6902–6911. ACM Press.
- Miller, H., Thebault-Spieker, J., Chang, S., Johnson, I., Terveen, L., and Hecht, B. (2016). Blissfully happy or ready to fight: Varying Interpretations of Emoji. *Proceedings of ICWSM*, 2016.
- Miltner, K. M. and Highfield, T. (2017). Never gonna GIF you up: Analyzing the cultural significance of the animated GIF. *Social Media+ Society*, 3(3):2056305117725223.
- Morris, M. R., Wobbrock, J. O., and Wilson, A. D. (2010). Understanding users' preferences for surface gestures. In *Proceedings*

- of *Graphics Interface 2010*, GI '10, pages 261–268, Toronto, Ont., Canada, Canada. Canadian Information Processing Society.
- Nacenta, M. A., Kamber, Y., Qiang, Y., and Kristensson, P. O. (2013). Memorability of pre-designed and user-defined gesture sets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1099–1108, New York, NY, USA. ACM.
- Nardi, B. A. and Miller, J. R. (1991). Twinkling lights and nested loops: distributed problem solving and spreadsheet development. *International Journal of Man-Machine Studies*, 34(2):161–184.
- Nardi, B. A., Whittaker, S., and Bradner, E. (2000). Interaction and outeraction: Instant messaging in action. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, CSCW '00*, pages 79–88, New York, NY, USA. ACM.
- Neustaedter, C., Brush, A. J. B., and Greenberg, S. (2009). The calendar is crucial: Coordination and awareness through the family calendar. *ACM Trans. Comput.-Hum. Interact.*, 16(1):6:1–6:48.
- Neustaedter, C. and Greenberg, S. (2012). Intimacy in long-distance relationships over video chat. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 753–762. ACM.
- Neustaedter, C., Pang, C., Forghani, A., Oduor, E., Hillman, S., Judge, T. K., Massimi, M., and Greenberg, S. (2015). Sharing domestic life through long-term video connections. *ACM Trans. Comput.-Hum. Interact.*, 22(1):3:1–3:29.
- Nouwens, M., Griggio, C. F., and Mackay, W. E. (2017). Whatsapp is for family; messenger is for friends: Communication places in app ecosystems. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 727–735. ACM.
- Oh, U. and Findlater, L. (2013). The challenges and potential of end-user gesture customization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1129–1138, New York, NY, USA. ACM.
- O'Hara, K. P., Massimi, M., Harper, R., Rubens, S., and Morris, J. (2014). Everyday dwelling with whatsapp. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing, CSCW '14*, pages 1131–1143, New York, NY, USA. ACM.
- Omanson, R. C., Miller, C. S., Young, E., and Schwantes, D. (2010). Comparison of mouse and keyboard efficiency. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 54, pages 600–604. Sage Publications Sage CA: Los Angeles, CA.
- O'Sullivan, P. B. (2000). What you don't know won't hurt me: Im-

