Interacting with signifier-less designs - the case of swhidgets

Nicole Ke Cheng Pong

Université de Lille
École Doctorale Sciences Pour l'Ingénieur
Laboratoire CRIStAL UMR 9189

INTERACTING WITH SIGNIFIER-LESS DESIGNS

– the case of swhidgets –

Thèse présentée par

NICOLE KE CHEN PONG

en vue de l'obtention du grade de docteur en informatique de l'Université de Lille

Sous la direction de :	Stéphane Huot	directeur
	Nicolas Roussel	co-directeur
	Sylvain Malacria	encadrant

Thèse soutenue le 21 octobre 2020
devant un jury composé de :

Rapporteurs	Nadine Couture	Professeure des universités, École Supérieure des Technologies Industrielles Avancées (ESTIA)
	Kris Luyten	Full Professor, Hasselt University
Examinateurs	Caroline Appert	Directrice de recherche, CNRS
	Laurent Grisoni	Professeur des universités, Université de Lille, président du jury
	Sylvain Malacria	Chargé de recherche, Inria, encadrant de thèse
Directeur de thèse	Stéphane Huot	Directeur de recherche, Inria
Invité	Nicolas Roussel	Directeur de recherche, Inria

RÉSUMÉ

Le cas d'un utilisateur confronté à une interface qui ne l'informe pas d'une possibilité d'interaction au moment où il en a besoin est un problème fondamental d'IHM. La présence de telles possibilités d'interaction *non-signalées* est fréquente dans les interfaces homme-machine modernes et potentiellement problématique, rendant nécessaire l'étude des interfaces et techniques d'interaction dites *sans signifiants*.

Un exemple de conception "sans signifiants" moderne est ce que j'appelle un *swhidget* pour "SWIpe-revealed HIDden WIDGET" : un composant d'interface normalement caché sous les bords de l'écran ou sous un autre objet, pouvant être révélé en le tirant à l'aide d'un geste de balayage selon une métaphore de manipulation physique. Les *swhidgets* sont des composants importants des interfaces de téléphones et tablettes à écran tactile, et sont la principale conception sans signifiant étudiée dans cette thèse.

En présence d'une conception sans signifiant, les utilisateurs peuvent être confus quant à ce qu'ils doivent faire pour atteindre leur but, ou être réduits à utiliser des méthodes sous-optimales parce qu'ils ne sont pas conscients de l'existence de meilleures alternatives. Il est donc généralement recommandé d'éviter de concevoir des interfaces sans signifiants. De telles interfaces sont pourtant courantes bien que les concepteurs soient conscient des problèmes qu'elles causent. Elles méritent donc une analyse plus approfondie, au delà du simple conseil de les éviter. En effet, il pourrait y avoir de bonnes raisons de concevoir des interfaces sans signifiants, qu'elles aient des qualités difficiles à mettre en évidence en l'état actuel de notre compréhension ou qu'il soit simplement impossible de les éviter.

Dans cette thèse, j'analyse les raisons pouvant inciter à la conception d'interfaces qui n'exposent pas clairement les possibilités d'interaction qu'elles offrent, en prenant les *swhidgets* comme objet d'étude principal. Pour cette étude initiale des *swhidgets*, je me concentre sur les points suivants : que sont les conceptions sans signifiants et quels aspects des *swhidgets* leur sont propres ? Les utilisateurs connaissent-ils les *swhidgets* de leurs systèmes ? Comment les ont-ils connus malgré l'absence de signifiants ? Quels avantages y a-t-il à ne pas avoir de signifiants ?

Les contributions de cette thèse sur ces points sont :

- Une definition des conceptions sans signifiants basée sur des observations de telles conceptions dans des interfaces.

- Une analyse des notions fondamentales requises pour la définition des conceptions sans signifiants : affordances, signifiants et sémiotique.
- Un modèle de la découverte et adoption par les utilisateurs de techniques d'interaction en général, reposant sur trois dimensions et leurs relations : les compétences et connaissances actuelles des utilisateurs, leur motivations, et les moyens par lesquels une interface peut informer ses utilisateurs.
- Les notions de degré de connaissance et de source de connaissance, dérivées de ce modèle, qui permettent d'évaluer expérimentalement à quel point les utilisateurs connaissent une technique d'interaction et comment ils l'ont découverte.
- La conception et les résultats de deux expériences sur les *swhidgets* d'iOS pour évaluer la connaissance qu'en ont les utilisateurs, comment ils les ont découvert, leurs éventuelles raisons de ne pas les utiliser, comment ils les perçoivent globalement et les intègrent dans leur façon de penser l'interaction. Ces études montrent que les *swhidgets* sont globalement appréciés et relativement bien connus, tout en laissant de la place pour des améliorations, surtout pour certains *swhidgets*.

Cette thèse ouvre des perspectives concernant le transfert de connaissances entre applications, la pertinence du concept d'expérience utilisateur pour la compréhension des *swhidgets*, et la possibilité de favoriser leur découverte lors de transitions animées entre vues.

ABSTRACT

At the heart of this thesis is a common but problematic situation that users of digital systems often face in their daily interactions: to interact with the system, they need some knowledge of an interaction possibility, some piece of information about the interface, but this information is not provided in the context in which they need it. I call such interaction possibilities *non-signified*, and *signifier-less designs* the interfaces and interaction techniques that rely on non-signified interaction possibilities.

An example of modern signifier-less design is what I call "*swhidgets*" for "SWIpe-revealed HIDden WIDGETS": widgets that are hidden under the screen bezels or other interface elements, out of view and not advertised by any graphical mark, but that can be revealed by dragging them into view with a swipe gesture relying on a physical manipulation metaphor. *Swhidgets* are an important component of touch-based smartphone and tablets interfaces, and will be the principal signifier-less design studied in this thesis.

When facing a signifier-less design, users may be confused about what they should do and how to achieve their goals; or they might have to use suboptimal ways of achieving their goals because they are unaware of the existence of more efficient options. It is thus usually advised to avoid signifier-less designs. Yet, despite designers' awareness of the problems they may cause, signifier-less designs are common in user interfaces. They thus deserve a deeper analysis than simply advising to avoid them in interface design. Indeed, there might be good reasons to apply this design: maybe they provide some benefits that are hard to see with our current understanding of these designs, or maybe there is no way to avoid them.

In this thesis, I study the question of why designers would create interfaces that do not clearly expose some of their interaction possibilities, taking the case of *swhidgets* as an example and focus of inquiry. As a preliminary work on *swhidgets*, I focus on the following questions: What are signifier-less designs and what aspects of *swhidgets* design make them unique? Do users know the *swhidgets* provided by their system? How did they get to know them despite their lack of signifiers? What are the benefits of not having signifiers in the design of *swhidgets*?

My contributions to these questions are:

- I define signifier-less designs and provide observations of this type of design in user interfaces.
- I provide an analysis of the fundamental notions required to define signifier-less designs: affordances, signifiers and semiotics.

- I propose a model of user discovery and adoption of interaction techniques in general, relying on three dimensions and their relationships: users' current knowledge and skills, users' motivations, and the design means of informing users provided by the interfaces.
- I propose the notions of Degree of Knowledge and Source of Knowledge derived from this model, that can be used in experiments to evaluate how well the participants know an interaction technique and how they discovered it.
- I present the design and results of two studies on iOS *swhidgets* that investigate how well users known them, how they discover them, their reasons for not using them, how they generally feel about them, and how they integrate them in the way they think about their interactions with the system. These studies revealed that *swhidgets* were globally appreciated and relatively well known by users, although there is still room for improvement, notably for some specific *swhidgets*.

I conclude with perspective for future works regarding the transfer of knowledge about *swhidgets* from one application to another, the pertinence of considering all aspects of user experience to understand the design of *swhidgets*, and the possibility to increase the discoverability of *swhidgets* by using animated transitions between interface views.

PUBLICATIONS

Pong, Nicole Ke Chen (Aug. 2017). "Understanding and Increasing Users' Interaction Vocabulary." In: *29ème conférence francophone sur l'Interaction Homme-Machine*. Ed. by AFIHM. Rencontres doctorales. AFIHM. Poitiers, France, 4 p. URL: https://hal.archives-ouvertes.fr/hal-01577901.

Pong, Nicole Ke Chen and Sylvain Malacria (2019). "Awareness, Usage and Discovery of Swipe-Revealed Hidden Widgets in IOS." In: *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*. ISS '19. Daejeon, Republic of Korea: Association for Computing Machinery, pp. 193–204. ISBN: 9781450368919. DOI: 10.1145/3343055.3359713.

ACKNOWLEDGMENTS

This dissertation would not have been possible without the support I received from an extended community of colleagues, friends, families, and people working in various places. I express here my sincere appreciation to these people and organizations.

First of all, I would like to thank my advisors and jury members for their sincere care, interest, and thoughtful feedback to the manuscript. To my director Professor Nicolas Roussel - thank you for giving me the opportunity to work with you as a student who came from more of a design domain, I appreciate your guidance and your always-heartwarming introduction of me in public events including my final defense, I am very grateful of your thoughtful attention all along my thesis. To my advisor Sylvain Malacria and my director Professor Stéphane Huot, thank you both for your vast reserve of patience, knowledge and long-term support in guiding me through this research.

To Professors Nadine Couture and Kris Luyten, I am deeply grateful to both of you for providing helpful and precious suggestions on reviewing my dissertation. Your various pieces of advice are precious for me and to this research, I shall carry them on. To Professors Caroline Appert and Laurent Grisoni, thank you both for participating in this journey right from the beginning of this thesis, for giving me good reflections enlarging my vision from the first year of my PhD (Caroline) and for participating in the experiment (Laurent). My deepest gratitude for all your thought provoking questions and suggestions during the defense.

I would also like to thank my colleagues from Team Loki and Team Mjolnir, where I get to work with the best ambiance and the best discussions: Marcelo, Ed, Géry, Fanny, Thomas, Mathieu, Karine, Julie, Jonathan, Alix, Justin, Amira, Thibault, Christian, Sébastien, Hakim, Axel, Mark, Philippe, Grégoire, Raiza, and all the remarkable persons I have met in Inria Lille, Gwenaël Desfontaines, Anne Rejl, Charlotte Hespel, Marion Blasquez, Alice De Raphaël, and especially Marie-Agnès Énard and Caroline Van Hoy (who made possible the interactive paper poster at IHM 2017). All your thoughtful help, interesting discussions and hanging-outs have made working with you these four years a great pleasure.

I would also like to extend my deepest gratitude to my beloved family and friends around the world. To my family in Taiwan, US, and France: my mum Yi-An and my brother Shuo-Hsiu Hsu for making themselves very good examples for me to carry on in the field of research and their countless support and encouragement alongside

all my studies in France; my sister Chia-Jen and my brother in law Ted Ku, my dear cousins Christine and Lucia Hsu along with my aunt in the States. Also to the Pillias family: Anne-Françoise, Camille and my love Clément, thank you for your endless support. To my friends around the world: To Can Liu, Jessalyn Alvina and Isabelle Sage, thank you for your encouragements and warm discussions during the most daunting moments of my PhD. To Shin-Yi Yu and Estelle Garenaux, James Eagan and Intissar Abdelhak, thank you for all the support that you gave me. To James JW Lin and Yung-wen Liu, thank you both for your warm encouragement and your advice that accompanied me through this thesis.

Much grateful thoughts to the persons who introduced me to the fields of computer science and game design, and led my research reflection in good direction: Professor Stéphane Natkin, Professor Pierre Cubaud, and the very regretted Alexandre Topol. To Wendy Mackay and Gilles Bailly, thank you for giving me a chance to present my ideas and encouraging me pursuing research in HCI. Also my appreciation to my colleagues and students in Asia University, Tatong University and National Taiwan University of Science and Technology in Taiwan: you made me realize how much I love teaching and this was an important motivation for getting a PhD.

I am also deeply indebted to my friends Miowowo Kuo (萬磁木), Chia-Chia Yang (楊家家), Chiafen Tsai, Olga Chang, Jenny Chang, Natalia Yu and all the friends in the 六喵軒 group, who were an endless source of what a PhD student needs the most: fun, cat, food and drinks distractions. They have to be thanked for this as well as for their dedicated and much more serious participation in the online survey they made possible and valuable, altogether with the Reddit members and administrators of /r/iphone and /r/ipad.

Finally, I want to thank my little « roi du monde », my cat Gucci, who has accompanied me from my master study in France all the way to my PhD, the best cat in the world.

CONTENTS

1 INTRODUCTION

The principle of visibility is violated over and over again in everyday things. In numerous designs crucial parts are carefully hidden away. Handles on cabinets distract from some design aesthetics, and so they are deliberately made invisible or left out. The cracks that signify the existence of a door can also distract from the pure lines of the design, so these significant cues are also minimized or eliminated. The result can be a smooth expanse of gleaming material, with no sign of doors or drawers, let alone of how those doors and drawers might be operated. Electric switches are often hidden: many electric typewriters have the on/off switch in the rear, difficult to find and awkward to use; and the switches that control kitchen garbage disposal units are often hidden away, sometimes nearly impossible to find.

Many systems are vastly improved by the act of making visible what was invisible before.

Donald A. Norman,
The Psychology of Everyday Things, 1988.

1.1 OBSERVATIONS

At the heart of this thesis is a common but problematic situation that users of digital systems often face in their daily interactions. This situation can be characterized simply like this: to interact with the system, users need some knowledge of an interaction possibility, some piece of information about the interface, but this information is not provided in the context in which the user needs it. Or, to paraphrase Norman: the information about a possible interaction is not "in the world" when it is needed and has therefore to be "in the head" of the user [Nor88]. In such situations, users may be confused about what they should do and how to achieve their goals; or they might have to use sub-optimal ways of achieving their goals because they are unaware of the existence of more efficient options.

The quote introducing this chapter comes from The Psychology of Everyday Things by Donald Norman [Nor88], a book that certainly had a great influence on the evolution of user interface design practice. In particular, Norman advised to expose interaction possibilities by making things visible. To do so, he discussed design concepts such as *affordances* (design elements that directly convey interaction possibilities to users), *mappings* (that create a similarity between an action and its result), and *constraints* (that limit possible actions to the meaningful ones). These concepts have been widely discussed by HCI researchers, who also proposed related notions such as *feedforward* (that informs about the purpose or consequence of an action) [Ver+13]. Norman finally proposed to simply call "signifiers" any design element of a system that conveys information about possible interactions with the system [Nor08]. Signifiers have become popular concepts for UI and product designers, taught in many popular introductory books to the field [Shn97; Wei11; LHB10; Coo+14; Spo08]. Ignoring these concepts, with the consequence of hiding some interaction possibilities, would have been perceived as a design flaw by most designers for decades after the first publication of Norman's book. I call *signifier-less* the designs that provide such non-signified interaction possibilities to users, which I will define more formally and study in this thesis.

Despite a general awareness by designers of the issues they can cause, non-signified interaction possibilities have been provided by the user interfaces of most systems in the past, and are still frequently used today. In desktop computer interfaces, signifier-less designs include: contextual menus revealed by clicking with the right mouse buttons on some objects (although nothing signifies which objects will provide specific entries in the menu); fast navigation keys such as using the tabulation key to navigate to the next UI element that can receive keyboard input, or using arrow keys with modifiers to move to the end of a word, line, or the end of the text; and hotkeys such as the famous `Ctrl-C` to copy, which are not signified when the user needs them, but only hinted in the application menu when users explicitly search for them or when they do not need the information anymore because they are actually using the menu to trigger the command. In the interfaces of handheld touch-based devices, short contextual menus may be triggered by a long press (or by pressing the screen harder), navigation in larger-than-screen lists and images is done with gestures such as pinch or flick but nothing in the interface informs users they can do so, and interface objects and screen bezels can be swiped to reveal controls that were hidden outside the screen. Even physical interfaces as simple and ubiquitous as a button may lack signifiers [Jan14]. For instance, we may understand that to get a coffee, we need to push the most prominent button on a coffee machine, and yet be unsure if the coffee will stop flowing when we release the button, when we press it again, or automatically after some predetermined time – of course, the same problem exists for virtual buttons.

As signifier-less designs are common in user interfaces despite designers' awareness of the problems they may cause, these designs deserve a deeper analysis than simply advising to avoid them. Indeed, there might be good reasons to use them: maybe they provide some benefits that are hard to see with our current understanding of these designs, or maybe there is no way to avoid them. In both cases, designers lack guidelines to decide when to include non-signified designs in the interfaces they create and how to minimize the problems it may cause. This thesis aims at providing theoretical concepts and user studies to better apprehend the issue of creating signifier-less designs, focusing particularly on swipe gestures that reveal non-signified widgets in touch-based handheld devices.

1.2 A CONTEXT FAVORING SIGNIFIER-LESS DESIGNS

The current industrial and scientific context favors the creation of signifier-less designs in three ways that I will describe in the next subsections: a diversification of interactive devices that put stronger constraints on possible signifiers, the emergence of new design philosophies in the last decade that puts usability concerns in new perspectives, and a lack of publicly available data and knowledge about how users adopt signifier-less designs.

1.2.1 *Diversification of interactive device types*

Today, with the broad availability and acceptance of devices like smart watches, communicating furniture in "smart houses", wearable computing, tangible computing toys, etc., designers face a diversification of digital devices and have to exercise their design skills far from the well-known model of the graphical interface on a screen which size can be expressed in inches. Some devices have simply not enough pixels to display signifiers, others must be operated in environments with a lot of disturbance or noise, or without even looking at the device. While researches are being conducted to create new modalities for signifiers, such as using device temperature or haptic clues, designers may still have to rely on signifier-less designs.

With the popularity of handheld devices like smartphones and tablets, designers are also incited to integrate in the applications they create for these devices most of the features that are available on the desktop computers versions of these applications. As these devices have limited output resources to provide signifiers for all these features, and since the techniques that provide hierarchies of features – like menus – are not well suited to these devices, designers may prefer to only signify the most basic interaction possibilities and use signifier-less designs for the more advanced ones.

1.2.2 *Design philosophies*

Designers put a high value in the aesthetics of the products they create, and while this is often misunderstood for a simple will to make things look nice, designers' aesthetic sensibility actually encompasses many aspects of the design, including usability concerns. Yet, what constitutes a good design, including designers' perspective on usability, is often defined by higher-order design philosophies, which evolve with time. In the next paragraphs, I present three design philosophies that became popular in the last two decades and had a great impact on the emergence of some modern signifier-less designs.

NATURAL USER INTERFACES (NUIS) Touchscreen-based smartphones and tablets became mainstream after the release of the iPhone in 2007, and this popularity was built in part on advertising campaigns that praised the simplicity and "naturalness" of some gestures (mostly swipe and pinch), which in practice were not signified to the user.

These interactions were claimed to be "natural" in that they rely on skills that users already possess from interacting with the real world, and users should thus come up with such gestures by themselves without any additional "hint" [Jac+08; WW11]. In this sense, NUIs focus strongly on usability concerns, since they bring the premise that anybody can use an interface without training or even wondering how to do things. Some signifier-less designs may thus be the consequence of designers' belief that the interactions involved in these designs are so obvious that they do not need to be signified to users, and that using them in the design will improve the usability of the interface.

Yet, despite many papers presenting new interaction techniques and interfaces claimed as "natural", HCI researchers have also criticized the NUI design philosophy for the lack of evidence supporting its fundamental assumptions, from the lack of evidence that users effectively use spontaneously the interaction techniques claimed as natural and the lack of agreed methods to measure naturalness [AHN17], to the fact that some similarity between physical and digital interactions does not systematically imply that skills can be transferred from the physical to the digital world [BR15], and up to the fact that the skills in question may not be universal and innate but might instead be cultural constructs learned by people [Nor10; Hor13].

FLAT DESIGN The visual design of touch-based GUIs evolved in the mid-2010s towards the so-called "flat design". Inspired by the Bauhaus design movement, and particularly the "international typographic style", flat design follows a philosophy that seeks harmony between the function of some product and its design, rejecting the use of decorations or non-functional elements to create aesthetic value.[1]

In particular, it rejects *skeuomorphic* designs that create familiarity with a new product by mimicking aspects of other products although such features are non-functional and not strictly needed in the new product – which was a type of design used by the first generations of iPhones. Graphical elements mimicking real-world materials and depth/elevation of surfaces were rejected for this reason, hence the

1 https://en.wikipedia.org/wiki/Flat_design

name “flat design”. However, these elements were also sometimes affordances according to Norman, as a button that seems to protrude signifies it can be pressed, and a hollow field signifies it can receive input [Gav91]. Together with the idea of natural user interfaces, such removals opened the way for the removal of other signifiers and may have contributed to a lesser attention to signifying interaction possibilities, going farther than what strictly belongs to the flat design philosophy.

UX DESIGN Inspired by industrial design practice, and as a reaction to the main focus on usability in User Interface Design, the concept of User Experience (UX) design gained popularity in the 2000s. At its core, UX assumes that usability is only one facet of the experience that people have with products, and that products are also meaningful to people by the emotions they convey, the place they have in people’s life, and how they support changes in the lives of their users (including self-improvement) [Nor04].

A popular framework thus distinguish between pragmatic and hedonic attributes of a product [Has03], where the pragmatic qualities concern the achievement of behavioral goals and include usability concerns, but the hedonic qualities relate to users’ self –Why I use this product?– and can be subdivided into "stimulation" and "identification". Stimulation reflects the basic human need of self improvement and relates to the challenges and the novelty of using such products, while identification addresses the human need to express one’s self through products.

With a focus on taking all aspects of UX into consideration when designing a product, trade-offs may appear between different qualities, and sometimes, usability concerns may be perceived as less important than hedonic ones. In the case of signifier-less designs, an interaction possibility may thus be voluntarily hidden to users so that its discovery –however it happens– is an event scoring high on the stimulation axis, contributing to building a good user experience in the long term [Kar+09].