- pression management functions of communication channels in relationships. *Human Communication Research*, 26(3):403–431.
- Oulasvirta, A. and Blom, J. (2008). Motivations in personalisation behaviour. *Interact. Comput.*, 20(1):1–16.
- Page, S. R., Johnsgard, T. J., Albert, U., and Allen, C. D. (1996). User customization of a word processor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 340–346. ACM.
- Park, C. Y., Faklaris, C., Zhao, S., Sciuto, A., Dabbish, L., and Hong, J. (2018). Share and share alike? an exploration of secure behaviors in romantic relationships. In *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*, pages 83–102, Baltimore, MD. USENIX Association.
- Park, J., Barash, V., Fink, C., and Cha, M. (2013). Emoticon Style: Interpreting Differences in Emoticons Across Cultures. In *Icwsn*.
- Pastel, R. (2006). Measuring the difficulty of steering through corners. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '06, pages 1087–1096, New York, NY, USA. ACM.
- Peissner, M. and Edlin-White, R. (2013). User control in adaptive user interfaces for accessibility. In *IFIP Conference on Human-Computer Interaction*, pages 623–640. Springer.
- Perkel, J. M. (2017). How scientists use slack. *Nature News*, 541(7635):123.
- Pohl, H., Domin, C., and Rohs, M. (2017). Beyond Just Text: Semantic Emoji Similarity Modeling to Support Expressive Communication. *ACM Transactions on Computer-Human Interaction*, 24(1):1–42.
- Postmes, T., Spears, R., and Lea, M. (1998). Breaching or building social boundaries? side-effects of computer-mediated communication. *Communication research*, 25(6):689–715.
- Riche, Y. and Mackay, W. (2007). markerclock : A communicating augmented clock for elderly. In *Lecture Notes in Computer Science - INTERACT 2007 Conference*, pages 408–411. Springer Verlag.
- Riche, Y. and Mackay, W. (2010). Peercare: supporting awareness of rhythms and routines for better aging in place. *Computer Supported Cooperative Work (CSCW)*, 19(1):73–104.
- Riordan, M. A. (2017). Emojis as tools for emotion work: Communicating affect in text messages. *Journal of Language and Social Psychology*, 36(5):549–567.
- Rossi, P. H. and Anderson, A. B. (1982). The factorial survey approach: An introduction. *Measuring social judgments: The factorial survey approach*, pages 15–67.
- Rost, M., Kitsos, C., Morgan, A., Podlubny, M., Romeo, P., Russo, E.,

- and Chalmers, M. (2016). Forget-me-not: History-less Mobile Messaging. pages 1904–1908. ACM Press.
- Ruiz, J., Li, Y., and Lank, E. (2011). User-defined motion gestures for mobile interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11*, pages 197–206, New York, NY, USA. ACM.
- Schildt, E., Leinfors, M., and Barkhuus, L. (2016). Communication, coordination and awareness around continuous location sharing. In *Proceedings of the 19th International Conference on Supporting Group Work, GROUP '16*, pages 257–265, New York, NY, USA. ACM.
- Scissors, L. E. and Gergle, D. (2013). "back and forth, back and forth": Channel switching in romantic couple conflict. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work, CSCW '13*, pages 237–248, New York, NY, USA. ACM.
- Shklovski, I., Barkhuus, L., Bornoe, N., and Kaye, J. J. (2015). Friendship maintenance in the digital age: Applying a relational lens to online social interaction. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, CSCW '15*, pages 1477–1487, New York, NY, USA. ACM.
- Singh, S., Cabraal, A., Demosthenous, C., Astbrink, G., and Furlong, M. (2007). Password sharing: Implications for security design based on social practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '07*, pages 895–904, New York, NY, USA. ACM.
- Sleeper, M., Melicher, W., Habib, H., Bauer, L., Cranor, L. F., and Mazurek, M. L. (2016). Sharing personal content online: Exploring channel choice and multi-channel behaviors. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 101–112. ACM.
- Smith, A. and Anderson, M. (2018). Social media use in 2018. *Pew research center*.
- Sotamaa, O. (2010). When the game is not enough: Motivations and practices among computer game modding culture. *Games and Culture*, 5(3):239–255.
- Stutzman, F., Vitak, J., Ellison, N. B., Gray, R., and Lampe, C. (2012). Privacy in interaction: Exploring disclosure and social capital in facebook. In *ICWSM*.
- Su, N. M. (2009). Temporal patterns of communication: Media combos. In *Proceedings of the ACM 2009 International Conference on Supporting Group Work, GROUP '09*, pages 387–388, New York, NY, USA. ACM.
- Sugiyama, S. (2015). Kawaii meiru and maroyaka neko : Mobile emoji for relationship maintenance and aesthetic expressions among japanese teens. *First Monday*, 20(10).

- Tabard, A., Mackay, W., Roussel, N., and Letondal, C. (2007). Pagelinker: integrating contextual bookmarks within a browser. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 337–346. ACM.
- Taber, L. and Whittaker, S. (2018). Personality Depends on The Medium: Differences in Self-Perception on Snapchat, Facebook and Offline. pages 1–13. ACM Press.
- Tang, J. C., Yankelovich, N., Begole, J., Van Kleek, M., Li, F., and Bhaldia, J. (2001). Connexus to awarenex: Extending awareness to mobile users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '01*, pages 221–228, New York, NY, USA. ACM.
- Tchernavskij, P. (2017). Decomposing Interactive Systems. In *CHI 2017-Tools workshop at CHI'2017*, page 4.
- Thayer, A., Bietz, M. J., Derthick, K., and Lee, C. P. (2012). I love you, let's share calendars: Calendar sharing as relationship work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work, CSCW '12*, pages 749–758, New York, NY, USA. ACM.
- Thayer, A., Sirjani, B., and Lee, C. P. (2013). Recalibrating the ratio: Enacting accountability in intimate relationships using shared calendars. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work, CSCW '13*, pages 203–214, New York, NY, USA. ACM.
- Tolins, J. and Samermit, P. (2016). GIFs as Embodied Enactments in Text-Mediated Conversation. *Research on Language and Social Interaction*, 49(2):75–91.
- Tossell, C. C., Kortum, P., Shepard, C., Rahmati, A., and Zhong, L. (2012). An empirical analysis of smartphone personalisation: measurement and user variability. *Behaviour & Information Technology*, 31(10):995–1010.
- Vinkers, C. D., Finkenauer, C., and Hawk, S. T. (2011). Why do close partners snoop? predictors of intrusive behavior in newlywed couples. *Personal Relationships*, 18(1):110–124.
- Vitak, J. and Kim, J. (2014). "You can't block people offline": examining how facebook's affordances shape the disclosure process. pages 461–474. ACM Press.
- Vitale, F., Janzen, I., and McGrenere, J. (2018). Hoarding and minimalism: Tendencies in digital data preservation. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI '18*, pages 587:1–587:12, New York, NY, USA. ACM.
- Walther, J. B. (1993). Impression development in computer-mediated interaction. *Western Journal of Communication*, 57(4):381–398.
- Walther, J. B. (1996). Computer-mediated communication: Impersonal,

- interpersonal, and hyperpersonal interaction. *Communication research*, 23(1):3–43.
- Walther, J. B. (2007). Selective self-presentation in computer-mediated communication: Hyperpersonal dimensions of technology, language, and cognition. *Computers in Human Behavior*, 23(5):2538–2557.
- Walther, J. B. and Burgoon, J. K. (1992). Relational communication in computer-mediated interaction. *Human communication research*, 19(1):50–88.
- Walther, J. B. and D’Addario, K. P. (2001). The impacts of emoticons on message interpretation in computer-mediated communication. *Social science computer review*, 19(3):324–347.
- Wilkinson, D., Namara, M., Badillo-Urquiola, K., Wisniewski, P. J., Knijnenburg, B. P., Page, X., Toch, E., and Romano-Bergstrom, J. (2018). Moving beyond a "one-size fits all": Exploring individual differences in privacy. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, CHI EA '18*, pages W16:1–W16:8, New York, NY, USA. ACM.
- Winograd, T., Flores, F., and Flores, F. F. (1986). *Understanding computers and cognition: A new foundation for design*. Intellect Books.
- Wiseman, S. and Gould, S. J. J. (2018). Repurposing emoji for personalised communication: Why pizza means “i love you”. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI '18*, pages 152:1–152:10, New York, NY, USA. ACM.
- Wobbrock, J. O., Morris, M. R., and Wilson, A. D. (2009). User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '09*, pages 1083–1092, New York, NY, USA. ACM.
- Wulf, V., Pipek, V., and Won, M. (2008). Component-based tailorability: Enabling highly flexible software applications. *International Journal of Human-Computer Studies*, 66(1):1–22.
- Xu, L., Yi, C., and Xu, Y. (2007). Emotional expression online: The impact of task, relationship and personality perception on emoticon usage in instant messenger. *PACIS 2007 Proceedings*, page 79.
- Yang, C.-c., Brown, B. B., and Braun, M. T. (2014). From Facebook to cell calls: Layers of electronic intimacy in college students’s interpersonal relationships. *New Media & Society*, 16(1):5–23.
- Yee, N., Ducheneaut, N., Nelson, L., and Likarish, P. (2011). Inverted elves & conscientious gnomes: the expression of personality in world of warcraft. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 753–762. ACM.
- Yoo, J. (2007). To smile or not to smile:): Defining the effects of emoticons on relational outcomes. In *annual meeting of International*