1.2.3 *Lack of knowledge about how users adopt signifier-less designs*

Unfortunately, there are very few studies and very little public data available to even start questioning or validating the design rationales behind modern signifier-less designs. In the next paragraphs, I will highlight three areas in which basic knowledge is missing: how users adopt signifier-less designs, how they discover them without signifiers, and how such designs affect users.

USER ADOPTION OF SIGNIFIER-LESS DESIGNS Studies on user adoption of signifier-less designs are very few but show that users have different levels of knowledge and actual usage of these techniques – i.e. different levels of *proficiency* in their use, between completely ignoring them and perfectly mastering the art of how and when to use them.

A study of iPad multitasking gestures adoption by Avery and Lank showed that despite being non-signified, efficient gesture-based techniques can get high levels of user awareness and willingness to use [AL16]. On the other hand, it also showed that users "often find [the gestures] difficult to discover and challenging to learn". Studies that compared hotkeys to other means of invoking commands in Microsoft Word [Lan+05; TWV13] revealed the following user behaviors: 1) belief of rarely using hotkeys, despite observed usage actually ranging from sporadic to almost systematic depending on the type of command the hotkey invokes; 2) users frequently missing opportunities to use the hotkeys they know; 3) perception of hotkey efficiency that is often wrong and depends on the command it invokes; 4) failure to use the technique they believe to be the fastest among the ones they know, even under time pressure.

Such studies call for further analyses and need to be conducted on more types of signifier-less designs. They also show that users have complex usage patterns of signifier-less designs –especially when considering generic designs like hotkeys that can have multiple distinct realizations in a same interface. This complexity is also an obstacle to the evaluation of signifier-less designs, since it makes the design of informative studies harder.

DISCOVERY OF SIGNIFIER-LESS DESIGNS In recent decades of HCI research, numerous studies addressed how users gain expertise as they use the interface, how they learn and what keeps them from becoming more skilled [Coc+14]. Yet there is almost no data explaining how users *discover* an interaction possibility before they "learn" to master it. Most HCI works simply assume that techniques are discovered, thanks to signifiers or exposure of the controls, preexisting skills or metaphors, or simply because they belong to a small core of basic knowledge that every users should be taught quickly before being able to use the system – including knowledge that has become a cultural evidence, like how to hold a mouse [Shn97; GFA09].

USER REACTION TO SIGNIFIER-LESS DESIGNS Very few research works investigate how signifier-less designs affect users beside the impossibility to use these interaction techniques, and users' frustration of not being able to accomplish their tasks [Nor88]. There is a lack of observations about how users react to the discovery of a signifier-less design, if they appreciate this design and consider using it systematically as their preferred way to accomplish a task, or if they feel fooled for not having been told about it – or incompetent for not discovering it by themselves. There is also a lack of studies assessing if users can adapt and build on their new knowledge of this design to discover other similar non-signified interaction possibilities, integrate this knowledge in a logic of their own, envision new usage patterns, and eventually, see opportunities to change the way they think about their daily tasks for more efficient ones.

1.3 APPROACH AND CONTRIBUTIONS

1.3.1 *Approach*

Despite the well-known issues they cause, signifier-less designs have maintained a continuous presence in successive generations of user interfaces. There must be a reason for this persistence, possibly that users appreciate these designs, or maybe designers think users will use or appreciate them. My approach during this thesis was thus to study such designs from a few different perspectives, not focusing only on their known issues.

A first aspect of my work was to investigate the design space of signifier-less designs. I reviewed and analyzed historical and modern signifier-less designs to identify a few different dimensions. I particularly investigated a modern smartphone and tablet design in which widgets are hidden on the side of the screen and revealed with a non-signified swipe – I call this design *swhidgets* for *Sw*ipe-revealed *hid*den *widgets*, and will provide a more complete definition of it in the next chapter. This investigation revealed different types of *swhidgets*, different roles for *swhidgets*, and a strong integration in the global design of the interface. Finally, I have also devised a categorization space about how users may discover or learn about signifier-less designs in the first place.

A second aspect of my work is the realization of two user studies to improve knowledge about how users discover, adopt and react to signifier-less designs – focusing more particularly on *swhidgets*. A first laboratory study allowed me to observe face to face how the participants react to –and generally think about– *swhidgets*. An online questionnaire with a larger and broader population of users allowed me to know more in detail how users discover, practice, think of, and feel about *swhidgets*. From these two studies, I draw some implications for new *swhidget*-like signifier-less designs and discuss some opportunities for improved design.

1.3.2 *Contributions*

During this thesis, I studied the question of why designers would create interfaces that do not clearly expose some of their interaction possibilities, taking the case of *swhidgets* as an example and focus of inquiry. My contributions to this research are:

- An analysis of the fundamental notions required to define precisely this research question, including affordances, signifiers and semiotics.
- The definition of *signifier-less designs* and the observation that this type of design is common in user interfaces and deserves more attention from HCI research.
- A model of user discovery and adoption of interaction techniques, relying on the categorizations of three dimensions and their interactions: user's current knowledge and skills, user's motivations, and the design means of informing users provided by the interface.
- The notions of *Degree of Knowledge* and *Sources of Knowledge* derived from this model, that can be instrumented in experiments to evaluate respectively how well participants know an interaction technique and how they discovered it.
- Two studies on *swhidgets* that investigated how well users known them, how they discovered them, how they feel about them, and how they integrate them in the way they think about the interaction. These studies revealed that *swhidgets* were globally appreciated and relatively well known by users, although there is still room for improvement, notably for some specific *swhidgets*.

1.4 PLAN OF THE DISSERTATION

This dissertation is divided in four parts:

INTRODUCTION AND SWHIDGETS To complete this introduction, Chapter 2 provides a more detailed example of signifier-less designs and exposes the diversity of problems and research questions raised by their study. It describes *swhidgets*, the modern signifier-less design that this thesis mainly focuses on. I categorize different types of *swhidgets*, analyze their roles in the interface, and discuss their discoverability and lack of signifiers in relation to the state of the art and other signifier-less designs. The chapter ends with a presentation of research questions that needs to be addressed to fully understand *swhidgets* and other signifier-less designs, defining a framework for this research and future researches, although not all of these research question can reasonably be addressed in this thesis.

FUNDAMENTAL NOTIONS AND DISCOVERY OF INTERACTION POSSIBILITIES The first part of this thesis aims at providing essential definitions for the main concepts involved in the study of signifier-less designs – notably *interaction possibilities* and *signifiers* – while also discussing how these concepts relate to the research questions identified in Chapter 2. In particular, this part provides a state of the art of how users may discover interaction possibilities. It starts by defining precisely the notion of interaction possibility as Gibsonian affordances in Chapter 3. Then, in Chapter 4, I present Norman's Seven Stages of Action, a common HCI model of how users interact with the world and interfaces, and discuss how it accounts for user discovery of new interaction possibilities. Finally, Chapter 5 describes the means by which design elements of the user interface – mostly signifiers – can help users discovering new interaction possibilities, the different types of such "design means", and how they are used in the different stages of Norman's model.

SIGNIFIER-LESS DESIGNS The second part of this thesis introduces signifier-less designs in Chapter 6, proposes a definition based on the concepts of affordances and signifiers, discusses their history and how their understanding evolved over time, to finally distinguish three sub-types of signifier-less designs depending on what affordance is not signified.

A MORE COMPLETE MODEL OF DISCOVERY In the third part of this thesis, I propose models of how users may discover non-signified interaction possibilities and develop their interaction vocabulary to become expert users, to extend the state of the art presented in the first part. In Chapter 7, I describe the necessary conditions for the discovery and adoption of new interaction possibilities, considering a user's current knowledge and motivations, and how they affect or are affected by the design means already discussed in Chapter 5. I introduce *degrees of knowledge* as a measure of a user's current knowledge that is suitable for experimental studies. A characterization space of the possible sources of knowledge of an interaction possibility is also provided as a contribution, leading to a discussion of user exploration of the interface.

STUDIES The fourth part of this thesis presents in Chapter 8 the design and results of two studies investigating the knowledge of *swhidgets* by iOS users and their reactions to this design: a preliminary laboratory study and an online study that complements the first study with more precise investigations of what iOS users know about *swhidgets*, how they discovered them, how they feel about them, and how they integrate them in their way of thinking the interface. I discuss the results of these two studies and draw implications for designs that include or get inspiration from *swhidgets*.

PERSPECTIVES The dissertation ends with a discussion of perspectives and future work, in which I highlight some possible research and design directions to improve signifier-less design in general and *swhidgets* in particular.

2 SWHIDGETS: DESCRIPTION, ANALYSIS, AND RESEARCH QUESTIONS

It is now common for smartphone users to slide a finger from the edge of the screen toward the center of the screen to drag into view the hidden drawer that was secretly hiding under the bezel. I define *swhidgets* (for *sw*ipe-revealed *hid*den *widgets*) as widgets that are normally entirely hidden from the user, who can uncover them with a simple swipe gesture relying on the metaphor of sliding physical objects to drag the *swhidget* into view.

The use of a physical drag metaphor is a distinctive feature of *swhidgets* that may provide the type of user guidance discussed by Norman as lacking from other gestural interfaces [Nor10] and builds over the design of navigation gestures omnipresent in modern smartphone interfaces. As such, *swhidgets* differ from swipe-based interactions techniques that do not rely on a sliding metaphor such as Bezel Menus or Bezel Taps [JB12; SLG13].

In this chapter,[1] I analyze *swhidgets* from popular commercial products, examine the apparently voluntary lack of elements signifying their presence, and discuss how users can discover them. While *swhidgets* can be found on various touch-based operating systems, I will focus on the iOS platform as this is a popular and consistent operating system without additional manufacturer overlay, relying on simple physical metaphors for interaction [App19]. However, the categories proposed in Section 2.1 are still likely to also apply to *swhidgets* on other platforms.

2.1 TYPES OF *swhidgets*

Surveying how *swhidgets* are used in iOS interfaces, I recognized three different types of *swhidgets*, each relying on different interface schemes that can be classified from the perspective of the physical metaphor they use and the type of functions they provide, as follows:

SYSTEM SWHIDGETS are "pulled over" the main view from an edge of the display, following the metaphor of a roll-up projection screen. Examples of *system swhidgets* are the Notification Center and the Control Center (Figure 2.1), which are revealed with sliding gestures from (and across) the edge of the display – respectively downward

1 Sections 2.1 to 2.5 of this chapter have been partially published as parts of a paper for the ISS'19 conference [PM19].

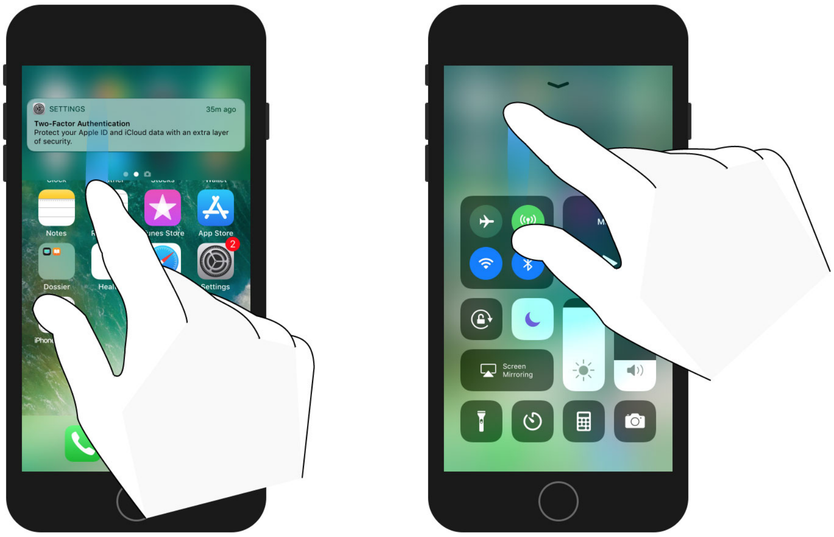


Figure 2.1: Examples of *system swhidgets* in iOS containing various controls that can be accessed from anywhere in the OS.
Left: a downward swipe from the top edge of the screen reveals the Notification Center.
Right: an upward swipe from the bottom edge of the screen reveals the Control Center.

Figure 2.2: Examples of *view* and *item swhidgets* in iOS containing various controls that are provided by some applications.
Left: A *view swhidget* in the Message application, where a leftward swipe from anywhere in the discussion view reveals the exact time each message was sent.
Right: An *item swhidget* in the Mail application, where a leftward swipe on an e-mail from the list reveals a set of buttons.

from the top and upward from the bottom.[2] *System swhidgets* can be revealed regardless of the application/view[3] and overlay the entire main view to provide various system commands and information.

VIEW SWHIDGETS are concealed on the side of a view or container, outside its viewport, and revealed by pulling them into view, i.e., by sliding the finger in the opposite direction of the *view swhidget*. Examples are the exact arrival time of SMS messages, revealed with a leftward sliding gesture from anywhere in the discussion view (Figure 2.2-left), or the search text field that is located at the top of most element lists (typically e-mail list in the Mail application) but hidden by default, only to be revealed when scrolling over or back to the top.

ITEM SWHIDGETS are hidden next to or under an object of interest within the application and provide contextual functions to interact with this object. Examples are the commands operating on e-mails in the Mail app, that can be revealed by sliding horizontally an individual e-mail (Figure 2.2-right). *Item swhidgets* differ from *view swhidgets* in the sense that their revealing gesture must be performed on a specific element.

2.2 ROLES OF *swhidgets*

Swhidgets are used conjointly with other types of controls in the design of mobile device interfaces and cannot be understood fully if taken in isolation. One question is thus: do *swhidgets* replace, complement, or double other interaction techniques?

Most *swhidgets* provide a faster access to some functionalities that can also be accessed through traditional widgets.[4] Yet, some *swhidgets* provide access to advanced features with no real equivalent, and where the conventional *tap and scroll* interaction paradigm would only provide access to limited versions of the feature. In iOS, this is typically the case of exact SMS arrival times that can only be accessed through a *view swhidget*, whereas conventional interaction would only provide discussion start and resume times. Similarly, the search feature in the Mail application is not accessible through another widget than the

2 On the iPhone X and iPad, both are revealed with a downward swipe, starting respectively from the left and right sides of the top bezel.

3 In some full-screen applications like games and video players, a protection against an involuntary activation of the *swhidget* is provided, in which the swipe gesture first only displays a small un-intrusive handle, and a second swipe gesture needs to be performed to confirm and trigger the opening of the *system swhidget*.

4 Often via a modal interface triggered by a button with a generic label like "modify" or an icon representing gears.

dedicated *view swhidget*,[5] but users can still search a mail by browsing the mail list.

In some cases, *swhidgets* are merely one of several ways to accomplish a task, which relative efficiency can depend on how users prefer to accomplish the task. For example, on an iPhone 6S running iOS 11, users have at least seven different ways to delete an e-mail in the Mail application:

1. From the list of mails/threads in a mailbox, by performing a leftward swipe on the mail/thread to delete (as shown in Figure 2.2-right):
   a) A short swipe to reveal an *item swhidget*, then tap the rightmost icon it contains (a red trashcan labeled "Delete").
   b) A short swipe to reveal the same *item swhidget*, then tap on the leftmost icon it contains (blue with three dots labeled "More" and suggesting "more options"), which opens a menu at the bottom of the screen, from which the user can choose "move to...", and then select the trash mailbox.
   c) A long swipe leftward to directly delete the mail.[6]
2. From the list of mails/threads in a mailbox, by tapping the "Edit" button, then selecting the mail(s) to delete, then tap the "Delete" button.
3. From the view to read a specific e-mail or thread:
   a) By taping the trashcan icon at the bottom of the screen.
   b) By taping the folder icon at the bottom of the screen, which opens a list of the mailbox and folders, then choosing the trash folder.
   c) If there are responses to the mail, it is a conversation view rather than a mail view, and users can then make a leftward swipe anywhere on the body of an e-mail to reveal three icons on its right, and use the rightmost one with a trashcan icon.

Here, users may prefer to read every incoming e-mail individually before eventually deciding to delete it with the trashcan icon at the bottom of the screen (option 3a), which deletes the mail and automatically opens the next one. Or they may prefer to first delete all irrelevant e-mails directly from the e-mails list, either using a swipe that directly provide feedback about the deletion (options 1a or 1c), or using the modal "modify" button to delete all mails in a batch (option

5 The search feature in mail is also accessible through Siri, which relies on a different interaction paradigm

6 This interaction can be seen as an extension of the *swhidget* design, an "expert mode" to trigger the main command in the *swhidget*. Although it is an interesting design that deserves a study, in this thesis I focus first on the basic *swhidget* design.

2). The preference, here, can depend on multiple factors: a user's expertise with the interface and e-mail management activities, the amount of unwanted e-mails that she receives, or external constraints affecting the ease of performing some gestures that can be imposed by the context of interaction (e.g., in a crowded subway train, or while walking).

2.3 DO USERS INTERACT WITH *swhidgets*?

It remains unclear how many *swhidgets* users actually use and how they are discovered. To the best of my knowledge, there is very little public data on current usage of *swhidgets*. The only available data set I am aware of can be found on a blog post reporting proportions of users (among 6 tested) who knew how to interact with a subset of *system swhidgets* in iOS 11 [Ram17], which suggests there are great differences in the knowledge of the different interactions, some being spontaneously used by all participants and others being understood by none of the participants. Unfortunately, the author does not accurately describe the methodology and experimental procedure employed, questioning the generalization of the results.

Academic research has mostly investigated the performance of novel edge-based interaction techniques, such as menus or keyboards [JB12; RT09; SLG13] that do not rely on a physical metaphor that may help users uncovering these features. Similarly, Schramm et al. [SGC16] investigated the transition to expertise in *hidden toolbars*, a name describing various types of user interfaces that make functionality available only when users explicitly expose them through a dedicated interaction, including menu bars (revealing commands when the menu bar is clicked), the "charms" in Windows 8 [Hop19], and some *swhidgets* (typically, *system swhidgets*). Their work focuses on the performance of such interfaces, proposing four hidden toolbar designs (all different from actual *swhidgets* designs) and comparing their performance in terms of selection time and learning of item locations. As a result, their work is informative regarding the performance of the hidden toolbar designs they propose, but is not adequate to fully understand how users currently interact with *swhidgets*.

Similarly, studies conducted on other related "hidden" or "expert" interaction techniques can help apprehending interaction with *swhidgets*. This is the case of the work conducted by Avery and Lank surveying the adoption of three multitasking multi-touch gestures on the iPad [AL16]. Their results show that users tend to have varying levels of knowledge and actual usage of these techniques, between completely ignoring them and perfectly mastering the art of how and when to use them. They also show that despite being "hidden", these gesture-based techniques can get high levels of user awareness and willingness to use. Regarding keyboard shortcuts on desktop, studies conducted on Microsoft Word [Lan+05; TWV13] revealed that users do not systematically use the hotkeys they know. This behavior may be explained in part by the fact that users often underestimate the potential benefits of hotkeys. This is however not the whole explanation, since users also often fail to use hotkeys when they know it will be faster, even under time pressure.

2.4 VOLUNTARILY SIGNIFIER-LESS: WHY?

By definition, *swhidgets* are instances of *signifier-less designs*, since their design provides no *signifier*, i.e., no perceptible indicator of the presence of the *swhidget* or of the possibility to make a swipe to reveal it (the notion of signifiers will be introduced in a broader context in Chapters 3 and 5, and the notion of signifier-less design in Chapter 6).

Some signifier-less designs lack of signifiers because there seems to be no satisfying way of permanently displaying signifiers where the input method can be used [May+18]. This is however not the case of *swhidgets*, where a handle could easily be displayed on the movable interface elements to provide a hint of their presence [RT09]. Indeed, a small dash-shaped handle used to be displayed near the bottom edge of the screen when the control panel was introduced in iOS, but more recent releases show it on extraordinary occasions only (see Footnote 3), or use it as a replacement for the physical home button (typically on iPhone X, which has no physical home button).

Swhidgets are by default "hidden" under the screen bezel, yet their existence is not signified to users, possibly as an intention to minimize visual clutter [SGC16] and keep the UI clean in order to improve user experience and performance [Nor04]. The iOS *Human Interface Guidelines* echo these expected benefits in the themes of *clarity* and *deference*: "Content typically fills the entire screen, [. . .] minimal use of bezels, gradients, and drop shadows keep the interface light and airy, while ensuring that content is paramount" [App19].

This focus on "clean" interfaces benefits all users, but *swhidgets*-aware users might additionally benefit from other aspects of the design. For instance, it could be said that *swhidgets* provide sometimes a more uniform way to interact with the system globally than alternative designs, since they rely on similar gestures and metaphors than navigation gestures. However, none of these benefits seems to justify a total removal of signifiers. On the other hand, the discovery of a *swhidget* could also be a positive experience for users and such a positive experience may justify the lack of signifiers, although there currently seem to be very little evidence in the HCI field supporting such an hypothesis.

2.5 DISCOVERY OF *swhidgets*

As pointed by Mayer *et al.*, the swipe gestures of *swhidgets* are not visually communicated to the users: Instead, device manufacturers rely on dedicated tips and animations that are either shown to the user when setting up the device, or showcased on stage when the system is presented to the technology-oriented press [May+18].

A way to discover *swhidgets* that is better integrated in everyday interaction is to explore the interface by performing series of inputs and expecting the system to respond to them. Schramm *et al.* [SGC16] advocate that with edge-based interactions becoming more common, users might appropriate the physical metaphor of sliding objects and become more likely to expect *swhidgets*, thus exploring the operating system to discover them. The design guidelines and *Design language*[7] (typically, *Material Design* on Android and the *Human Interface guidelines* on iOS) eventually declare a consistent interaction environment that promotes such metaphors and swipe-based interaction, which might lead users to discover these *swhidgets* through interactive exploration.

Moreover, *swhidgets* are by design exploration-friendly, since 1) they do not trigger commands directly but only reveal widgets, and it is thus safe to try revealing them; 2) testing the presence of a *swhidget* has a low interaction cost since it only requires to slide a little bit a finger on a candidate surface or edge; and 3) in case of success, this test provides immediate feedback with a visibly moving element.

However, Schramm *et al.* also acknowledge that one cannot expect novice users to guess the availability of *swhidgets* nor the actions used to access them regardless. Therefore, users still need to have discovered a first *swhidget* before being able to reproduce a similar input somewhere else in the interface. In addition, it still requires users to *explicitly* explore the interface in the first place, while they may be too engaged in their tasks to do it, even if it would eventually improve their performance as users [CR87].

Another possible way to discover *swhidgets* is through *accidental revelation*, that is, when the *swhidget* is revealed whereas it was not the first intention of the user. For instance, one of iOS *swhidgets* that is likely to be accidentally revealed is the search *view swhidget* located on top of some lists, that may be brought into view when the user overshoots while scrolling back to the top of the list. Other *swhidgets* may be less likely to be accidentally discovered. For instance, with the *item swhidgets* in the Mail application, there is no reason to expect users to perform an horizontal swipe on an item of a vertically scrollable list. Moreover, accidental activation can be confusing and users might be unable to reproduce the input operation that triggered that accidental activation.

7 https://en.wikipedia.org/wiki/Design_language

Finally, users might discover the existence of *swhidgets* from their own social network, typically through friends, colleagues or family. Indeed, impact on input mechanism adoption of witnessing others performing them has already been observed in the context of keyboard shortcuts [Per+04].

2.6 GENERAL RESEARCH QUESTIONS IDENTIFIED

In this section, I will recall and formalize the questions that were quickly raised about the design of *swhidgets* in the previous sections. These questions point at individual phenomena that need to be understood before being able to get a full understanding of the design of *swhidgets*. However, the scope of these phenomena extends far beyond the single case of *swhidgets*, and understanding each of theses phenomena is a general topic in the field of HCI. It is therefore not possible, in this thesis, to address all these phenomena, nor to contribute directly to the understanding of all of them. The list of phenomena and research questions that I give below is therefore defining a framework into which research on *swhidgets* has to be conducted, and highlights the benefits of studying *swhidgets* to improve general knowledge about human-computer interaction as a phenomenon [Bea04; HO17]. The next chapters will discuss basic notions implied by this framework.

2.6.1 *Integration of swhidgets in relation with other interface features*

We have identified three types of *swhidgets* (Section 2.1), in which specific interaction techniques (position and semantic of the swipe gesture) match with domains of application of commands: *system swhidgets* provide system-wide commands and are attached to the screen bezels, a feature of the device itself; *view swhidgets* provide commands about the set of data displayed in a view and are revealed by scroll-like interactions with this view; *item swhidgets* require to interact with the item targeted by the command. This matching raises two types of questions, depending on whether its role is considered in a specific instance of interaction or in multiple instances of interactions across different applications.

OVERARCHING INTERFACE LOGIC This matching between interaction techniques and domains of application of commands is an overarching logic in the structure of the interface, a general principle ruling the design of the whole operating system and specific applications. It suggests that users can learn this overarching logic (Section 2.5), and later use it to deduce where they should search for a specific command, even if the widget providing this command is hidden. Do users really exploit such an overarching logic? How do they learn it, and how do they think about exploiting it? What cognitive processes are activated to exploit this knowledge? These questions fit in the general HCI research about how users develop a conceptual understanding of the interface (as rules, strategies, mental models, etc.) and exploit them in their goal-oriented, problem solving interactive behaviors [Nor88].

CONSISTENCY WITH OTHER INTERACTIONS The overarching interface logic is only one aspect of a consistent use of *swhidgets* in the design of the system and applications, but other aspects of the design can also be consistent. For instance, the mailbox view in the Mail application (Figure 2.2-right) and the conversations view in the Message application both share a similar list layout with similar *item swhidgets*, although there are slight variations between the two designs. Can users notice this similarity, infer a pattern from it, and use their knowledge of the pattern to discover *swhidgets* in other applications? Are there aspects of the design, of the activity supported by an application, or user skills that can help or prevent noticing the pattern or similarities with other interfaces? Beyond the consistency of uses of *swhidgets* in design, what role plays the consistency of *swhidgets* with other types of interactions such as navigation gestures (Section 2.4), or the consistency of patterns that involve both *swhidgets* and other types of controls? How important is a consistent use of *swhidgets* in design for their adoption by users? These questions fit in the general topics of transfer of knowledge from other contexts of use, and other benefits of consistency in user interfaces [Gru89].

BENEFITS OF A CLEAN INTERFACE Since the design of *swhidgets* seems to purposefully avoid the use of signifiers (Section 2.4), and since a possible reason for this is to reduce interface clutter, improve readability, and focus on content, it seems important to understand the benefits of having a "clean" interface and how these benefits are obtained.

2.6.2 *Interaction with specific swhidgets*

ACTION-FUNCTION COUPLING Beside the existence of an overarching interface logic, is there something specific about how *swhidgets*' designs connect the semantics of the interactions with the semantics of the commands they trigger? *Swhidgets* use an interface element that connects the domain of commands and the interaction technique by being more or less loosely related to both: a bezel, a view, or an item – but does the strength of this relation affects users' ability to understand it or to integrate this understanding in how they think about *swhidgets* and commands? Behind these questions, there is the whole domain of HCI research about the consequences of relating the purpose or effect of an interaction with the way it is performed [WDO04; Dja+04].

PHYSICAL METAPHOR *Swhidgets* rely on the physical metaphor of sliding interface objects to reveal what is under them. Unlike action-function coupling, such metaphors are concerned with the way the interaction if performed but not with the effect of the interaction or its purpose (beside "revealing some widgets"). As such, they fit in the general study of metaphors as a tool to help users make sense of the interface and learn how to use it by themselves.

However, in the case of *swhidgets*, there is a little more than that, since the metaphor also contributes to unify the design of *swhidgets* with other types of interactions promoted by the system (Section 2.4), and is thus related to consistency questions.

The specific metaphor used for *swhidgets* also allows to bridge the gap between uni-stroke gestures and direct manipulation, with consequences on the learnability of the gestures or users' understanding of what they did or are doing (feedback). It therefore has to be analyzed both relatively to the problem of learning how to interact with a system, and to the problem of facilitating interaction itself.

2.6.3 *Discovery and adoption of Swhidgets in the long-term*

MULTIPLE WAYS TO DO A TASK *Swhidgets* can coexist with other methods for triggering the commands they provide, with sometime differences in performance or ease of use (Section 2.2). It raises the general questions of what methods users prefer to use, and of what aspects of the methods, task, and context affect these preferences. But it also raise questions about the evolution of such preferences in time as users get more familiar with them and better understand the advantages of each method. In particular, are some methods supposed to replace others as users get more experienced, and if so, what aspects of the design of each interaction method makes it better suited for different types of users?

REASONS NOT TO ADOPT The example of hotkeys (Section 2.3) reveals that measuring users' performance with an interaction technique and the proportions of users who know how to use it at a given level of performance is not enough to understand users' adoption of the technique: we also need ways to better explain the non-usage behaviors observed – including reasons for users not to use a technique despite performing well with it. A study of *swhidgets* should thus also evaluate users' perception of the technique's performance, ease of use, reliability, mental and physical costs of activation, etc., to determinate if an inaccurate perception of these qualities justifies the non-usage of *swhidgets*.

Beyond the evaluation of such factors in experimental studies, there are also opportunities to analyze how the design of the interface helps users evaluating these qualities of an interaction technique. Such analyses may have to be performed in relation with other aspects of the design – e. g., for interface consistency: the metaphor of sliding physical objects and its consistent use in *swhidgets* and other navigation techniques may help users making more accurate evaluations of *swhidgets*' performance and ease of use, by leveraging their knowledge of other parts of the interface that rely on similar gestures and animations.

USER EXPERIENCE OF DISCOVERY Because they rely on interaction possibilities that are not signaled to users, a major obstacle to the use of *swhidgets* is their discovery. But conversely, what does the discovery of a *swhidget* entail for the user experience? In addition to the benefits of being able to use a new interaction technique, what are the consequences of the discovery on users, especially from the point of view of emotions, value judgments, and other reactions? The study of user experience (UX) has become an important theme in HCI research, although it is often understood from the point of view of a specific situation (notably, customers' first contact with a design) rather than as developing in the long term [Haso3].

EXPLORATION OF THE UI Exploration of the user interface is one way to discover *swhidgets* (Section 2.5), but despite a long term interest in HCI for the phenomenon of user exploration of the UI, there seems to be a lack of models that can explain how and when users explore the interface, and how they could do it to discover hidden features, or what properties of the hidden interaction possibilities can encourage users to explore the interface and search for them.

EXTERNAL AND ACCIDENTAL SOURCES OF KNOWLEDGE Users may learn about *swhidgets* from other sources than the interface itself or through accidental revelation (Section 2.5). While HCI research with a focus on usability usually considers that the knowledge about the interface should be conveyed through or by the interface itself, the reality of interaction is that it happens in a cultural and technological context that provides users with multiple opportunities to learn about the interface, to the point that it can be questioned if this knowledge really needs to be also conveyed by the interface itself. What opportunities have designers to leverage external sources of knowledge, and what are their chances of success in doing so?

Conversely, designers also have opportunities to create interaction techniques that have greater chances to be discovered through accidental activation, or by otherwise exploiting interactions that users may perform without assuming they will be understood by the system.[8] Game designers, for instance, frequently force players in situations to which they will naturally react with some action, to teach players that these actions are meaningful in the game[9] – to what extent could UI designers rely on similar tricks, and what would be the guidelines to do it well?

2.6.4 *Methodological aspects of the study of Swhidgets*

UNDERSTANDING OF SIMILAR INTERACTION TECHNIQUES *Swhidgets* are quite unique interaction techniques in that they have both aspects of gestures and direct manipulation of objects, in addition to being hidden. It makes it relatively difficult to understand this design by comparing it with other well-known interaction techniques. For this reason, I have chosen in this thesis to extend the analysis of *swhidgets* to a larger category of interactions that are not signified (Section 2.3) – which turn out to be more frequent than I initially believed. However, to be able to compare *swhidgets* with other *signifier-less designs*, it will be necessary to define as precisely as possible the notions of signifiers and interaction possibilities – notions that are complex and have been the subject of many debates in the HCI literature, as we will see in Chapter 3. Doing so helped me identify some of the previous research questions in this list, and I hope that in turn, my contributions about *swhidgets* may suggest new approaches to the study of other signifier-less designs.

MEASURING PROFICIENCY Because *swhidgets* are a family of interaction techniques rather than a single one, it is difficult to answer the question "how well are they known by users". Indeed, not only different users can have different knowledge of *swhidgets*, but one user's knowledge can also vary for different instances of *swhidgets*. Knowledge itself is a vague term, as there can be various levels of awareness, understanding, ability to use or showing expertise in use. Knowledge can evolve in time and be closely connected to user's preferences, motivations, and activities practiced. For these reasons, I prefer to consider the notion of proficiency in use to evaluate the design of *swhidgets* and the consequences of their reduced discoverability due to the lack of signifiers.

8 An example of the latter is provided by a macOS feature: when the user do not find the mouse pointer, they tend to shake the mouse to spot the pointer thanks to this motion, but the system actually recognizes this interaction and then displays temporally a much bigger pointer to make it even easier to locate.

9 See for instance Egoraptor's popular and humorous YouTube video about the game Mega Man X: `https://youtu.be/8FpigqfcvlM`.

2.7 HOW I ADDRESS THESE RESEARCH QUESTIONS

Although all the general research questions presented in the previous section deserve dedicated researches, I could only address some of them in this thesis. I therefore defined the following sequence of specific research questions that could reasonably be addressed during a thesis, and needed to be addressed in this order:

1. What is a signifier-less design and how *swhidgets* are different from other signifier-less designs?
2. Do users know *swhidgets*?
3. If yes, how is it possible that they know *swhidgets* despite their lack of signifiers?
4. Even if there are reasons to believe users can get to know *swhidgets* despite their lack of signifiers, what are the benefits of not providing signifiers?

I will now give a quick overview of how each of these questions relates to the general research questions from the previous section, and what contributions I provide in this thesis to address them.

2.7.1 *What is a signifier-less design and how swhidgets are different from other signifier-less designs?*

In the general theme "Understanding of similar interaction techniques" (Section 2.6.4), I will provide in Chapter 6 a definition of signifier-less designs and a review of historical signifier-less designs.

For obvious reasons, this definition relies on the notion of *signifier*, which will be discussed in Chapter 5. There are two additional reasons to discuss signifiers in the study of signifier-less designs: First, it is important to discuss signifiers to understand what information they usually convey but is missing in signifier-less designs. Second, users' ability to interact with signifier-less designs could rely on their ability to learn recognizing as signifiers some aspects of designs that would initially be considered as signifier-less – a phenomenon related to concerns about "Consistency with other interactions" (Section 2.6.1). I thus propose to consider signifiers in the framework of semiotics (from which they originate) so as to understand how users can learn to recognize signifiers and associate them with interaction possibilities.

Since we are interested in signifiers that inform more or less directly about "interaction possibilities", this notion needs to be precised, and I claim it can be done through Gibson's notion of *affordances*. The definition of signifier-less designs that I propose relies on affordances for this reason, but also to precise the relation between the missing signifiers and all the interaction possibilities (affordances) that are connected to the one considered, either as intermediary steps or as the purpose of the interaction (e. g., opening a menu to select a command). Since the notion of affordance is complex and different definitions have been used in the HCI literature, it will be discussed in Chapter 3 from the point of view of some concerns that are relevant for signifier-less designs, including the meaningfulness of interaction possibilities for users, their learning and acquisition by users, how they relate to other affordances, and the possibility to interpret them.

2.7.2 *Do users know swhidgets?*

Answering this question is the main objective of the two studies that I will present in Chapter 8. To do so, and in the general theme "Measuring proficiency" (Section 2.6.4) – and in particular its implication "what does it mean to know a family of interaction techniques?" – I will propose in Chapter 7 the notion of *Degrees of Knowledge*, a simple characterization of how well a user knows an interaction technique that can be instrumented in experiments.

However, answering this question would be of little scientific value without a consideration for the phenomena that can explain users' lack of knowledge of *swhidgets*. I study two such phenomena in this thesis: First, the consequences of having "multiple ways to do a task" (Section 2.6.3) will be discussed theoretically in Chapter 7, and investigated in the studies of Chapter 8 as the knowledge of *swhidgets* is compared to that of other interaction techniques that can be used to complete the same tasks and users' preferences are evaluated. Second, "Reasons not to adopt" *swhidgets* (Section 2.6.3) will also be discussed theoretically in Chapter 7 and investigated in the second study of Chapter 8 with questions targeting specifically this concern.

2.7.3 *How can users know swhidgets despite their lack of signifiers?*

To answer this question, I first ask how the discovery of *swhidgets* can happen. I discuss in Chapter 4 the general theme of "Exploration of the UI" (Section 2.6.3) in relation with user's goals. In the theme of "External and accidental sources of knowledge" (Section 2.6.3), I also propose in Chapter 7 a characterization space of the *sources of knowledge* based on the dimensions of *distance* and *intentionality*, which I use to investigate the sources of knowledge of *swhidgets* in the two experiments reported in Chapter 8, showing participants claimed to use mainly the interface itself as the source of their knowledge although other sources are important as well. I also identify as perspectives (Chapter 9) some opportunities for new designs that may increase the discoverability of *swhidgets* by providing sources of knowledge in a rather unexplored area of the characterization space (small distance and intentionality).

I also investigate some aspects of *swhidgets* and signifier-less designs that help users remembering those they have discovered, taking the habit to use them, and leveraging on their current knowledge to discover other *swhidgets*. The online study presented in Chapter 8 contains questions specifically included to get preliminary results on the topics of "Overarching Interface Logic" (Section 2.6.1), "Consistency with other interactions" and transfer of knowledge (Section 2.6.1), "Action-Function Coupling" (Section 2.6.2), and "Physical metaphor" (Section 2.6.2).

2.7.4 *What are the benefits of not providing signifiers?*

I have identified three general themes in the previous section that can suggest possible benefits of not providing signifiers: "Benefits of a clean interface" (Section 2.6.1), "Consistency with other interactions" (Section 2.6.1), and "User experience of discovery" (Section 2.6.3). I discuss in Chapter 7 how these concerns fit in the model of discovery and adoption of interaction possibilities that I propose, respectively as interactions between design means that affect their efficiency, as interactions between design means and user's current knowledge, and as interactions between user's current knowledge and motivations. The online study presented in Chapter 8 contains questions specifically included to get preliminary results on these topics, and perspectives for future works on the role of user experience considerations on the design of *swhidgets* are discussed in Chapter 9.

Part I

DISCOVERING INTERACTION POSSIBILITIES

State of the arts about affordances, signifiers, and discovery models.

3 AFFORDANCES

An important aspect of this thesis is to understand the phenomenon of people managing (or not) to discover relevant interaction possibilities that are not signified to them in the environment/interface. The terms "relevant interaction possibilities" thus need some clarification and examples.

Instead of reinventing the wheel developing the concept behind these terms, I will claim that it is exactly the concept of affordances, as originally introduced by Gibson. This concept has many important but subtle aspects that Gibson covered in length in his books and papers, but that he never synthesized in a short definition of affordances. As a consequence, many works in HCI have been based on the following partial definition:

> "*The* affordances *of the environment are what it* offers *the animal, what it* provides *or* furnishes*, either for good or ill.*"
>
> [Gib79, page 127]

Since the term *affordance* is frequently used in HCI works (see [Kap14] for a review), but often in a way that is not compatible with Gibson's view [KN12], there is a risk that readers of this dissertation already have an understanding of the term that does not match the one I want to present here. It is also possible that despite the warnings raised by some authors [Nor99; Tor03], the word has been used in so many ways that it has become meaningless or unclear for some readers.

Therefore, before providing more detailed explanations of Gibson's concept of affordances in the remainder of this chapter, I will temporarily provide the following personal interpretation and wording of the concept of affordance so that readers are not confused by the possible differences between their own understanding and the one I present in this chapter:[1]

AFFORDANCE: an interaction possibility offered by an environment to a specific individual in it, that has meaning and value in an

1 Unlike the other parts of this dissertation, this chapter will rely heavily on quotes from other authors rather than simply citing their works. Indeed, the concept of affordance has many non-obvious implications and it is thus very easy to apply your own understanding on the concept to what an author wrote, and misinterpret it. I have myself many times read articles about affordances that attributed to other authors some interpretation of the concept of affordance that I could not see in the cited works. I hope that relying on actual quotes will highlight the parts of the cited works that convey the interpretation I discuss.

> ecological system, expresses the complementarity of the individual and the environment, and is independent of the individual's awareness of such a possibility, of her ability to perceive it, or of her current need for it.

Understanding the concept of affordances is crucial when investigating signifier-less designs, since the notion of *signifier* has been introduced as an alternative to the perception of affordances. Indeed, there is a common belief that signifiers are necessary only when affordances cannot be directly perceived by users. As Norman said when he introduced the concept to the HCI community:

> *"When affordances are taken advantage of, the user knows what to do just by looking: no picture, label, or instruction is required. Complex things may require explanations, but simple things should not. When simple things need pictures, labels, or instructions, the design has failed."*
>
> [Nor88, page 9]

In other words, the potential of a signifier-less design to cause usability issues depends on users' ability to directly perceive its affordances. It thus seems that to investigate signifier-less designs, we cannot avoid a careful examination of how people perceive, and more generally, get aware of affordances.

As we will see later in this chapter, there is much debate in HCI about what affordances are. The reason may be that there are many subtleties behind the concept of affordances, which cannot be conveyed easily by short presentations and definitions like the one I provided above.

I will thus discuss in detail some aspects of affordances in the next sections, starting with Gibson's work in Section 3.1 and continuing with other works from authors in the field of human-computer interaction. I will discuss aspects of affordances on which HCI authors agreed and focused, notably their organization in hierarchical structures (Section 3.2), as well as debated aspects (Section 3.3). By presenting the different views that authors have expressed about affordances, a few concerns will be highlighted that go beyond the pure theoretical analysis of affordances. The discussion will thus inform more generally on how users can get to know about the interaction possibilities provided by their devices and applications. It will notably highlight the possible roles of the various ways interaction possibilities can make sense to users (Section 3.4), of an usage culture (Section 3.5), and of explicit communication from the designer (Section 3.6).

3.1 GIBSONIAN AFFORDANCES

The term and concept of *affordance* were created by the psychologist James Jerome Gibson [Gib77; Gib79], and were later introduced (with modifications) to the HCI community by Donald Norman [Nor88]. The HCI community has embraced the term but has also often extended the concept in ways that are arguably incompatible with Gibson's original motivations and approach [KN12; Kap14]. I will thus attempt to clarify this complex and often-misunderstood concept.

3.1.1 *Origin of the concept in perception theory*

When he introduced the concept of affordances, Gibson's focus was on understanding human (and animal) visual perception, which is a topic that far extends the concerns of this thesis. I will however have to present quickly some aspects of his theory of perception that will have implications later in this dissertation, concerning how affordances can be discovered.

Gibson introduced the concept of affordances as a way to challenge the prevalent theories of perception. Notably, he challenged the cognitive view, for which perception is a sequence of information processing stages, which start with raw sensations from the sensing organs, integrating this sensory information with memory to build symbolic representations of the environment, in a process influenced by mental states and needs of the perceiving individual. Instead, Gibson believed in the *direct* perception of *affordances* through a process of *information pickup* based on the recognition of *invariants*:

DIRECT PERCEPTION AND INFORMATION PICKUP These two concepts are defined in the following quote:

> "*[...] when I assert that perception of the environment is direct, I mean that it is not mediated by* retinal *pictures,* neural *pictures, or* mental *pictures. Direct perception is the activity of getting information from the ambient array of light. I call this a process of* information pickup *that involves the exploratory activity of looking around, getting around, and looking at things. This is quite different from the supposed activity of getting information from the inputs of the optic nerves, whatever they may prove to be.*"
>
> [Gib79, page 147]

INVARIANTS How is information picked up? Influenced by his background in Gestalt perception, Gibson argues that we globally recognize *invariants*, i.e. structures of the environment that do not change as the perceiver evolves in the environment during the information pickup, causing changes in what she sees: the structures are *invariant* but what she sees *varies*.