Communication Association Conference. Chicago, IL.

- Zhao, X., Lampe, C., and Ellison, N. B. (2016). The Social Media Ecology: User Perceptions, Strategies and Challenges. pages 89–100. ACM Press.
- Zhao, X., Schwanda Sosik, V., and Cosley, D. (2012). It's complicated: How romantic partners use facebook. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, pages 771–780, New York, NY, USA. ACM.
- Zhou, R., Hentschel, J., and Kumar, N. (2017). Goodbye Text, Hello Emoji: Mobile Communication on WeChat in China. pages 748–759. ACM Press.

Titre : Conception dédiée aux écosystèmes d'applications de communication

Mots clés : Communication virtuelle, écosystèmes d'applications de communication, co-adaptation, interaction humain-machine

Résumé :

L'utilisation de plusieurs applications de communication au lieu d'une est de plus en plus commune. En particulier, ces applications permettent à l'utilisateur de communiquer de diverses façons avec son partenaire, les membres de sa famille ou ses amis proches. Je soutiens qu'il est nécessaire de comprendre comment les gens communiquent au travers d'un écosystème d'applications.

Dans ce thèse, je décris comment les pratiques de communication d'un utilisateur via une application sont influencées par les contacts et fonctionnalités d'autres applications de l'écosystème. En plus, je montre comment les fonctionnalités exclusives à une application empêchent les utilisateurs de s'exprimer de manière cohérente dans leur écosystème d'ap-

plications. Ensuite, j'explore comment améliorer la communication via un écosystème d'applications en réutilisant des mécanismes déjà disponibles dans les systèmes d'exploitation des téléphones mobiles : des notifications, des commandes gestuelles, et des claviers virtuels.

En conclusion, je soutiens que lorsque les utilisateurs communiquent via de nombreuses applications, chaque application affecte la manière dont l'utilisateur communique dans les autres applications de l'écosystème. Nous devrions cesser de concevoir uniquement des applications isolées mais concevoir des mécanismes qui aident les utilisateurs à préserver leurs lieux de communication et à exprimer leur identité et leur intimité avec leurs proches de manière cohérente dans leur écosystème d'applications.

Title : Designing for ecosystems of communication apps

Keywords : computer-mediated communication, ecosystems of communication apps, co-adaptation, human-computer interaction

Abstract : More and more, people communicate via not one, but many messaging apps and social media platforms. In particular, couples, close friends and families use multiple apps to express caring in diverse ways throughout the day. I argue that we need a deeper understanding of how people communicate via ecosystems of communication apps.

In this thesis, I show that the way people communicate in an app is not only influenced by its contacts and functionality but also by the contacts and functionality in their other apps. I also show how app-exclusive features prevent people from expressing themselves consistently across their apps. Moreover, I present novel designs, technologies and studies that explore how to better support ecosystems of commu-

nication apps by repurposing three mechanisms currently available in mobile operating systems : notifications, which users can overlay on top of any open app ; gesture commands, which users could perform on any app that recognizes gestures ; and soft keyboards, which appear in any app where users can type text.

In conclusion, I argue that when people communicate via multiple apps, each app shapes how communication happens in others. We should shift from building isolated apps to designing mechanisms that help users preserve their communication places and express their identities and intimate bonds with others consistently across their apps.