For instance, the appearance of a static solid surface changes as the observer moves relatively to it, but in a way that is correlated to the observer's own movement: as she moves the head to her left, the surface moves to the right of her field of view and is distorted so that its parts closest to the observer move faster than the ones further away – a phenomenon called *parallax*. During this movement, the appearance of the surface varies in a way that correlates with the perceiver's movement and reveals the invariant structure of the surface.

Of course, the combination of multiple invariants is itself an invariant, and the information pickup aggregates information from all the senses into complex invariants – from isolated surfaces to the layouts of complete rooms and landscapes, up to more complex physical events and animal behaviors.

AFFORDANCES In Gibson's view, the recognition of invariants is neither the goal nor the result of perception, but only the mean by which it operates. The purpose of perception is to allow the perceiver to focus her attention on specific invariants, which are the *affordances*: what the environment offers to the individual. As Gibson says:

> "*[Objects] can be said to have properties or qualities: color, texture, composition, size, shape and features of shape, mass, elasticity, rigidity, and mobility. Orthodox psychology asserts that* we perceive these objects insofar as we discriminate their properties or qualities. *[...] The psychologists assume that objects are* composed *of their qualities. But I now suggest that what we perceive when we look at objects are their affordances, not their qualities. We can discriminate the dimensions of difference if required to do so in an experiment, but what the object affords us is what we normally pay attention to. The special combination of qualities into which an object can be analyzed is ordinarily not noticed.*"
>
> [Gib79, page 134]

3.1.2 *Affordances and ecology*

The focus on affordances as the object of perception was the novelty of Gibson's theory. As special cases of invariants, affordances must be defined as objective relationships between an individual and the environment. But affordances must also express possible interactions between the individual and the environment that bring something to the individual, independently on her needs or interest in such things:

> "*I have described the environment as the surfaces that separate substances from the medium in which the animals live. But I have also described what the environment* affords *animals, mentioning the terrain, shelters, water, fire, objects, tools, other animals, and human displays. How do we go from surfaces to affordances? And if there is information in light for the*

perception of surfaces, is there information for the perception of what they afford? Perhaps the composition and layout of surfaces constitute *what they afford. If so, to perceive them is to perceive what they afford. This is a radical hypothesis, for it implies that the "values" and "meanings" of things in the environment can be directly perceived. Moreover, it would explain the sense in which values and meanings are external to the perceiver.*

The affordances *of the environment are what it* offers *the animal, what it* provides *or* furnishes*, either for good or ill. The verb* to afford *is found in the dictionary, but the noun* affordance *is not. I have made it up. I mean by it something that refers to both the environment and the animal in a way that no existing term does. It implies the complementarity of the animal and the environment."*

[Gib79, page 127]

Gibson's theory thus rely on the possibility to define the meaning of actions objectively by grounding the actions into a broader system that contains the actor and its environment: the ecological system [Gib79, chapter 3]. As Gibson says:

"The theory of affordances is a radical departure from existing theories of value and meaning. It begins with a new definition of what value and meaning are*. The perceiving of an affordance is not a process of perceiving a value-free physical object to which meaning is somehow added in a way that no one has been able to agree upon; it is a process of perceiving a value-rich ecological object. Any substance, any surface, any layout has some affordance for benefit or injury to someone. Physics may be value-free, but ecology is not."*

[Gib79, page 140]

This is an important point of the theory that seems to me often overlooked in HCI works, and departs from cognitivist psychology views in which the meaning of actions can only be understood as higher-level cognitive constructs built over the representations created by perception and the subject's psychological state and needs. Gibson is clear on this point:

"Note that all these benefits and injuries, these safeties and dangers, these positive and negative affordances are properties of things taken with reference to an observer *but not properties of the* experiences of the observer*. They are not subjective values; they are not feelings of pleasure or pain added to neutral perceptions."*

[Gib79, page 137]

3.1.3 *Tools and learning: Affordances change for equipped actors*

The field of Human-Computer Interaction is not really concerned with the general interactions between animals and their environment, but focuses on human beings interacting with technological artifacts. Gibson's theory of affordances can thus be of interest to HCI researchers and practitioners only if it provides an account of the phenomena that happen when people interact with artifacts. In particular, if an affordance like seating provided by a chair only exists for the animal that has a body compatible with this action (excluding, for instance, children too small to reach the seat), can we say that an affordance only exists for the person who has the right tool, who is in the right condition for doing the action, or knows how to do it?

Gibson embraces the idea that tools become extensions of their user's body. This idea is quite common in HCI, notably in the field of embodied interaction, following the views of philosophers like Heidegger and Merleau-Ponty [Sva13; Bon09]. Gibson indeed said:

> "*When in use, a tool is a sort of extension of the hand, almost an attachment to it or a part of the user's own body, and thus is no longer a part of the environment of the user. But when not in use, a tool is simply a detached object of the environment, graspable and portable, to be sure, but nevertheless external to the observer. This* capacity to attach something to the body *suggests that the boundary between the animal and the environment is not fixed at the surface of the skin but can shift. More generally it suggests that the absolute duality of "objective" and "subjective" is false. When we consider the affordances of things, we escape this philosophical dichotomy.*"
>
> [Gib79, page 41]

What we learn from this quote is that, for Gibson, the "animal" is not a static entity physically defined by its body. It can thus acquire new abilities, and the environment may hold affordances only for the actor that has become equipped for doing the action, including by learning to do it:

> "*Of course, a horizontal, flat, extended surface that is* nonrigid, *a stream or lake, does not afford support for standing, or for walking and running. There is no footing, as we say. It may afford floating or swimming, but you have to be equipped for that, by nature or by learning.*"
>
> [Gib79, p. 132]

3.1.4 *Perception of affordances is learned*

If affordances can exist only for the people with the right tools and knowledge, it means that the direct perception of such affordances should itself be possible only in these conditions. Indeed, as discussed by Bærensten et Trettvik [BT02], the perception of affordances is *always* learned, and Gibson provides some accounts of how this learning can happen in his perceptual framework (including when the perception itself relies on tools, e.g., binoculars).

Kaptelinin, on the other hand, note that experimental validations of Gibson's theories "predominantly deal with processes that take place in stable life conditions (e.g., perceptual learning during infancy)" [Kap14, page 79]. As such, Gibson does not account for how people learn to perceive "disruptive" affordances, which he claims is often the case with new technologies. Kaptelinin thus concludes:

> "*explicitly taking affordances into account means that supporting users' discovery of affordances and learning how to use them should be a key designer's concern. Currently there is a lack of evidence on how exactly people learn, unlearn, and re-learn new affordances.*"
>
> [Kap14, page 80]

3.2 THE ROLE OF AFFORDANCE STRUCTURES IN HCI

Early HCI works on affordance by Vincente and Rasmussen, Gaver, and McGrenere and Ho [VR90; Gav91; MH00] simply accepted that computer systems have affordances without really questioning the possibility to apply the concept to such systems. They all highlighted, however, an aspect of affordances that Gibson did not comment much, which is the existence of structure and relations between affordances, and the role of such structures and relations in how people make sense of the world around them. They claim that the set of affordances provided by some (computerized) environment to a user is actually structured, and that understanding this structure may be beneficial to the design of user interfaces. The next subsections will describe some aspects of the structure in affordances.

3.2.1 *Means-end relations*

Vincente and Rasmussen remark that the affordances provided by a system are not independent, as some may be involved in others [VR90]. For instance, a hammer affords striking, which allows to use it for various purposes such as destroying things or forcing a sharp object into another thing (typically, a nail into a piece of wood). The specific affordance of striking provided by a hammer can thus be a *mean* to provide the more general affordances of destroying something and forcing something into another thing.

Conversely, the affordance of destroying something can be provided in many other ways than by striking it, such as throwing it or throwing something else on it. According to Vincente and Rasmussen [VR90], the relation between the affordances of striking and destroying things provided by a hammer can be seen as *means-end relations*: a first affordance can be a mean to provide a second affordance, which conversely would be an *end* (goal/reason) for using the first affordance.

Obviously, the means-end relations form more complex structures: the affordance of destroying something can itself be a mean for other affordances like getting access to some place by destroying an obstacle, obtaining the materials of the destroyed thing, playing, preventing other people to use the object, etc. A first affordance can be a mean to provide a second affordance, which in turn can be a mean to provide other affordances, etc.

Gaver invoked a similar idea in his concepts of *sequential* and *nested* affordances [Gav91]. Although he was more interested in the way affordances are perceived (or not), his examples of first grabbing the handle of a door (or scrollbar) so that it can be rotated and pulled (or slid, for the scrollbar), contain implicitly the idea that one affordance (grabbing) can be a preliminary condition for – and thus, a mean for – another affordance (pulling/sliding). In addition, in his examples, the second affordance is itself a mean toward a third affordance: a door handle is rotated to unlock the door or pulled to open the door, a scrollbar handle is slid to scroll the view.

McGrenere and Ho also acknowledge that Gibson's theory of affordances implies they can be nested [MH00], mentioning that "a word processor affords writing and editing at a high level, but it also affords clicking, scrolling, dragging and dropping". For them, "The functions that are invoke-able by the user are the affordances in software" and "affordances exist (or are nested) in a hierarchy" although "the levels of the hierarchy may or may not map to system functions".

According to Vincente and Rasmussen, creating such a means-end hierarchy for a specific computer system provides an account of how users of the system can learn its operation and find ways to achieve their goals by moving up or down the hierarchy [VR90].

3.2.2 *The problem of unclear relations between affordances*

In [Nor88], Norman introduced the concept of affordances in HCI to explain how people could successfully interact with objects that they had never seen before, yet have issues interacting with other everyday objects. Among the numerous examples he provides of taking advantage of affordances, we can cite the case of someone perceiving that a metal plate fixed on a door affords pushing the plate to open the door. Conversely, Norman provides examples of affordances that are not perceived, like a glass door that can be pushed to be opened, but the hinges of the door are hidden in such a way that people have no clue whether they should push on the left or right side of the door – both sides looking alike. He also provides examples of cases where people do not know which of multiple perceived affordances will allow them to reach their goals: a doorknob affords pushing as well as pulling, so people might be confused about which one of these two actions they should perform to open the door.

However, when it comes to opening doors, Norman only considers the affordances for pushing plates and bars, pulling handles, grabbing knobs, etc. Such actions are only means toward a higher end: opening the door. This is also an affordance, and as Gaver acknowledges in his concept of nested affordances (as we have seen above), both types of affordances are connected by a means-end relation.

What Norman's examples make clear is the general problem of a broken perceived structure of affordances: when people need to use some mean to reach some end, they need to perceive the affordances of both the mean and the end, but they also need to perceive the relationship between the two. In Norman's examples, problems arise either because some affordance is not perceived, or because a mean-affordance is not perceived to be uniquely and unambiguously connected to an end-affordance. Although these two problems have different causes and may require different fixes design-wise,[2] they have the same consequences on people confronted to them: confusion, distraction from their tasks, need to focus on the problem, and to find solutions by experimenting, searching clues, or asking people. Signifiers (and other ways to instruct users of a system how to use it) are only required when there is a risk that affordances and their means-end relations are not perceived clearly.

3.2.3 *Instrumental affordances*

The problem of unclear relations between affordances leads to the question of how people can understand the relationship between a mean-affordance and an end-affordance. Gaver's concept of nested and sequential affordances [Gav91] provide some elements of answer, by exposing the role of spatial and temporal relations between the objects that provide the affordances. This idea is more clearly established in the concept of *handling* and *effecter* affordances that Kaptelinin and Nardi introduced in their discussion of the affordances of instruments as mediators of an activity:

> "*The distinction between handling affordances and effecter affordances is obvious in case of traditional instruments, discussed by Gibson, such as knives. Such instruments usually comprise two distinct parts, responsible, respectively, for handling (e.g., a handle) and affecting objects (e.g., blade). Apparently, the distinction also applies to digital technologies. For instance, common*

2 For instance, Norman proposes to use *natural mappings* when multiple devices are controlled by similar controllers and it is thus unclear which controller affords controlling the targeted device.

> *user interface widgets, such as the scroll bar, enable two types of actions [Bea00]; operating on the widget (e.g., dragging the scroll box) and operating on the object of interest (e.g., displaying in the window a certain portion of the document). These two facets correspond, respectively, to handling and effecter affordances of a widget."*
>
> [KN12]

In Kaptelinin and Nardi's view, the *instrument* embodies the relation between mean-affordances (its handling affordances, defined as possibilities for interacting with the artifact in question) and end-affordances (its effecter affordances, defined as possibilities for employing the artifact to make an effect on an object of interest), the combination of which they call *instrumental affordances*. But the tool can also provide affordances to ease its maintenance (for instance, a battery indicator), its aggregation with other instruments (e.g., a jack plug to connect a headset), or help learning about the instrument itself (e.g., a tooltip) [KN12]. Other authors have suggested similar notions to instrumental affordances, notably Hartson's notions of *physical* and *functional* affordances [Har03] which are closely related to handling and effecter affordances respectively.

Obviously, the concept of instrument is useful to analyze human interaction with physical tools and computers, as suggested by Michel Beaudouin-Lafon [Bea00]. Yet, if we are concerned with people understanding relations between affordances, the concept of instrument is only useful if users recognize in the interface an instrument that binds these affordances together: When Beaudouin-Lafon mention an instrument for drawing rectangles with a mouse by making a drag gesture that starts at one corner of the rectangle and ends at the opposite corner, the instrument is not a visible object of the interface, and users might not think about binding the handling affordance of making a drag gesture with the effecter affordance of drawing a rectangle.

For all interaction techniques, and particularly for signifier-less designs like this rectangle drawing instrument, user's ability to discover, understand, and recognize or remember the binding between the handling affordance and the effecter affordance is an important factor to explain their (non-)use of the instrument and the usability issues it can cause. This ability is an element of answer to the research question "How can users know *swhidgets* despite their lack of signifiers?" (Section 2.7.3) and justifies considerations for aspects of the design of *swhidgets* that affect this ability, such as "Overarching Interface Logic" (Section 2.6.1), "Consistency with other interactions" (Section 2.6.1), "Action-Function Coupling" (Section 2.6.2), and "Physical metaphor" (Section 2.6.2).

3.3 DEBATED ASPECTS OF AFFORDANCES IN HCI

Does a button afford clicking? Does a scrollbar afford scrolling? Such questions are actually not easily answered. There has indeed been much debate in HCI about what should be considered as affordances in graphical user interfaces and digital devices [Nor99; BT02; Tor03; Kap14; KN12]. The debate might be technical, but it highlights some subtleties in the application of the concept of affordances to user interfaces and digital devices. To discuss these subtleties, I will start with the divergence of views about the affordances of buttons and scrollbars expressed by Gaver and Norman, coworkers and early promoters of affordances in HCI.

For Gaver [Gav91], "onscreen buttons seem to protrude from the screen; they afford pushing, but not moving or editing". Indeed, for him:

> "*[the concept of affordances] implies that the physical attributes of the thing to be acted upon are compatible with those of the actor, that information about these attributes is available in a form compatible with a perceptual system, and (implicitly) that these attributes and the action they make possible are relevant to a culture and a perceiver.*"
>
> [Gav91]

For Norman, on the other hand, "affordances reflect the possible relationships among actors and objects: they are properties of the world" [Nor99]. Hence, for him, "the computer, with its keyboard, display screen, pointing device, and selection buttons (e.g., mouse buttons) affords pointing, touching, looking, and clicking on every pixel of the screen" but a button does not provide an affordance for clicking:

> "*It is wrong to claim that the design of a graphical object on the screen 'affords clicking.' Sure, you can click on the object, but you can click anywhere. Yes, the object provides a target and it helps the user know where to click and maybe even what to expect in return, but those aren't affordances, those are conventions, and feedback, and the like.*"
>
> [Nor99]

The opposition between Gaver's and Norman's views stem from different interpretations of Gibson's concept of affordances. A careful examination of these interpretations revealed that they were essentially built upon different assumptions about three specific concepts related to affordances:

1. The *meaningfulness* of what is afforded for the actor: to say that an environment affords something, should we simply require that this thing is possible, or should we additionally require that

it has some meaning for the actor? And if so, where does this meaning come from?

2. *Social aspects* of the environment: does the distinction between physical and social aspects of the environment matter in the definition of affordances?

3. *Interpretation* of affordances: does the concept of affordances allow for interpretations of affordances or only their direct perception?

These elements will be addressed in the next three sections, respectively in relation to Gibson's notions of *complementarity*, *ecology*, and *human display*.

3.4 DEBATE 1: MEANINGFULNESS OF AFFORDANCES

3.4.1 *Norman's view: affordances as action possibilities only*

As we have seen in his quotes above, Norman associates affordances with properties of the (physical) world, with no reference to their meaningfulness for the actor. The button only exists in software and as a virtual onscreen object, but not in the physical world, so for Norman it does not afford clicking more than would any other area of the screen: only the possibility of performing the physical action is considered in Norman's definition of affordances.[3] This focus on the possibility of performing the action is made obvious in the following quote:

> *"It is possible to change the physical affordances of the screen so that the cursor appears only at spots that are defined to be "clickable." This would indeed allow a designer to add or subtract the affordance of clicking, much as many computer forms afford the addition of characters only in designated fields. This would be a real use of affordances.*
>
> *In today's screen design sometimes the cursor shape changes to indicate the desired action (e.g., the change from an arrow to hand shape in a browser), but this is a convention, not an affordance. After all, the user can still click anywhere, whatever the shape of the cursor. Now, if we locked the mouse button when the wrong cursor appeared, that would be a real affordance."*
>
> [Nor99]

3 Note that for Norman, "physical" actions can however involve visually tracking the position of an onscreen virtual object like the mouse pointer.

3.4.2 *Meaningfulness as potential*

Norman's examples of affordances in user interfaces totally ignore the question of the meaningfulness of actions: clicking anywhere in the screen is certainly a possible action, but it is also a meaningless one in the sense that it has been abstracted from its purpose, as Hartson remarked in his introduction of the concept of *functional affordances* [Har03] (see Section 3.2.3).

Such an abstraction of purpose is common in Gibson's examples of affordances, too. For instance, a flat ground affords walking, and it does not matter if you have a reason for doing so or not. This kind of abstraction of purpose implies however that the meaningfulness of an action comes from its potential to be a mean toward higher-level ends, or to be a mean for higher-level affordances that can have a clear ecological value. For instance, you can walk to reach food or a safe place to rest.

Unfortunately, Norman does not mention the existence of such higher-level affordances, even if he acknowledges that clicking on a button is a meaningful action for the user: "This is what the interface designer should care about: Does the user perceive that clicking on that object is a meaningful, useful action, with a known outcome?" [Nor99]. Yet, this meaning is only defined in relation with the user's goals and needs in a concrete situation. The action of clicking on a button, abstracted from the context of use, does not seem to be meaningful in itself for Norman.

On one hand, Norman acknowledges the purpose-abstracted affordance of clicking anywhere, but on the other hand he seems to dismiss the higher-level purpose-abstracted affordance of clicking on a button. I can only see one reason for this apparent inconsistency in Norman's concept of affordance, which is that it ignores any notion of meaningfulness to only consider the possibility of a physical action.[4]

3.4.3 *Norman's view: Meaning is conveyed by signifiers*

Norman's view implies that after having perceived an affordance of clicking anywhere on screen, the user has to interpret this meaningless affordance to deduce the possibility of a meaningful action (clicking the button to achieve something). For Norman, the meaning of the action has to be added on the perceived affordance, which is exactly the view to which Gibson objected, as we have seen in section 3.1.2. Adding meaning to a meaningless "perceived affordance" makes it a *signifier*, a concept that Norman introduces like this:

---

4 It is also possible, however, that there is a little ambiguity in Norman's wording, and that he does not reject the idea of a real (gibsonian) affordance for clicking on buttons, but only rejects it for his own concept of *perceived* affordances. This possible interpretation will be addressed in the next section.

> *"People need some way of understanding the product or service they wish to use, some sign of what it is for, what is happening, and what the alternative actions are. People search for clues, for any sign that might help them cope and understand. It is the sign that is important, anything that might signify meaningful information. Designers need to provide these clues. What people need, and what designers must provide, are signifiers. Good design requires, among other things, good communication of the purpose, structure, and operation of the device to the people who use it. That is the role of the signifier.*
>
> *The term* signifier *has had a long and illustrious career in the exotic field of semiotics, the study of signs and symbols. But just as I appropriated* affordances *to use in design in a manner somewhat different than its inventor had intended, I use* signifier *in a somewhat different way than it is used in semiotics. For me, the term* signifier *refers to any mark or sound, any perceivable indicator that communicates appropriate behavior to a person."*
> [Nor13, page 14]

As things that convey "appropriate behavior", signifiers can provide information about affordances but are not restricted to that. For instance, according to Norman, if you're running to catch a train and finally reach the station, an empty platform can be a signifier – in this case informing you that you have missed the train. Affordances are only concerned about whether you can get in the train or not, but do not depend on your will to do so. On the other hand, signifiers can provide information related to the actor's goals and expectations: as we will see in Chapter 5, they can provide feedback or feedforward, inform about the mapping between controls and the things controlled, etc.

3.4.4 *Meaningfulness from complementarity*

Unlike Norman, Gaver explicitly states that affordances should be relevant to a culture and a perceiver. For him, clicking on a button is a relevant action for the user, not only because it allows her to reach her goals, but also because the mouse and onscreen buttons are complementary elements of a system. He says for instance that "just as affordances of door handles imply the complementarity of handles and the motor system, so do the affordances of onscreen buttons imply the complementarity of buttons and mouse-driven cursors" [Gav91].

This notion of complementarity is another way to define the meaningfulness of affordances, in agreement with Gibson's statement that an affordance "refers to both the environment and the animal in a way that [...] implies the complementarity of the animal and the environment" [Gib79, page 127]. There are multiple definitions of the term “complementarity” in dictionaries, so one can ask which of these definitions is used by Gibson and Gaver. I will argue that they use the term “complementarity” to describe a situation where two elements (or more) form a system in which none of these elements can exist without the others. Animals and their environment are complementary in that sense, since an animal cannot live all by itself without an environment to support its life, and a physical world without living animals in it would not deserve to be called an “environment” (what would it be an environment of?). Moreover, because the environment also includes other species and resources, its ecological stability depends on the presence of the animal considered, or at least of a set of animals that inhabit the same ecological niche.

In that sense, it is obvious that buttons and mouse-driven cursors[5] are complementary: A button would not be usable if there was no way to point at it somehow and then trigger an event (both possibilities being provided by moving the mouse and clicking on the mouse’s physical button). Conversely, if there was no buttons and no other interface elements that make a good use of the mouse, there would be no point in having mouse-driven cursors in the system. Although there are other ways to use a mouse,[6] an onscreen button provides a reason to use the mouse and to exploit its affordance for clicking at positions on screen, and its design fits well this action. Indeed, the onscreen button has some area and a regular and convex shape, which both make it well adapted to the action of pointing at it and clicking on it,[7] in the same way than the shape and size of a doorknob is well adapted to the action of grabbing it.

As long as a user has integrated the operation of a mouse – i.e., how to manipulate it to move the cursor at the desired position, and how to press the physical buttons of the mouse – then she has integrated that the system affords clicking at a particular position. She is equipped for clicking at positions in the screen, as would Gibson say (see Section 3.1.3). But what positions should be clicked at? Onscreen buttons (as well as other graphical user interface elements) allow users to translate the action of clicking at positions into the action of clicking on things. In this sense, the button’s design and the action of clicking on it are complementary, and this complementarity is a property of the system that can be understood by – and makes sense to – users who are equipped to operate mice. There is a complementarity of the button and the mouse-equipped user, which makes clicking on the

5 Or, more generally, pointing devices, including touch screens or keyboards with arrow keys and action keys like ‘Enter’.

6 For instance, in some video games there is no pointer, and moving the mouse controls instead the camera orientation in a 3D world. Even when there is a pointer, using the mouse to draw a shape is a very distinct action than clicking on a button.

7 On the contrary, clicking a specific pixel of the screen can be very hard: a mouse and display may afford clicking “anywhere in the screen” but the affordance should rather be defined as “clicking in any sub-region of the screen” with the pixel as the smallest division of the screen, because there is no such thing as absolute pointing.

button a meaningful action, and thus allows to consider clicking on buttons to be an affordance of the system.

For similar reasons, one can talk about the affordances of dragging an onscreen handle, scrolling with a scrollbar, and other more elaborate interaction techniques, as long as the user has integrated the basic principles of these interactions.[8] This however implies that affordances may only exist for people with the required prior knowledge and skills. Gaver and Gibson embrace this view, while Norman rejects it to only consider the actions than an actor can physically perform with her body.

For Norman, an onscreen button is a signifier conveying the meaning that it can be clicked to trigger some event, and it conveys this meaning through a mix of mechanical, logical, and cultural constraints, as well as through a user mental model of how the system operates [Nor99]. These are representations that exist in the head of the user, while Gibson and Gaver see the complementarity of the user and the design of some user interface element as an objective property of the user+GUI system.

A consequence of Gibson's and Gaver's view is that affordances cannot be understood in isolation but need to be studied as elements of a system of affordances. Similarly, user's ability to exploit an affordance may depend on her understanding of a whole subset of the system of affordances, which is a concern related to the general research questions of *overarching interface logic*, *consistency with other interactions*, *physical metaphor* that convey a system of affordances, and *multiple ways to do a task*, that I have identified for *swhidgets* (Section 2.6).

3.5 DEBATE 2: SOCIAL ASPECTS OF HCI AFFORDANCES

3.5.1 *Norman's view: Affordances, conventions and symbols*

As we have seen, Norman acknowledges that clicking on a button should be a meaningful action for the user in relation to her goals and needs. However, the goal-related meaning of clicking on a button and of similar interaction techniques is for him a cultural, learned convention:

> *"Cultural constraints are conventions shared by a cultural group. The fact that the graphic on the right-hand side of a display is a "scroll bar" and that one should move the cursor to it, hold down a mouse button, and "drag" it downward in order to see objects located below the current visible set (thus causing the image itself to appear to move upwards) is a cultural, learned convention.*

8 This is not restricted to human-computer interaction techniques: it is also what allows to say that a bike affords locomotion, which obviously only holds for someone who knows how to ride a bike.

> *The choice of action is arbitrary: there is nothing inherent in the devices or design that requires the system to act this way. The word "arbitrary" does not mean that any random depiction would do equally well: the current choice is an intelligent fit to human cognition, but there are alternative methods that work equally well."*
>
> [Nor99]

He insists that "a convention is a constraint in that it prohibits some activities and encourage others" and "symbols and constraints are not affordances" [Nor99].

Strangely, however, the fact that a mouse pointer moves in the same direction than the mouse itself, or even that a mouse has to be moved to control the pointer[9] – which are also conventions – does not prevent him from saying that the mouse and display together afford pointing. And yet, someone who has never seen a mouse may not think about moving it to move the cursor (or for that matter, that it is useful to move the pointer). Indeed, the choice of action is also arbitrary. So what makes the basic operation of the mouse an affordance for Norman but not the operation of a scrollbar?

As I have discussed in sections 3.4.1 and 3.4.3, Norman distinguishes affordances and signifiers: the former are (for him) meaningless physical action possibilities, and the later are what can be interpreted so as to convey the meaning of actions. In that view, it does not matter that moving a mouse to move the pointer is a convention, because it is also a physically possible action which meaning is not considered, and thus Norman can say that it is an affordance for him. Yet, the mouse as an object is also a signifier that conveys the meaning that it can be used to move the pointer, which is a convention. The scrollbar, on the other hand, is only a signifier because dragging an onscreen handle and scrolling the view are not physical actions.

After having distinguished affordances and signifiers, Norman finally focused on signifiers and highlighted their social origin, minimizing the importance of the role of (perceived) affordances in the interaction with everyday objects. For instance, after having talked so much about the affordances of door handles in [Nor88], he later said the following:

> "*A doorknob has the perceived affordance of graspability. But knowing that it is the doorknob that is used to open and close doors is learned: it is a cultural aspect of the design that knobs, handles, and bars, when placed on doors, are intended to enable the opening and shutting of those doors. The same devices on fixed walls would have a different interpretation: they might offer support, for example, but certainly not the possibility of*

9 Which is not the only way to use a mouse, as we have seen in Footnote 6.

> *opening the wall. The interpretation of a perceived affordance is a cultural convention."*
>
> [Nor13, page 145]

If we follow Gaver's parallel between mouse-driven cursors and door handles, Norman's ideas about doorknobs would translate as follow: the affordance of clicking anywhere on screen can be interpreted differently by the user whether the click happens on a button or elsewhere, and this interpretation is a cultural convention. Hence, the onscreen button is a signifier that conveys the meaning of clickability through a cultural convention.

Unlike Gibson who acknowledges that users can understand the complementarity of mouse-driven cursor and button designs and use this complementarity to perceive the affordances of buttons, Norman only acknowledges interpretations of perceived affordances rooted in conventions. In the field of semiotics from which Norman borrowed the concept of signifiers, the signs that convey their meaning through a cultural convention are called *symbols*, and are only one of the three primary ways signs can convey meaning (with *icons* and *indexes*). Section 3.6 will discuss more in details the link between affordances, symbols, and other types of signs.

3.5.2 *Gaver's view: Culture only highlights some affordances*

Contrarily to Norman, Gaver explicitly states that affordances should be relevant to a culture and a perceiver. The fact that interactions are based on conventions or not does not seem to matter in his understanding of affordances. A button affords clicking on it because this action has a specific and relevant meaning for the user, even if the notion of onscreen buttons is a social construct. Buttons afford clicking even if this design is a convention.

Like Norman, Gaver acknowledges that culture has an impact on the perception of affordances. However, for him, it is more a question of learning to pay attention to some affordances that are then directly perceived, rather than learning to interpret affordances:

> *"The actual perception of affordances will of course be determined in part by the observer's culture, social setting, experience and intentions. Like Gibson I do not consider these factors integral to the notion, but instead consider culture, experience, and so forth as highlighting certain affordances. Distinguishing affordances and the available information about them from their actual perception allows us to consider affordances as properties that can be designed and analyzed in their own terms. Learning can be seen as a process of discriminating patterns in the world, as opposed to one of supplementing sensory information with past experience. From this perspective, my culture and experiences*

may determine the choice of examples I use here, but not the existence of the examples themselves."

[Gav91]

This view is shared with Gibson and other authors, particularly those who approach HCI from the angle of Activity Theory – notably by Bærentsen and Trettvik [BT02] who also comment on how the design of artifacts emerge from activity in a way that necessarily account for people's ability to learn perceiving their affordances, before becoming a cultural aspect of the activity.

3.5.3 *Gibson's notion of Ecology*

The origin of the divergences in Norman's and Gaver's views on affordances may be found in Gibson's notion of an *ecology*, which is the framework in which the meaningfulness of the interactions is defined. The word *ecology* can be understood as a system of animals living in their natural physical environment, as seems to do Norman – and Gibson provides many examples of affordances that fit in such an understanding. But Gibson actually uses the word *ecology* in a much broader way, which includes aspects of human life in a technological society that can seem very remote from life in natural environments. For instance, he insists that...

"*[... the man-built environment] is not a* new *environment – an artificial environment distinct from the natural environment – but the same environment modified by man. It is a mistake to separate the natural from the artificial as if they were two environments; artifacts have to be manufactured from natural substances. It is also a mistake to separate the cultural environment from the natural environment, as if there were a world of mental products distinct from the world of material products."*

[Gib79, page 130]

Gibson also gives examples of affordances of man-made tools and even mentions that a "postbox affords letter-mailing to a letter-writing human in a community with a postal system" [Gib79, page 139]. It implies that his notion of ecology is broad enough to incorporate postal systems, and that in such an ecology, "letter-mailing" is a meaningful action beyond simply inserting a flat floppy object called letter in a slot of a box called postbox. Where Norman would only acknowledge the affordance of inserting a flat floppy object in a slot and consider that this affordance has to be interpreted as the possibility to post a letter, Gibson acknowledges an affordance for posting letters that can be directly perceived.

Since Gibson rejects the distinction between natural and cultural worlds, and accepts that affordances can exist for actions that only have a culturally-defined meaning, Norman's view on affordances seems a little bit limited. It thus seems acceptable to talk about the affordances of digital devices and widgets in graphical user interfaces, as did Gaver, but also Vincente and Rasmussen, and McGrenere and Ho (see Section 3.2). However, Gibson only sets up a framework that is compatible with an HCI notion of affordance, but does not provide more practical insights for a proper use of affordances in HCI. For this reason, HCI authors like Gaver, Bærensten and Trettvik, or Kaptelinin and Nardi [Gav91; BT02; KN12] have discussed how Gibson's original concept could be precised or extended for the needs of HCI.

Although entering the details of these works is not necessary in this thesis, they provide important concepts for the study of the general research questions on *external and accidental sources of knowledge* (Section 2.6.3) and *understanding of similar interaction techniques* (Section 2.6.4). They also provide hints about the possibility to answer the question *how can users know swhidgets despite their lack of signifiers?* (Section 2.7.3) in terms of propagation of knowledge through social networks or by building an "interaction culture".

3.6 DEBATE 3: SYMBOLS AND AFFORDANCES IN HUMAN DISPLAYS

3.6.1 *Symbols involved in affordances*

We just saw with the example of the postbox that Gibson's notion of affordances includes interactions with man-made objects and designs that have a socially-constructed meaning. This view presents however some challenges when it comes to understanding what aspects of an object create a specific affordance, as it sometimes implies conventions and symbols (i.e., marks that have a conventional meaning). A naive understanding of the role of symbols and other interpreted signs – including Norman's signifiers – in the creation and perception of affordances might thus lead to nonsensical conclusions, such as an affordance being conditioned by the presence of a signifier. For instance, Bærentsen and Trettvik discuss the affordances of bills and coins for paying [BT02]: one could believe they afford paying because they display the symbols of their monetary value and proofs that they are legit.

Affordances often come from the fact that people follow some convention, but not from the fact that this convention is signaled by a specific design or symbol. For instance, as long as people accept your bills for payment, the bills afford paying, whatever they look like. A fake bill still affords paying if people do not realize it is fake. As long as people put letters in a box and the postman collects them, the box affords mail posting, even if it does not display the logo of the postal service. Conversely, a box that is in all points identical to the postboxes installed by the postal service may be a fake one designed to steal your mails, or an old postbox that is not used anymore: in both cases it does not afford mail posting despite its design.

This is not even an aspect of affordances that is specific to objects with a conventional design: pears afford eating, and they have a specific shape that is not a conventional design but the result of a process of natural (or human) selection. That people use the shape of the pear and conventional designs to perceive affordances make no doubt, but the affordance itself does not depend on an individual's ability to perceive it. And for Gibson, the shapes, symbols and conventional designs are not perceived as such to be interpreted as signifying affordances: instead, the affordance is directly perceived in a process that integrates these conventional features without focusing on them (see Section 3.1.1).

3.6.2 *From Gibson's human displays to Norman's signifiers: affording knowledge*

In many aspects, a graphical user interface fits the definition of what Gibson calls *human displays*:

> "*A* display, *to employ a useful generic term, is a surface that has been shaped or processed so as to exhibit information for more than just the surface itself. For example, a surface of clay is only clay, but it may be molded in the shape of a cow or scratched or painted with the profile of a cow or incised with the cuneiform characters that stand for a cow, and then it's more than just a surface of clay.*"
>
> [Gib79, page 42]

Gibson insists that the perception of human displays is different and more complex than the direct perception of affordances in the environment [Gib79, chapters 15 and 16], and that the affordances of human displays are two-fold: the affordances of the object itself (e.g. grabbing it) and the affordance of communicating and holding knowledge: "It can be suggested in a preliminary way, however, that images, pictures, and written-on surfaces afford a special kind of knowledge that I call *mediated* or *indirect*, knowledge at second hand." [Gib79, page 42]. This is particularly clear in his analysis of the perception of pictures that represent scenes:

> "*A picture always require two kinds of apprehension that go on at the same time, one direct and the other indirect. There is a direct perceiving of the picture surface along with an indirect awareness of virtual surface [in the scene depicted] – a perceiving, knowing, or imaging, as the case may be.*"
>
> [Gib79, page 283]

The idea that human displays can afford mediated knowledge is quite close to Norman's concept of signifiers. In that spirit, other researchers have proposed to consider similar types of affordances:

Hartson proposes to distinguish four types of affordances: *cognitive affordances*, *physical affordances*, *sensory affordances*, and *functional affordances* [Har03]. While he associates *physical affordances* with Norman's definition of (perceived) affordances, he also recognizes *cognitive affordances* that match Norman's more recent concept of signifiers. Indeed, Hartson defines cognitive affordances as *"a design feature that helps, aids, supports, facilitates, or enables thinking and/or knowing about something"*. He proposes *"a button label that helps users know what will happen if they click on it"* as a canonical example of cognitive affordance, and discusses how the design of doors and doorknobs includes both physical and cognitive affordances – in a way that is not dissimilar with the analysis of affordances and signifiers that I have done in Section 3.5.1.

Kaptelinin and Nardi propose a mediated action perspective on affordances [KN12], which also distinguishes between four types of affordances: *instrumental*, *maintenance*, *aggregation* and *learning* affordances. Although they do not provide an explicit definition of learning affordances, it seems clear from their examples that they imply something similar to Hartson's cognitive affordances – maybe only more limited in scope: learning affordances can teach how to use the artifact that provides the affordance while cognitive affordances can inform about anything. Indeed, Kaptelinin and Nardi give the examples of tooltips, help screens, and standardized icons (such as an USB icon inscribed next to an USB socket), and mention that shoelaces lack a learning affordance since most people would not figure out by themselves how to lace shoes.

Gibson, Hartson, Kaptelinin and Nardi thus seem to support the idea of what could generically be called "knowledge affordance". It can however been questioned whether this concept really fits in Gibson's framework or if it should be considered close to, but distinct from affordances. Indeed, Gibson only evokes the idea very quickly, taking precautions ("it can be suggested in a preliminary way"...). His wording does not allow to determine with absolute certitude if he means that, for instance, the word "pull" engraved on a door's metal handle affords the knowledge that the door should be pulled to be opened (which Norman would disagree with), or if he means in a general way that surfaces like metal afford being carved to communicate knowledge – the affordance would then not be to convey some knowledge, but to be turned into a signifier. The former would confuse the notions of affordances and signifiers, while the later would reduce the notion of knowledge affordance to a question of being able to produce signifiers.

In the definition of signifier-less designs in Section 6.1, I will use an intermediary interpretation, which I think is closer to Gibson's thought: the idea that some object in the environment affords to hold information if it is interpreted by the individual, although this affordance exists and can be recognized by the individual even if she is unable to actually interpret the information (e. g., a poster that is too far to be read, or with text written in a foreign language).

3.7 conclusion and recap

affordances as interaction possibilities In the introduction of this chapter I claimed that the terms "relevant interaction possibilities" needed some clarification and examples and that what I meant by that was exactly the concept of affordances as originally introduced by Gibson. Affordances are interaction possibilities for an individual in an environment, but the terms "interaction" and "environment" have to be taken in a broad sense: *"Interaction"* includes high-level interactions like nutrition or safety as well as low-level interactions like breathing, walking, or grabbing an object (Section 3.2.1). *"Environment"* also includes socially-constructed artifacts like the user interfaces of computerized devices, where interactions can range from mouse clicks and keystrokes at the lowest level, to complex application functionalities such as editing text or chatting with friends at higher levels (Section 3.2.1).

affordances are meaningful Affordances are not only interaction possibilities, but *meaningful* ones: They have a meaning that comes from the *ecological system* in which the individual is involved (Section 3.1.2), where "ecological" has to be understood in a way that includes socio-cultural systems and activities (Section 3.5.3). Their meaning also comes from their *potential to reach higher goals* (Section 3.4.2), a potential that can be expressed in the form of means-end hierarchies of affordances (Section 3.2.1). Their meaning also comes from the *complementarity* of the individual and the environment in the sense that the afforded interaction possibility can only exist in the specific system created by the environment, system that would not exist without this interaction possibility (Section 3.4.4).

dynamics of affordances Affordances can change for an individual as she *equips tools* or *learns new skills* like swimming or clicking with a mouse (Section 3.1.3). For human beings, the ability to perceive a specific affordance is not innate but *always learned* (Section 3.1.4). Once learned, perception of affordances is *direct*: it does not rely on intermediary representations or interpretations (Section 3.1.1). But affordances exist independently of an individual's ability or need to perceive them (Section 3.1.2).

AFFORDANCES HELP KNOWING WHAT TO DO Affordances play an important role in knowing how to interact with the environment: Individuals can exploit *means-end relations* between affordances to determine what to do to reach their goals (Section 3.2.1). The discovery of means-end relations between affordances can exploit spatial or temporal relations between the objects providing the affordances, notably when they can be understood as handling and effecter affordances of an *instrument* (Section 3.2.3). However, when affordances and their means-end relations cannot be perceived by an individual, *usability issues* may arise (Section 3.2.2). Such issues can to some extent be avoided by designs that embed *signifiers* (Section 3.4.3), i.e., anything that can be interpreted by individuals to inform on what they can/should do – although there is some unresolved debate about the possibility to also interpret affordances to get such information (Section 3.6).

The notion of affordance will be useful for the derivation of a precise definition of signifier-less designs in Chapter 6, as an answer to the research question *what is a signifier-less design and how swhidgets are different from other signifier-less designs?* (Section 2.7.1). It is also important to provide possible answers to the research question *how can users know swhidgets despite their lack of signifiers?* (Section 2.7.3), from the idea that users could learn to perceive affordances even when they are not explicitly signified, to the idea that affordances are meaningful and belong to systems of affordances and means-end hierarchies that can help users foreseeing unknown affordances to discover them through an exploration of the interface. This last point is the reason why I will discuss in the next chapter a classical model of interaction with the world that integrates stages of perception and interpretation.

4 THE SEVEN STAGES OF ACTION

To answer the research question *how can users know swhidgets despite their lack of signifiers?* (Section 2.7.3), we need to understand why signifiers would be necessary to discover and adopt interaction techniques in the first place. I have presented in Section 3.2.2 how Norman understands the issues caused by unclear relations between affordances and how such issues could be solved with signifiers. But this only says that signifiers can convey useful information (which is basically how they are defined by Norman, as seen in Section 3.4.3), without specifying how people will actually use such information to interact with the world. The model of the Seven Stages of Actions, also proposed by Norman, is a popular model to investigate users' informational needs as they interact with the world (e. g., [Har03; Ver+13]), and it will be used to organize the discussion around the different types of signifiers in Chapter 5. The idea that affordances should be signified if not unambiguously perceptible is born from the way of thinking human-computer interaction that also gave birth to this model: the two are closely connected. Therefore, if we want to understand why interaction techniques like *swhidgets* are designed without signifiers, it might be useful to go beyond the HCI philosophy sustaining this model – which is why, in this chapter, I will also highlight its limits.

4.1 THE SEVEN STAGES

Hutchins, Hollan, and Norman proposed the model of the Seven Stages of Action as a way to discuss the feeling of "directness" in *direct manipulation* interfaces [HHN85]. Norman later popularized the model and promoted it as design aids [Nor88; Nor13]. This model of human interaction with the world, illustrated in Figure 4.1, describes the mental and physical activities that people perform while completing task goals, and has become a classical concept of HCI to explain the way people interact with digital devices.

Norman structures the action cycle into two aspects: *executions* and *evaluations* (shown respectively on the left and right sides of Figure 4.1). The idea is that we achieve what we want to happen by executing some actions and then evaluating how these actions affected the world. Both execution and evaluation can bring their share of difficulties as we interact with the world to satisfy goals, which is why Norman talks about the *Gulf of Execution* and the *Gulf of Evaluation* to express the difficulties that can happen between the formulation of a goal and the actual interaction with the world (or interface) that one does to

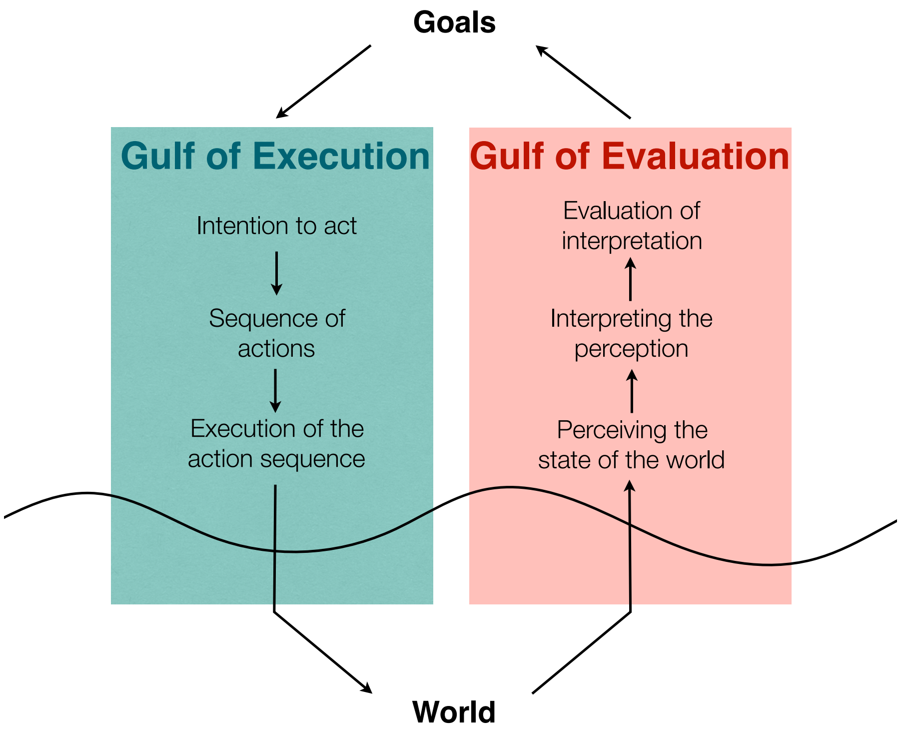


Figure 4.1: The Seven Stages of Action.

reach this goal. These two aspects of execution and evaluation are then divided into three stages each, to give six stages – which makes seven stages by adding the formulation of a goal as a stage. I will now describe these stages in the order of the cycle starting from the formulation of a goal.

GOALS According to Norman, "A *goal* is something to be achieved, often vaguely stated". Norman starts his presentation of the Seven Stages of Action in [Nor88] with the example of a precise goal: preparing a film projector for a presentation. But he quickly acknowledges that everyday behavior is usually concerned with much less precisely defined goals, and he gives examples such as "get something to eat", "get to work", "get dressed", or "watch television". The important point is that if goals specify something to be achieved, they do not specify what should be done to achieve it: goals define *what* should be achieved, not *how*. Specifying this *how* is the matter of the next two stages in the Gulf of Execution.

INTENTION For the Gulf of Execution, the first stage after the formulation of a *goal* is the stage of *Intention*, where people consider the various possible ways to reach their goals and select one as an intention to act. Norman defines an *intention* as "a specific action taken to get to the goal" and a "specific statement of what should be done" [Nor88, page 46]. He gives the example of a goal defined as "having more light so that I can read" and different ways to satisfy this goal: turning the bed light on, turning the ceiling light on, opening the curtains to let daylight enter the room, moving to a place where there is more light, etc. In the stage of Intention, people thus need to translate their goals into intentions to act, i.e., things to achieve, which expected effect in the world will satisfy their goals. Yet, as Norman says, "even intentions are not specific enough to control actions" [Nor88, page 46].

SEQUENCE OF ACTIONS After the stage of Intention in the Gulf of Execution, comes the stage of *Sequence of Actions*. In this stage, sequences of actions are specified to realize the intention to act selected in the stage of Intention. An important difference between the two stages is that they consider actions either from an external or internal point of view: intentions to act are concerned with the results of actions in the world, and thus consider actions from a point of view that is external to the actor. But in the stage of Sequence of Actions, we consider what Norman calls "internal commands", which he defines as actions "that can control my muscles" [Nor88, pages 47 and 48]. In this stage, we thus consider actions from a point of view that is internal to the actor.

We can illustrate the difference by expending on Norman's example of getting more light to read, with the case where "turning on the bed light" is the selected intention to act. Then, multiple sequences of "internal" actions can be devised to reach this result: one could reach the lamp switch with the right hand, or with the left hand, or throw an object on it from a distance, or even ask another person to do it. Note that these actions can involve other persons and objects, but are nonetheless defined in terms of what the actor has to do. While the stage of Intention was characterized by a search of effects of actions, this stage is characterized by a search for a specific sequence of actions that one can do among those that will cause the chosen effect, and the result of this stage is a plan for action.

EXECUTION STAGE There is not much to say about the stage of *Execution*[1]: once a plan for action has been devised in the actor's head, she just has to physically perform the planned actions. This closes the Gulf of Execution.

PERCEPTION The Gulf of Evaluation starts with the *Perception* of the state of the world, after this state has been affected (or not) by the execution of actions. Here, perception should not be understood as in Gibson's theory (see Section 3.1.1), but rather as in the cognitivist view

1 One subtle terminology issue with the model of the Seven Stages of Action is that the Gulfs of Execution and Evaluation are named after the last of their three stages. The reader should then take care of not mixing up the stage of *Execution* with the *Gulf of Execution*, or the stage of *Evaluation* with the *Gulf of Evaluation*.

on perception that Gibson challenged – i.e., the creation of a mental representation of the world's state.

In the case of the lamp switch example, perception could be described as building a 3D mental representation of the lamp, switch, and environment involved in the interaction, as well as the tactile perception of touching the switch, the auditory perception of the sound made by the switch when it flips, and of course, the visual perception of light coming out of the lamp.

INTERPRETATION Norman provides much less definitions and examples for the Gulf of Evaluation than he did for the Gulf of Execution, and as a consequence, the definition of its stages can be relatively unclear. In the case of the stage of *Interpretation*, he only says that "perception must be *interpreted* according to our expectations" which he also describes as "trying to make sense of it" [Nor88, pages 47 and 48]. In the case of the lamp switch example, I guess this stage would consist in interpreting the change in the geometry of the switch as a motion of the part pushed, interpreting the tactile and auditory sensations as feedback that the switch has been flipped, and interpreting the light coming out of the lamp as the result of flipping the switch.

EVALUATION STAGE In the stage of *Evaluation*, the user has perceived how the world responded to her action and interpreted this response by giving it some meaning, and now has to compare this meaning relatively to what she expected to happen: was the intended effect produced, or did something else happen instead? In the case of the lamp switch, this stage would consist in two things: validating the sequence of actions and the intention. First, acknowledging that there is light coming out (or not) of the lamp as expected if the actions that we took were correctly executed and everything worked as expected. Second, checking that there is now enough light to read, since it was the goal that started the cycle. In other words, it consists in asserting that the intention selected was indeed a valid way to satisfy the goal. This closes the Gulf of Evaluation.

STARTING A NEW CYCLE After the stage of Evaluation, a new cycle is started with a goal that depends on the result of the Evaluation stage: If the evaluation failed, the same goal may be used for a new cycle with the constraint to do things differently. Or a new goal can be formulated as understanding why the things did not go as expected. But the goal can also simply be abandoned in favor of another activity. If the evaluation succeeded, then a new cycle can start with a new goal defined by the result of the interaction (for instance, if the previous goal was to get directions to go somewhere, then the next goal could be to follow these directions). But we also often simply resume an interrupted activity, returning to a previous goal.

Typically, in the case of the lamp switch, the goal would be to resume reading if there is enough light, and to find another way to get light otherwise, or I could decide that it is not the time to read anymore and start something else, setting a new goal. Yet, in this example, reading itself might be an activity that fulfilled a higher-level goal, such as enjoying a work of fiction or searching for an information needed to accomplish another task, and these activities can influence the formulation of the next goal.

IN AND OUT OF THE CYCLE Norman makes it clear that a goal can lead to sub-goals and intentions to sub-intentions, so that one stage of the cycle can trigger a full sub-cycle, and conversely, that a full cycle can sometime be seen as just one intent or action of a higher-level cycle. Not all cycles have to be completed neither: activities can be interrupted, to be resumed after the interruption or at a later time, or even to be abandoned in the end. He also mentions that a cycle can be started from any stage, not only from the formulation of a goal: people "may respond to the events of the world (in what is called data-driven behavior) rather than to seek out plans and goals. An event in the world may trigger an interpretation and a resulting response. Actions may be executed before they are fully developed" [Nor88, page 49].

4.2 LIMITATIONS AND AMBIGUITIES OF THE MODEL

4.2.1 *Status of the model and nature of the stages*

Norman acknowledges that "the seven stages form an *approximate model*, not a complete psychological theory. In particular, the stages are certainly not discrete entities. Most behavior do not require going through all stages in sequence, and most activities will not be satisfied by single actions" [Nor88, page 48]. Hornbæk and Oulasvirta additionally note that "whereas Norman's model provides important concepts, it has limited empirical support; we cannot think of a single empirical challenge of the model" [HO17].

One aspect of the model that is not described by Norman is the exact nature of the stages: should they be understood as cognitive processing operations that happen sequentially in time? Or as cognitive layers in the brain that process information in parallel but receive information only from the previous layer and transmit information only to the next layer? Or as more abstract descriptions of human-world interactions in terms of dependencies between cognitive processes (e.g., "a sequence of actions can only be considered if a corresponding intention has been specified first")?

4.2.2 *Distinction between intentions and sequences of actions*

I notably find it hard to work with the concept of intention to act, because it is hard to separate an intention such as "switch on the lamp" from corresponding actions such as "get up, walk to the switch, and flip it". So, the line between the stage of Intention and the stage of Sequence of Action seems blurry. For instance, in [Nor88], Norman says that if I change my intention from "push the switch button on the lamp" to "ask someone passing nearby to do it for me", then I change both my intention and the sequence of action – but in [Nor13], he provides the same example and seems to say that it would only be a change in the sequence of action, not in the intention: should the intention be only "switch on the ceiling lamp", or can intentions also include constraints on how to act, such as in "asking someone else to do it for me"?

It is also hard to accept Intention and Sequence of Actions as successive steps: it is much more likely that when selecting an intention to act, we also consider the sequence of actions that would be required to fulfill this intention to act. Indeed, the amount of physical effort, the chances to miss, or the ability to perform the sequence with little thinking are all criterion used in the selection of an intention to act that depend on the precise sequence of action considered.

The separation of Intention and Sequence of Actions also seems arbitrarily in that they are both concerned with finding a way to act that satisfies a goal specified in the previous stage (the intention, in the case of Sequences of Actions). Instead of having two successive stages, the model could as well have only one stage but rely more heavily on the definition of sub-goals and sub-cycles to account for the decomposition of goals into intentions and sequences of actions. Other models such as GOMS[2] [CNM83] rely on such a decomposition of goals into sub-goals, which allows to consider intermediary levels between goals and sequences of actions (for instance, getting up, walking to the switch, and flipping the switch can be considered as three actions in a sequence, each with their sub-sequences of actions), as well as other cognitive tools such as strategies for dealing with specific problems.

This new stage resulting from the fusion of Intention and Sequence of Actions could also be defined in terms of the gibsonian affordance framework that I presented in Chapter 3 (despite Norman's disagreements with this framework): in this stage, people would identify combinations of (perceived) affordances at all levels of a means-end hierarchy, from a high-level affordance providing a mean to reach their goals (instead of an intention), down to low-level affordances for basic movements (instead of a sequence of actions). Norman's opposition between "external" and "internal" actions is compatible

2 Goals, Operators, Methods, and Selection Rules.

with the fact that the bottom part of means-end hierarchies contains de-contextualized affordances that are actions that the user can do, while the upper parts contains higher-level affordances that the user can exploit indirectly.

4.2.3 *Focus on problem-solving rather than discovery*

Norman's cycle of action suggests that we interact with the world by solving interaction problems, starting from a goal to reach (the problem) and finding a way to interact with the world (the solution) so that the goal is satisfied.[3] Hornbæk and Oulasvirta [HO17] observe that HCI views on interaction in general, and Norman's model in particular, "say little about how intentions are formed or affected by interaction". They are concerned by the fact that "HCI, via its concepts, has had an overwhelming tendency to understand interaction as one-sided – as channeling and realization of human intentions through a computer, furthermore assuming that these intentions are outside the realm of interaction itself. This has created a 'blind spot', with implications for our ability to address some important contemporary topics. For designers, this means that they are not able to anticipate how user experience changes over the course of interaction, as users discover new opportunities for action." – which is one of the general research questions that I have identified for this thesis (*user experience of discovery*, Section 2.6.3).

Having the goal as a first stage in the cycle suggests we only use a specific way to solve problems, that could be called *top-down*: by working backward from the goal to find ways of interacting with the world that satisfy this goal (first as intentions, and then as sequences of actions). But the reverse approach – *bottom-up*, starting from the action possibilities observed in the world to find which ones are likely to get us closer to the goal – is also a valid approach, as well as mixed strategies that would consider simultaneously both ends of the problem. As a consequence of the lack of precision on the exact nature of its stages, it may seem that Norman's model assumes that we solve problems only with the top-down approach.

In a critical commentary of Norman's model, Kirsh suggests that a bottom-up analysis of possible actions is a common user strategy to refine an insufficiently defined goal: "Often we explore the world in order to discover our goals. We use the possibilities and resources of our environment to help shape our thoughts and goals, to see what is possible, and we have no clear idea of what we want to do any more than we always have a clear idea of what we are going to write before

3 Norman claims that his model also covers opportunistic actions triggered by the realization of an opportunity to do them rather than by their relevance to the completion of a pre-established goal, but this claim is not well supported and is often ignored by other authors when they present the model.

we begin the process of writing." [Kir97]. Kirsh thus concludes that "the goal of an interactive interface is not merely to allow users to do what they want to do, it must also allow them to discover what they want to do" [Kir97].

One could attempt to express in Norman's model the behavior described by Kirsh, and I see two ways to do so: The first is to assume an implied goal of "finding what to do", which result will define the goal for the next iteration of the cycle (in a same way as solving a puzzle would imply to first understand the win conditions of the puzzle). The second is to assume an implied higher-level activity with its own (imprecise) goal, and to start a sub-cycle from the Perception stage, until the Evaluation stage establishes that something was relevant for the goal of the higher-level activity.

These two approaches correspond respectively to what de Bruijn and Spence call *opportunistic browsing* and *involuntary browsing*, which they distinguish from the *search browsing* that is involved in top-down problem solving [DS08]. Opportunistic browsing is a search for information and interface elements that is "intentional, but the user is unaware of any goal being pursued. The attitude underlying opportunistic browsing is 'let's see what's there'." Involuntary browsing is "unintentional, though again the user is unaware of any latent goal that might be pursued. An example is provided by a scenario in which a user's eye gaze, naturally moving rapidly between a series of fixations, serendipitously fixates on an information item that may lead to the answer to a longstanding query." Opportunistic and involuntary browsing both start from the Perception stage and go through the Gulf of Evaluation only, processing information in a way that de Bruin and Spence assume to be unconscious. While Norman does not mention opportunistic browsing behavior, he acknowledges involuntary browsing – which he associates with data-driven behavior such as encountering a friend and remembering we had a question to ask him, leading to what he calls (unfortunately for the coherence of terminology) "opportunistic actions".

Kirsh's criticism should thus not be that Norman's model does not account for behaviors of goal discovery, but rather that it can only describe this important aspect of interaction with the very generic concepts of sub-goals and sub-cycles in the generic framework of problem solving. Goal discovery behaviors rely on the types of information searches described by de Bruijn and Spence, which deserve a more explicit description in a model of user interaction with the world like the Seven Stages of Action. Involuntary and opportunistic browsing are also important aspects of *user exploration of the UI*, one of the general research questions for the study of *swhidgets* (Section 2.6.3), and are close in principle to the behaviors of accidental revelation of *swhidgets* that were described in Section 2.5.

4.2.4 *Focus on information used in short-term interaction*

Kirsh also claims that for designers, Norman's model puts the highlight on the necessity of making the action possibilities and their consequences visible, which suggests that the model may not be the most appropriate to understand signifier-less designs as it somehow assumes the necessity of signifiers. Indeed, Norman's model involves specifying intentions and sequences of actions without explaining where the knowledge of such possibilities for action comes from, and thus assumes that this information is available, either because it is visible or already known. This contrasts with Gibson's concept of information pick-up (see Section 3.1.1) that explains how we are always searching for relevant action possibilities as a primary way of interacting with the world, and how we can also pick up goals from the environment as described by Kirsh.

The focus of Norman's model on problem-solving and information used in short-term interaction also implies that it does not include the discovery of affordances in any of its stages, but only accounts for such discoveries as the result of solving "what can I do?" problems as sub-goals, or as the result of serendipitous epiphanies like "Oh! I can do that". As a consequence, the model can only describe discoveries of affordances that happen in a disruptive way, since a sub-cycle has to be started from a new (sub-)goal or from the Perception stage, which interrupts the current interaction cycle. It is then not surprising that Norman sees the problem of unclear relations between affordances (see Section 3.2.2) only as a problem and never as an opportunity to create a positive experience in the long term – which is one of the general research questions identified in this thesis (*user experience of discovery*, Section 2.6.3), as a possible direction to answer *what are the benefits of not providing signifiers?* (Section 2.7.4).

Because the model only accounts for disruptive discoveries of affordances, it is also not suited to investigate other types of discoveries, such as users deducing the existence of affordances through a reflection on what they already know about the whole system of affordances and the activity. For instance, a user who has learned recently about the possibility to apply a style to a section of text in a word processor might deduce that such a possibility only makes sense if there is also a way to define styles, and might search for a way of doing so. Where would such a deduction fit in Norman's Seven Stages of Action? Although Norman provide concepts to explain such deductions, such as his notions of mental models or logical constraints (which will be discussed in Section 5.3.8), he does not explain when the discovery of such constraints and the construction of such models happen in the cycle. This is a case of the general research question on *overarching interface logic* (Section 2.6.1) and a possible answer to the question of *how can users know swhidgets despite their lack of signifiers?* (Section 2.7.3).

Finally, the last criticism from Kirsh is that by presenting interaction as a feedback loop, it is hard to account for phenomena that arise in long-term interactions. For instance, it does not account for learning and acquisition of automatic behavior. It also ignores the dependence of user behavior on the past history of interaction and its visible consequences in the current state of the world, including voluntarily setting the world in a specific state so that it will ease later interactions (including, but not restricted to the need of resuming an interrupted activity) or help the user forming habits. Indeed, the only way the Seven Stages of Action can model these behaviors is by invoking an interaction between high-level cycles dedicated to long-term activities and lower-level cycles for the sort interactions, although the model does not explain how such interactions between cycles can happen. Understanding such phenomena is however very important to model user adoption of specific interaction techniques or strategies, and generally how users change their usage patterns and habits – especially when it involves conscious decisions to invest time in improving their knowledge of the interface through practice of interaction techniques or *exploration of the UI* (Section 2.6.3).

4.3 CONCLUSION

The Seven Stages of Action is a model of user interaction with the world and interfaces that describes the different processing stages users go through to interact. It allows to discuss users' informational needs in each stage, and to analyze how well a design satisfies these needs. As such, it promotes the use of signifiers as a general design solution and suggests that a lack of signifier is usually problematic.

Despite its usefulness to analyze the notion of signifiers, the model has limitations that makes it not well suited to the study of signifier-less designs: unclear definition of the stages, especially for the distinction between intentions and sequences of actions; focus on goal-oriented problem solving that insufficiently acknowledges the role of goal-discovery user behaviors, exploration of the interface, and accidental discovery; focus on information used in short-term interactions that insufficiently explains how users discover interaction possibilities, what are the consequences of such discoveries on user experience, and users' engagement in learning about the interface in the long term.

For these reasons, I propose an alternative model of user discovery and adoption of affordances in Chapter 7, which is based on different principles: First, an equal consideration to the three dimensions that are users' current knowledge and skills, the aspects of the interface design that can inform users, and the reasons why learning happens (including users' motivations for acting, but not restricted to their goals). Second, the organization of user actions in a cycle is replaced with considerations for interactions between the three dimensions that reflect evolution of user's knowledge and skills in time.

5 SIGNIFIERS & DESIGN MEANS

In this chapter, I consider the *means* by which elements of design in the user interface or environment can help users improve their knowledge of the interface or of the activity supported by the system, and become more skilled in using the interface. This chapter will be organized accordingly to the stages of Norman's Cycle of Action (see Section 4.1), starting from the world rather than user's goal, since the design elements concerned by this chapter belong to the world. I will thus discuss in the next sections how the design of the interface can influence the stages of perception, interpretation, and evaluation.

5.1 PERCEPTION OF DESIGN MEANS

There are two ways by which design means affect perception: one is by providing things to be perceived, as the output of the stage of perception that will serve as an input to the following stage of interpretation; the other is by affecting how well things are perceived, allowing designers to control in a limited way what users are likely to perceive and in what order.

5.1.1 *Affordances as output of perception*

As we have seen in Chapter 3, affordances are directly perceived in Gibson's (and Norman's) view. It means that perceived affordances (real or false) can be an input to the stage of interpretation and can thus be interpreted. Gibson does not discuss much the possibility to interpret affordances, but this possibility exists in his framework, as we have seen in Section 3.6.2. Norman, however, embraces the idea, as we have seen in Sections 3.4.3 and 3.5.1.

Djajadiningrat *et al.* have also suggested that designs could convey meaning through the possible actions of users [Dja+04], and in particular that the way a controller can be manipulated could inform users on the purpose of this controller as well – if not better – than labels or icons. According to them, designers have thus the opportunity to design interaction possibilities (which, in my terms, are affordances) to convey information, and should therefore not focus their design only on the problem of making these possibilities easy to perceive. We will see in Section 5.3 how affordances can convey information in practice.

5.1.2 *Sensory affordances*

In his analysis of Norman's view on affordance, Hartson's proposes the concept of *sensory affordances*, which he defines as *"a design feature that helps, aids, supports, facilitates, or enables the user in sensing something"* [Har03]. He associates sensory affordances with the stage of perception, although they have a supporting role in other stages as well. Unlike Gibson who defined affordances as relations between an environment/interface/design and its inhabitant/user, Hartson follows Norman in defining affordances as features of a design. He thus sees affordances as design means to help users improve their knowledge of the interface.

This focus on considering affordances as design means is clear in his choice of the term "sensory" over "perceptive". Indeed, he justifies this choice by the fact that *perception* can sometime also encompass higher-level cognitive processing, notably to compensate difficulties in sensing the world. For instance, if the letters of a word are hard to see because the text is too small, users may still be able to guess the word correctly from context: on a door handle, you expect the words "push" or "pull" and for some users, being able to distinguish between the shapes of "sh" and "ll" at the end of the word without being able to read the full world can be enough to "perceive" the right word.

For Hartson, designers should not expect such abilities to hold for all the users in the targeted population. Therefore, designers who want to design for all users should make sure that the word can be read easily by everybody, and hence focus on making it easy to sense rather than perceive.[1] In his own words, "In the context of signal processing and communication theory, this kind of sensing would be about whether the messages are received correctly, but not about whether they are understood" [Har03].

Obviously, signifier-less designs that hide affordances and thus make them accessible only to a subset of the targeted population do not follow this design philosophy. Yet, Hartson's point about the difference between sensing and perceiving illustrates the fact that things can be perceived or understood even if they cannot really be sensed, which raises the questions of how this phenomenon can happen in general and whether it can help understanding signifier-less designs in particular.

Hartson's sensory affordances actually cover a large range of sensing-related features, from the contrast and size of fonts that "afford" legibility, to the Gestalt laws that I will discuss in the next section [Har03]. Issues related to the quality of the sensory affordances of a design element can be understood in generic design terms as problems of its:

1 Hartson only focuses on the population of users, but his argument could be extended to the case of a single user at different times.

1. noticeability (likeliness to be sensed when not trying to attend to it),
2. detectability (likeliness to be sensed when explicitly trying to attend to it),
3. findability (likeliness of finding it when searched among other elements),

When the design element is additionally a signifier (cognitive affordance, in Hartson's terms), these generic problems can induce more specific issues like visibility, discernability (differentiating it from other similar signifiers or from non-signifying elements, including the background), recognizability, and identifiabililty. Some types of media also have specific issues, like the legibility of text and audibility of sound.

To address these issues, designers can manipulate qualities of individual design elements, like color, size, location, timing of appearance, quality of graphics/sounds/haptic signals, etc. They can also manipulate the qualities of individual elements in a composition to manipulate qualities of the whole composition, like contrast, layout complexity, or redundancy (especially, the availability of multiple presentation format can help users with sensory disabilities). They should also take into consideration how the design will affect user's attention, e.g., avoiding elements competing to attract user's attention, or designs that force users to pay attention to multiple things at the same time to complete a task (divided attention).

While designers have all these means to provide/improve sensory affordances, they also have to face trade-offs between different types of affordances. A classical example is the trade-off between adding more functional affordances in the design to allow users doing more things with the designed product, and having to integrate these functional affordances in the design without compromising other types of affordances. Hartson illustrates this trade-off with a computer-assisted design (CAD) application that displays a large number of small drawing tool icons: it causes difficulties in pointing at the icons because of their small size (poor physical affordance), but can also cause sensory affordance issues with difficulties in finding a specific icon, distinguishing similar icons, and recognizing a small icon.

Different strategies can be used to deal with such trade-offs, which can be grouped broadly in two approaches: The first one is to alleviate sensory affordance issues at the cost of reduced cognitive affordances. In the above example of drawing tool icons, it could consist in reducing visual clutter by removing some signifiers like the borders of buttons (flat design), using less-efficient signifiers like icons instead of text, or abstract but easy-to-recognize icons instead of more evocative icons, etc. Since this approach affects cognitive affordances that allow users to understand how to interact with the product, it has the consequence that it favors more experienced users who do not need these cognitive affordances anymore. The other approach is to only display a subset of the commands at a given time, relying on additional interactions to change the subset of displayed commands. Modal interfaces, menus, accordion widgets, scrollable palettes, tabs, etc. are designs that all rely on such an approach. Here, the visibility of some widgets is voluntarily decreased in order to improve the sensory affordances of the others. In Chapter 6, we will see that signifier-less designs can often be partially explained by such considerations.

5.1.3 *Gestalt Laws*

Gestalt Psychology is a descriptive theory of how humans perceive the world, based on the idea that shapes are perceived so as to form a coherent whole rather than by decomposing them into simpler shapes. In this framework, Gestalt laws are generic principles that govern the identification of such coherent wholes. Research in Gestalt Psychology has identified a few such laws, which are still commonly used today by designers as guidelines [War12], despite the Gestalt theory being now only considered as a descriptive theory.

In particular, Gestalt laws explain why some sets of objects are perceived as forming a coherent whole, by identifying existing relations between these objects such as symmetry, alignment, being inside or containing, etc. The perceived objects can become the focus of visual attention, and the relations between them can be used to move the focus of attention from one to another. A set of objects identified as a coherent whole can be attended to as a single object and interpreted in relation to its parts or in relation to the objects it is itself a part of. For instance, in Figure 5.1-a., we can perceive all four individual buttons controlling the stoves as separated objects, but we can also perceive the group of buttons as a single feature, and we are able to consider the position of each button relatively to the group.

The reason I include Gestalt laws in this review is that they can provide information to users, by creating meaning in the interface structure with no extra signifiers. They are thus useful to understand how an interface can provide "invisible" signifiers. Indeed, many of the design principles we follow arise out of gestalt laws such as closure, or common regions and good continuation in the law of Prägnanz [War12].

For instance, a frequent use in design of the law of good continuation is the alignment of widgets and fields in order to create a straightforward *completion path*: a virtual line that helps knowing in what order the fields should be filled. Another example is the use of spacing to create groups of buttons for related commands thanks to the law of proximity [Joh14, chapter 2]. The effect of this law can be seen in Figure 5.1, as we visually distinguish three groups of buttons

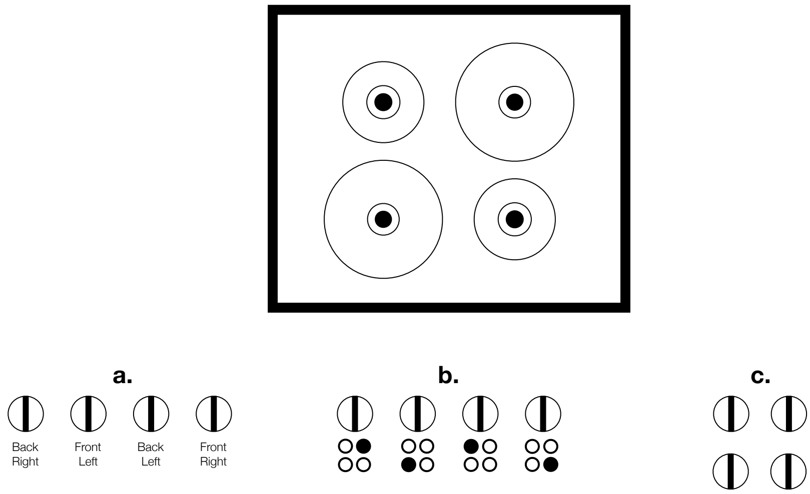


Figure 5.1: Three possible designs (bottom row: a, b, c) for the controllers of some stove (top). How can users know which controller should be used to change the power of a specific burner? In a., users have to read the textual labels underneath each controls. In b., the labels provide a visual representation that may require less cognitive processing than text. In c., no extra visual sign is used in labels, as the pattern of buttons matches the layout of the stoves and the position of the controls signify directly which stove they control. It is an example of *natural mappings* that will be discussed in Section 5.3.4.

(labeled a., b., and c.), thanks to the greater distance between buttons of different groups than between buttons of a same group. In groups a. and b., the law of proximity is also what allows to associate the button labels with the closest button. Group b. also illustrates the law of *similarity*, which is often used in conjunction with proximity. It is based on the fact that the human eye tends to build a relationship between similar elements within a design when they share visual attributes such as shapes, colors, size, etc. Here, the four disks representing the stoves are perceived as a group thanks to proximity and similarity, but there is additionally two subgroups perceived where similarity breaks: one subgroup for the three white disks, and one subgroup for the black disk that represents the stove controlled by the nearby button. Group c. is an example of *natural mapping*, a design solution that will be discussed in Section 5.3.4 and cannot be explained by perception alone although perception is obviously involved in the ability to understand the similarity of patterns between the stoves and buttons.

The description of perception in Gestalt Theory also provides some mechanisms to interpret the forms and structures perceived. We just mentioned how the law of proximity allowed to group the stove buttons with their labels in Figure 5.1: this proximity relation between the buttons and labels can be interpreted as "the label informs about

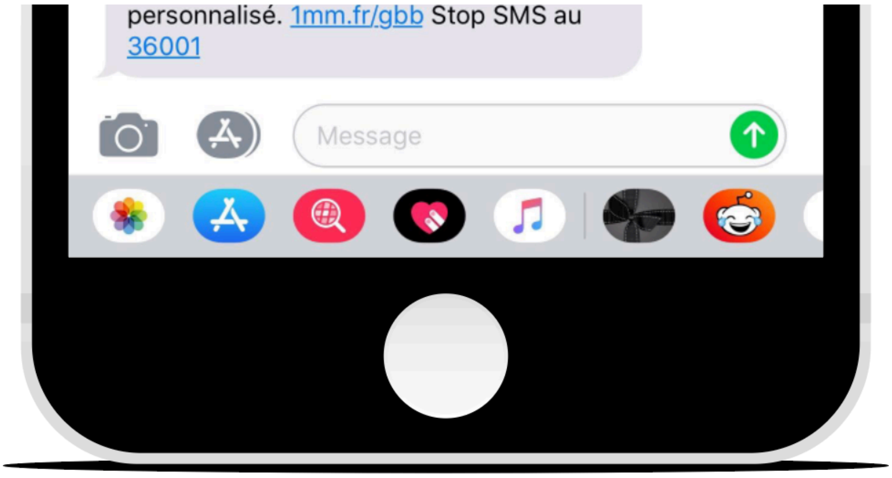


Figure 5.2: Using the Gestalt law of closure as a source of information on possible interactions: In the *Message* application (iOS 12.4), a row of icons at the bottom of the screen allows to share directly in the conversation some content provided by the corresponding applications. The spacing and size of icons is such that when there are too many icons to be all shown, one icon (white, here) is made only partially visible on the right edge of the screen. Users can then understand that this bar can be scrolled to the left to reveal more application icons. This is important because users may not expect this interaction possibility, as this bar would otherwise look like the icon bars that iOS uses at the bottom of the screen in most applications, which are not scrollable.

the nearby button". Similarly, the alignment perceived in a completion path design is interpreted as an ordering of the input fields. The interpretation can also be made in terms of action possibilities. An example is the case of an object partially out of a scrollable view: the Gestalt law of closure allows to recognize that there is an object even if it is only partially visible, and provides the information that this object is partially outside the view boundary. Users can then use this information to infer that the view can be scrolled. This strategy is used in some designs to replace dedicated signifiers, as shown in Figure 5.2.

5.2 Definition of signifiers in Peirce's semiotics

When he introduces the concept of signifiers, Norman mentions their origin in *semiotics*:

> "*The term* signifier *has a long and illustrious career in the exotic field of* semiotics, *the study of sign and symbols.*"
>
> [Nor13, page 14]

The theory of *semiotics* by Charles Sanders Peirce has been invented to describe how people can interpret signifiers, and is to my knowledge one of the very few models to account for what happens in the stage of Interpretation without entering into cognitive theories. Norman is right about the theory's *long and illustrious career* (it dates from the late 19th century), and about its *exotic-ness* for HCI practitioners and researchers: it is complex, filled with terminology that one can hardly memorize, it aims at being an universal description of very distinct phenomena, and it is rooted in disciplines like philosophy, logic, and epistemology rather than experimental science. And yet, the theory is surprisingly pertinent to better define what a signifier actually is and how users can interpret them.

HCI researchers like Gaver [Gav89] have also noticed this pertinence early. Other authors advocated semiotics-based approaches to HCI, although they focused on semiotics concerns that are more in the spirit of the works of Ferdinand de Saussure, co-founder of the field of semiotics. Unlike Peirce who addressed *semiosis*, the processing of signs by an individual, Saussure focused more on the study of systems of signs as cultural phenomena. HCI works in this tradition have thus focused on using semiotics to understand digital media rather than particular interfaces [ONe08; And01], or to analyze a design in terms of an act of communication from a human product designer to a user [deS05] or between a human user and the computer [And91; Ben94]. Although semiotics have been used as the core concept in methods for analyzing the interactions with a system or application [OBT02], I am not aware of any work that uses Peirce's semiosis concepts in order to classify interaction techniques and design solutions beside Gaver's.

5.2.1 *Components of a Sign: Signifier, Object, and Interpretant*

According to Peirce, a sign is, broadly speaking, something that allows an interpretation. Signifier, object, and interpretant are the only components of a sign for Peirce.[2] They are revealed by applying respectively his metaphysical concepts of Firstness, Secondness, and Thirdness to the sign as answers to the question "what is a sign?":

THE SIGNIFIER is something that can be interpreted: an image, a sound, an action, a word or sentence, a volatile object like smoke, a country flag, a layout, an event, a pattern, a law, etc. – but considered as a thing in itself, before any interpretation. The signifier exists by itself outside the interpretation but provides a ground for it, i.e., the abstract feeling that can trigger the interpretation.

THE OBJECT is what will be used in the interpretation, what the signifier stands for or is about: what is shown in the image, the meaning of the word, the fire that emits the smoke, the country of the flag, the rule that governs the pattern, etc. The object determines

2 Peirce has used many different terms for these concepts, and I chose these three terms for their compatibility with Norman's usage of the term *signifier*.

the signifier, not the other way around. The object is only accessed through its relation to the signifier, and only the qualities of the object that are involved in this relation are relevant, as they determine the signifier and its ground.

THE INTERPRETANT can be broadly understood as the person (or computer!) that do the interpretation, or as the process of the interpretation, but is more exactly defined as the effect that the interpretation has on the person doing it. In particular, it is the understanding that she has of the signifier-object relation. But it may also be the emotions that she feels because of the interpretation, or other types of reactions. The interpretant puts the signifier in relation with the object through its mediation.

5.2.2 *Combination of Signs*

In Peirce's view, signs can be combined in potentially infinite chains of interpretations. Indeed, Peirce wanted to create a framework that could describe cognition in a general way, including for instance logic and inquiry. He saw signs as the building bricks of thought, and discussed thought-signs (signs that only exist as thoughts, by opposition to the elements of signs embedded in physical objects).

For Peirce, an interpretant is also a thought-sign and can thus become a new signifier or a new object in a new interpretation. For instance, if I see a line drawn so as to form a star on a tool button in a drawing application, I can first interpret this line (signifier) as representing a star (object) and thus, being a star icon (interpretant). But I can then further interpret this star icon (previous interpretant as new signifier) as representing a drawing tool (new object), and interpret it as a tool to draw stars that will be activated by clicking the button (new interpretant).

5.2.3 *Nature of the signifier-object relation*

The following categories of signs are the most often discussed in HCI and graphic design. They come from an interest in the relation between the signifier and its object. There can be three different ways to be interested in this relation:

ICON The relation between a signifier and the object of the sign exists by itself outside of any interpretation. It can be perceived directly, and thus necessarily consists in a similarity between qualities of the signifier and object, which "are alike". Since the object determines the signifier, it can be said that the signifier signifies the object by virtue of its resemblance to it.

In user interfaces, graphical icons are often semiotic icons too, as for when the concept of a "new document" is represented by a blank sheet of paper or "printing" is represented by the image of a printer. A document represented in a WYSIWYG editor is a semiotic icon, as is a file icon that is a snapshot of the document. Figure 5.1 shows how iconic signifiers can be used, first to improve on the textual labels in (a) by using iconic labels presenting a similitude between the stove pattern and the icons in (b), and then how this design can be improved in (c) to eliminate the need for button labels by configuring the buttons themselves in a pattern similar to the stoves – effectively creating a type of signifier that Norman calls a *natural mapping* [Nor88].

In the category of icons, Pierce has distinguished three subcategories: *images* that rely on a perceptual resemblance (like portraits), *diagrams* that rely on a similarity between structures so that the relations between parts of the signifier match relations between parts of the object (as for the stoves in Figure 5.1), and *metaphors* for which the similarity is arbitrarily imposed as a rule but allows to connect parts of the signifier to corresponding parts of the object (e.g., Shakespeare's "All the world's a stage, And all the men and women merely players").

index The relation between a signifier and the object of the sign exists in virtue of an actual, factual, natural relation between the two, such as a cause-consequence relation (the smoke is an index of fire because it is produced by the fire) or a spatial relation (the finger pointing at the Moon is an index for the Moon, to which it is related by the spatial alignment of the two), or a mix thereof (the cat stands next to its empty food plate because she is hungry).

In user interfaces, it is very common to associate labels to the widgets they describe using a proximity relation, as we will discuss in Section 5.3.2. Another common type of indexes is the use of carets, cursors and other pointers to indicate an active element or position in a set, which is signified by the proximity relation between the cursor and the active element. Other examples will be provided in Section 5.3.

symbol The relation between a signifier and the object of the sign exists by convention, habit, law, or any other entity that creates the relation through its mediation.

This is a very common type of signs in user interfaces: many graphical icons have a meaning only by convention, such as the triangle pointing to the right which means "play a media" and the two vertical bars that mean "pause the media", the × cross that means "close the window", the concentric arcs that mean "Wi-Fi", etc. Other signs may have an historical origin as icons but are now used only by convention, such as the diskette symbol that still means "save the file" despite the fact that such diskettes have been deprecated decades ago. Beside graphical icons, most instances of text are also symbols in that the shape of a word is usually unrelated to its meaning (except in onomatopoeia such as "cock-a-doodle-do" or "splash"). Symbols can also be seen in some hotkeys, since a key combination like Ctrl-Z is arbitrary and not related to the command "undo" that it triggers and can thus be understood as its semiotic object.[3]

5.3 interpretation of design means

HCI authors have proposed concepts for the design and analysis of signifiers, which can be discussed in a Peircean framework, notably with the distinction between icons, indexes and symbols.

5.3.1 *Perceived affordances*

A perceived affordance is a signifier in Norman's views, and it can also be understood as a signifier in Peirce's semiotics. Indeed, the action possibility (the Gibsonian affordance) that has been perceived is a signifier in a sign which semiotic object is the set of physical objects involved in the action, and the interpretation would be an understanding that such an action can be performed with these physical objects.

The sign is an index because the affordance is a consequence of the qualities of these objects, notably their shapes and dimensions, and comes neither from a resemblance or an arbitrary relation. This classification highlights the absence of cultural, arbitrary or learned elements in the interpretation, which is exactly why the concept of perceived affordance was praised by Norman.

5.3.2 *Labels*

The most common type of signifiers beside perceived affordances are labels, understood in a broad way as informative marks displayed on or next to the element concerned by the information, whether it is the text "push" on a door or the image of a trashcan on an on-screen button. Such labels require a combination of signs to be interpreted:

First, the label itself has to be interpreted, independently from its placement near another object. The label as signifier is not a sensation or object, but is a concept (the word "push", a trashcan), and it is interpreted as a quality of something (pushability, deletability) that is an abstract affordance (it will be made concrete by applying it to the door or button). The way the signifier is linked to the semiotic

3 Other hotkeys such as Ctrl-S for "Save" can been understood as icons because they share a property with the name of the command in its initial letter, but even then this is more a mnemonic and historical relation than a true one and the hotkey has still an arbitrary relation to its command.

object, here, does not matter: the word push is a symbol (conventional) and the trashcan is an icon (looking like a real trashcan) used as a metaphor.

Second, the label has to be interpreted as informing about an interaction possibility of the door or button as an index: The label is spatially related to the object of the sign (door or button) that determines its position, and it is thus an index.

5.3.3 *Interface Metaphors*

Metaphors – also sometime called analogies – allow users to interpret elements of the interface by mapping them onto objects and events of a real-world environment that is supposedly more familiar. The most famous example is the Desktop metaphor that draws an analogy between computer concepts such as file systems, databases, etc. and a desk, with documents that can be opened on the desktop or stored into drawers, etc. While the suitability of metaphors as tools to learn the interface has been criticized [HM82], they are still used today as core elements of the design of new products [HW10].

Peirce's definition of metaphors as a particular type of icon is compatible with the usage of the term in HCI [BBN04], and Gaver, based on Peirce's works [Gav91; Gav89] additionally discusses how interface icons can be created in interface metaphors either as semiotic symbols (e.g., an arbitrary sound associated with an action), as metaphors themselves (e.g., a sound of decreasing pitch and loudness or a graphical icon fading into transparency to represent the deletion of a document), or as icons that use the interface metaphor indexically (looking or sounding like an object or event implied by the interface metaphor, such as the sound and visual animation of paper scrambled to represent the deletion of a document).

5.3.4 *Natural Mappings*

The notion of iconic signifiers can be found in Norman's *natural mappings* [Nor88], a design mean aiming at making it easier for users to identify the relationship between a set of controls and what they affect in the world. To do so, natural mappings consist in using a structural analogy between the pattern of controllers and the pattern of things controlled, or between the actions to perform on these controllers and the effects of these actions. It is believed that such a design leads to an immediate understanding of the mapping, since the analogy can be understood simply by looking at the controllers and controlled elements and does not require any external knowledge.

For instance, the controls for adjusting a car seat (Figure 5.3) look exactly like the seat and users can move the controls corresponding to the different parts of the seat directly in the direction they want the

Figure 5.3: Example of a natural mapping for the controls allowing to adjust a car seat.

seat part to be adjusted. Another example is the display c. of stove controls in Figure 5.1, which leads to immediate understanding of which control affects each stove. Mapping problems are abundant, for Norman, it is one of the fundamental causes of difficulties in the interaction with everyday things. In Peircean semiotics, natural mappings would be icons, and more precisely diagrams.

An idea related to natural mappings is the concept proposed by Djajadiningrat *et al.* [Dja+04], consisting in designing products so that users can not only perceive an affordance (possible action) but also interpret it as an information about the purpose of this action. For instance, they propose the design of a digital camera in which the picture taken, visualized on the camera display, can be transferred on a memory card by physically pushing the display so that it connects with the visible memory card. Although there is no mapping in this case, the action of connecting the display showing the picture and the memory card is a metaphor for the transfer of the image to the card.

5.3.5 *Interaction Frogger*

Wensveen, Djajadiningrat, and Overbeeke proposed the *Interaction Frogger*, "a design framework to analyze person-product interaction" [WDO04]. In this framework, they propose to distinguish between inherent, augmented and functional information, conveyed either as feedback of feedforward.

FUNCTIONAL INFORMATION *Functional feedback* is defined as "the information generated by the system when performing its function, e.g., sound, light or motion" and an example would be how the TV screen lights up when the user presses the power button.

Functional feedforward is information "about the more general purpose of a product and its functional features", and an example would be how "the speakers and a screen on a black box inform the user about the audiovisual functionalities of the product".

AUGMENTED INFORMATION *Augmented feedback* "refers to information not coming from the action itself (which is inherent feedback), but from an additional source", and an example with the TV would be the red light emitted by a diode next to the power button when it is pressed.

Augmented feedforward is "information from an additional source about the action possibilities, or the purpose of the action possibilities", and examples would be "on-screen messages indicating what to do" and "lexical or graphical labels communicating the purpose of the action possibility", e.g., on a button or next to it.

INHERENT INFORMATION *Inherent feedback* "is the information that is returned from acting on the action possibilities", and an example would be the displacement, feel, and sound of a button when pushed.

Inherent feedforward "is the information that communicates what kind of action is possible (pushing, rotating, sliding) and how this action can be carried out (the amount of force that is possible, which parts of the body, etc.)" and the authors notice it can be understood "as a limited interpretation of the concept of affordance", citing Gibson although their definition of inherent feedforward as an information is closer to Norman's understanding of perceived affordances (see Chapter 3).

RELATION TO PEIRCEAN SEMIOTICS Since the six types of information introduced in this framework are conveyed by elements of a product's design (and thus, by design means), these elements can be understood as signifiers, which semiotic objects are the interaction possibilities they concern (i.e., in the terms of this thesis: affordances). Augmented feedback and feedforward are then distinct from the other types of information in that they are symbolic, but the others are indexes. Indeed, when feedback and feedforward is "augmented", it is because they use as a signifier an element that is not involved in the interaction, and is symbolic like a label (see Section 5.3.2). On the other hand, inherent and functional feedback and feedforward are indexes in that they are directly caused by the actions and functions on which they inform. The difference between inherent and functional information then comes from the fact that they are concerned with (Gibsonian) affordances at different extremities of the mean-ends hierarchy (see Section 3.2.1).

5.3.6 *Natural Signals*

In his book The Design of Future Things, Norman criticizes the designs of appliances that use artificial and unpleasant beeps as feedback of operation [Nor09]. What he criticizes is not only the annoying character of these sounds, but also their artificiality: by being purely functional (having only qualities designed to ensure they are perceived and noticed), they miss the opportunity to positively affect users emotionally. He discusses the benefits of using in design *natural signals*, which take inspiration from signals of the natural world (notably sounds). There are two ways to understand this: one is to create signals that are similar to signals occurring in nature, such as bird chirps; the other is to use signals that are naturally emitted by the appliances.

The former type of signals relies on similarity between the appliances' signals and those perceived in a natural environment, but would not be iconic signifiers in that their object is not the sound they reproduce: a microwave oven that plays a bird sound instead of beeping when your meal is cooked informs about the cooking, not about a bird singing. The relation between the object and the signifier would still be arbitrary and the sign would thus be a symbol. However, the interpretant in Peirce's theory can also consist in an emotional response and the object of a sign can also be the feeling of peace that one can experience as she listens to the birds in a country garden: in that case, this type of natural signal could be understood as being an iconic sign.

The second approach to natural signifiers consists in having appliances communicate their state through signals that are naturally emitted as a consequence of their operation. Norman gives the example of the sound emitted by a vacuum cleaner, which may not be enjoyable but clearly informs on the operation of the device. Moreover, users can notice subtle changes in the sound and interpret them as signs of specific events, such as when a small object get stuck in the vacuum cleaner, producing a higher pitched sound, or when the dirt bag gets full and the aspiration seems less powerful. As signs, these signals are indexes since they are event signifiers which objects are the state of the device, and the relation between the objects and the signifiers is a cause-consequence relation: the state of the device causes the signifier directly.

5.3.7 *Feedback and feedforward*

When design means are considered as signifiers, it can also be interesting to categorize the possible semiotic objects of the corresponding signs. As we have seen in the examples provided in the preceding sections, the semiotic object can be an interaction, and a very basic distinction is then between signs providing feedback and feedforward: for signs that provide feedback, the semiotic object is an actual action done or being done by the user, while for feedforward it is a possibility of (future) action [DOW02; Ver+13].

This distinction is based on the timing of events in an action model relying on a loop, such as Norman's Seven Stages of Action (see Chapter 4), but a more detailed analysis can be done (especially for feedforward). For instance, in the framework *Interaction Frogger* (see Section 5.3.5), the distinction between functional and inherent information can be understood as applying to different extremities of the product's means-end hierarchy of affordances: different types of action possibilities. A similar distinction was done in [DOW02] to introduce the concept of feedforward as distinct from perceived affordances, where the former is understood to concern the purpose of an interaction and the latter to only concern the possibility of an interaction.

A related concept is the distinction between signs which object is the purpose of an interaction and those which object is the consequence of an interaction. For instance, a modern design (especially on smartphones, tablets, and the web) is to propose a generic menu accessed by taping or clicking on an icon displaying three horizontal lines stacked vertically. This icon can be understood as a signifier representing the menu as a semiotic icon: the three horizontal lines being like the entries of the menu, which also are elongated horizontal rectangles stacked vertically. In other word, this icon says nothing more than "click here to open a menu": it informs on the consequence of the action, but not on its purpose, as it says nothing about the kind of commands that can be found in this menu.

Another example of signifier conveying the consequences of an interaction rather than its purpose is provided by widgets that are disabled ("grayed") when activating them would lead to an error message. Fortunettes [Cop+19] is a technique to convey the consequences of interactions with some widgets that change their state or the state of other widgets (for instance, a disabled submit button that becomes enabled when the data provided in a form become valid), by changing the visual appearance of these widgets.

The amount of information provided about the consequences of an action can actually change the way the interaction is done. Guillon *et al.* investigate this effect in target expansion techniques [GLN15], for which they distinguish between implicit and explicit feedforward: an explicit feedforward shows the boundary of the cells where clicking has different consequences (different targets are selected) but an implicit feedforward when the user only gets the information that she entered a different cell.

Finally, signifiers can inform about the final state of the system after an interaction, in a way that is both the consequence and the purpose of the action. For instance, in his analysis of cognitive affordances (which are like Norman's signifiers), Hartson discusses the design of a switch in his FM player to change the mono/stereo output setting [Har03]. He argues that a good design would have used only the labels "mono" and "stereo" on each side of the switch so that users understand that moving the switch towards one label sets the system in the corresponding state. Such labels inform on the purpose of the switch (and each of its two positions), on the consequence of interacting with the switch, as well as on how to interact with it in order to get the desired setting. On the contrary, the actual design of his FM players shows the same labels but inverses their positions and adds arrows depicting the direction in witch the switch must be moved in order to get to the labeled state. Harston concludes that the labels and the arrows provide conflicting cognitive affordances and that the design is confusing because of that.

It raises the question of whether it is always better to use signifiers that are about the intended system state or if it is sometime better to use signifiers that are about the gesture that one should perform to reach this state (as communicated by the arrows in Hartson's FM player). Interaction techniques like Octopocus [BM08] provide feed-forward for gestures that concerns both the shape of the gesture to perform and the command it triggers (which is both the purpose and the consequence of the gesture). In this case, the information is made non-ambiguous by the fact that the labels describing the commands are placed at the end of the lines that represent the gestures, which all start from the same position, and disappear as the user performs a gesture that is not compatible with them.

5.3.8 *Constraints*

Constraints are also proposed by Norman in [Nor88]. They are properties of objects from which users can deduce possible operations. While affordances "suggest" possible actions in Norman's views, constraints "limit" the number of alternatives to consider. Properties of the objects perceived must be interpreted as constraints in the stage of Interpretation, but information about constraints can be used at two places in Norman's cycle of action: in the (future) stages of Intention and Sequence of Actions to help users determine what to do, and in the stage of Evaluation that follows to allow people to understand why an attempted action failed, as not satisfying a constraint. Norman discusses four types of constraints: physical, semantic, cultural and logical.

Physical constraints are action possibilities that are limited by a mechanical system. For instance, the closing line between the screen and the keyboard of a laptop is a perceived affordance that informs the laptop is made of two parts that can be moved relatively to each other, the presence of a hinge constrains the direction in which users can open the laptop. Another example, inspired by Norman's analysis of a Lego toy assembly [Nor88, page 83-84], would be how the pieces of a jigsaw puzzle cannot be assembled together when their shapes do not match. Physical constraints exist also in signifier-less designs, for instance an *item swhidget* like the ones in the Mail application can only be triggered laterally, as a vertical swipe would conflict with scrolling.

Semantic constraints "rely upon the meaning of the situation to control the set of possible actions". The example to open the laptop in the correct direction is also a semantic constraint in that the screen fits above the keyboard in the user's mind, and thus, opening the laptop ends with a system in a usable configuration (which is the goal of the user for opening the laptop). In the jigsaw puzzle example, semantic constraints would be that the image formed by assembling the pieces should make sense as a picture. Semantic constraints exist also in signifier-less designs, for instance *item swhidgets* can only exist on items that would benefit from them as they can be the targets of specific commands.

Cultural constraints rely upon accepted cultural conventions to limit the set of acceptable behaviors. They are often meant to be read, even if they do not affect the physical or semantic operation, such as car lights: red lights placed in the rear signify the action "stop", while white lights is the standard for headlights which go in the front.

Logical constraints come from the possibility to deduce what should be done from assumed rules. For instance, when there are two switches and two lights and we already know what light is controlled by the first button, we can deduce that the other switch will control the other light (assuming that all lights need a controlling switch). Similarly, Norman give the example of a scrollview with an element half visible on the view border (as in Figure 5.2), from which we can deduce that the view must be scrollable [Noro8].

5.4 evaluation of design means

The stage of Evaluation ends the Gulf of Evaluation that is concerned with understanding the world. As a conclusion of this chapter, we can see how the design means have been interpreted so as to create expectations that can be verified in this stage.

As we have seen in the previous section, design means can provide feedback and feedforward. Users can confront the information provided as feedback to the expectations created by feedforward before acting. The considerations of different types of feedback and feedforward that I presented in Section 5.3.7 allow to better understand these expectations, how they can be verified in the Evaluation stage, and how users can react in case the expectations are not met. I chose to present the information provided by feedback and feedforward together in the stage of Interpretation because it is were this information is obtained, but it can also be considered that this information is used in other stages, or is required for other stages, as do [Har03] and [Ver+13].

The Evaluation stage could also be concerned with evaluating the practicality of a Gibsonian affordance: instead of checking the interpreted world against predefined goals, users can as well check these interpretations of design means against *possible* goals, as suggested by Hirsh (see Chapter 4.2.3). This idea can be found in McGrenere & Ho's idea to treat affordances as a continuous number rather than as possible / not possible, and the observation that people can evaluate directly the "climbability" of a staircase [MH00]. Indeed, if we consider an ill-defined and abstract "find what to do next" goal for actions that start with the environment, we can consider that the evaluation stage is about determining if observed interaction possibilities are things worth doing next.

An important concern for the evaluation of design means is the notion of "level of detail" [BM08; Ver+13], which is related to how well the signifiers provide information required in the evaluation stage. This notion of level of detail focuses on the amount of relevant information provided, how often it is provided, etc. However, the notion of how clearly the information is provided is harder to define and equally important. I provided hints at how the semiotic theory of Peirce can be used to understand how signs are interpreted in human-computer interaction, in hope that it can provide insight about this question, notably by exposing the knowledge required to perform some interpretations and how some designs can avoid to rely on such knowledge.

Part II

SIGNIFIER-LESS DESIGNS

6 SIGNIFIER-LESS DESIGNS

In this chapter, I finally introduce *signifier-less designs* more formally. In Section 6.1, I first propose a simple definition of signifier-less designs, then expose the difficulties of establishing such a definition and discuss its limitations. I propose to solve these issues by defining the notion of *directly signified* affordances, building upon some subtle notions introduced in Chapter 3: instrumental affordances and affordances for mediating knowledge. In Section 6.2, I observe that signifier-less designs have been provided by the user interfaces of most systems in the past and present, review a selection of signifier-less designs and discuss how the lack of signifiers relates to the principles on which these designs are built. I will finally present in Section 6.3 three different sub-types of signifier-less designs based on which affordance is not signified, and discuss possible user reactions when they encounter them.

6.1 DEFINING SIGNIFIER-LESS DESIGNS

As Norman said in the quote that introduced this dissertation, from an industrial design perspective, some controls are hidden away for aesthetic reasons [Nor88]. For example, many electronic devices like desktop computer displays have a power switch at their back, avoiding accidental press on the button, but also hiding the button to users. Switching power is a function that only needs to be done once (and another one when shutting down), but it is crucial to access all other functions of these devices. From this example (and others that will be provided in section 6.2), we can propose the following broad definition:

SIGNIFIER-LESS DESIGNS are interface designs in which some interaction possibilities are not directly signified to the user when she needs it.

6.1.1 *Edge cases and limitations of the definition*

The expression "signifier-less design", strictly speaking, has a natural meaning that only entails "interaction possibilities for which there is no signifier". My definition adds two subtle requirements on top of this natural meaning, to exclude the two following cases: 1) the existence of signifiers that are not present at the moment users need them; and 2), interaction possibilities that are signified indirectly – notably, on user's request.

The reason why these additions are required is that without them, the signifier-less aspect of some designs might be subjective, and the natural meaning of the expression is not precise enough to decide if these designs should be named "signifier-less" or not. Take for instance a door that has to be pulled to be opened, but has a classical handle that affords both pulling and pushing, and no visible hinge that could inform the user that the door has to be pulled. Such a door should certainly be called a "signifier-less design". On the other hand, if the word "PULL" is written in big letters on the door, it should certainly not be called a signifier-less design. But what about the following situations?

1. The word "PULL" is not painted on the door but printed on a poster, and the poster is pinned...
   a) ...on the door itself.
   b) ...on the wall next to the door.
   c) ...on the wall, at the other end of the corridor, ten meters from the door.
2. The word "PULL" is absurdly painted on the other side of the door, so that you have to open the door before seeing it.
3. The word "PULL" is painted on the door, but you do not need to open it.
4. The word "PULL" is painted on the door, but the door actually does not open.
5. There is no word "PULL", but...
   a) ...every person who needs to open this door has been communicated the information that it should be pulled before even facing the door.
   b) ...there is always someone nearby to tell you that the door has to be pulled before you reach it.
   c) ...there is always someone nearby to tell you that the door has to be pulled if they see you hesitate or try to push it.

In cases 1a) to 1c), it could be argued that the signifier (the poster) is not part of the original door design, which should thus be called "signifier-less". But there is no reason to restrict the definition to the original design of the objects/interfaces rather than applying it to the current design of objects and environments constituting the context of use. Indeed, case 2 shows that applying the definition solely to objects rather than to the context of use could lead to nonsensical use of the expression. The terms "when the user needs it" in my definition allow to precise this context of use. However, the distinction between the cases 1a) to 1c) shows that there is no way to define exactly at what distance from the door the poster would still be visible "when the user needs it". This issue will be addressed in section 6.1.2 below, in relation to the "directness" of the signifiers.

In addition, understanding the relation between the poster and the door gets harder as the distance between the two increases, and we now have a problem of defining how well something should be signified to not be considered as a signifier-less design. I will simply not enter into these considerations: signifier-less design are not about insufficiently signified interaction possibilities – which users could simply and rightfully dismiss as bad design – but about a total lack of signifier, which can be a voluntary design or an unavoidable one as there is no efficient way to signify this type of interaction. With an insufficiently signified design, users are still supposed to discover the interaction possibility thanks to the inefficient signifier, although for signifier-less designs we have to assume a different path of discovery. As a consequence, designers should also not expect the same reaction from users when they discover a signifier-less design than when they discover an insufficiently signified one.

Case 3 shows another limit of my definition: here, the interaction possibility is signified but "when the user needs it" does not happen. This design should not be labeled "signifier-less", and in the definition above, "when the user needs it" should be understood as assuming the user needs it and as only providing a restriction about when the signifiers should be exposed. For the same reason, the problem of the false affordance in case 4 is not concerned by my definition of signifier-less designs, which assumes true affordances.

In cases 5a) to 5c), signifiers are replaced by a more general act of explicit communication. The goal is twice: first, to remind that material signifiers are not the only possible source of information about affordances. Second, to reinforce the idea that the problem of signifier-less designs is not a problem of having something in a design, but having an information provided or not in a context of use that requires it. The difference between 5b) and 5c) also allows to question "when the user needs it": is it just before the user actually needs it? Or after she realized that she needs the information? If the information is only provided just before the user actually needs it, distracted users could miss the information, and in that case the interaction possibility is not signified anymore to them. So, the (un)availability of the information after the user realized she needs the information is what really matters in the definition.

6.1.2 *Directly signified affordances*

Finally, the question of the directness of the signifier has to be addressed, as when a user realizes she needs some information, she is free to try many different methods to obtain that information. In particular, it is possible that she knows exactly where to find the information. In the case 5a), for instance, the information could have been sent to her by mail and she forgot it, but when needing the information she remembers it was sent to her, and she can take her smartphone and search for the mail. Is it not "when she needs it"? An HCI example would be the hint provided in a menu about the hotkey that can be used to trigger the command: if a user needs to know this hotkey, she can simply open the menu and look for it.

The problem with this approach to how users can access the information they need is that it is extremely hard to make a distinction between these cases and others for which the method to access the information is not provided by the design or by the context of use. For instance, what if the user knows exactly what to search on her preferred web search engine to find the information as the first response? Or what to ask to her digital assistant? In many instances, the difference between a generic tool to access information like a search engine and an interface element specifically designed to provide this information is just a matter of user familiarity with the tool. The concept of directness is introduced in the definition to reject the use of such generic or specific search tools as "signifiers", and to accept as signifier-less designs interfaces that include such tools, so that we can also question the efficiency of these tools as sources of information.

I build the concept of directness of signifiers upon Gibson's idea that human displays afford some kind of mediated knowledge (see Section 3.6.2). The idea is that signifiers have a part that is an affordance for mediating information: either the signifier is directly a perceived affordance in Norman's meaning of the expression, either it is a sign that needs to be interpreted, but then, its design relies on an affordance for mediating information. Indeed, we are not concerned here with steganography, i.e., the art of hiding messages so that only the people who know that there is a message can see it. Signifiers are normally[1] recognizable as signifiers even before trying to interpret them: whatever is printed on a poster, it says "this is information" and "I am a signifier". So does a label, a pictogram, etc. They have an affordance for conveying information, and people can perceive this affordance directly in a Gibsonian way.

The definition of direct signifiers also builds on Kaptelinin' & Nardi's notion of handling affordance: if, from a position from which a user could reach the handling affordance of an interaction instrument, she can perceive an affordance for mediating information just by looking around, and it mediates an information about the corresponding instrumental affordance, then this instrumental affordance is directly signified. Otherwise, it is a signifier-less design.

---

1 This is actually the result of design and conventions. However, when it does not happen and people cannot perceive a signifier as being a signifier, it is more a question of a poorly designed signifier than signifier-less design.

6.2 historical signifier-less designs

Non-signified interaction possibilities have been provided by the user interfaces of most systems in the past and present. The goal of this section is not to retrace the history of signifier-less designs as an historian would do. The goal is, on one hand, to provide examples of signifier-less designs that will inform the study of this notion; and on the other hand to show that the phenomenon is not new, and not coming from the specifics of modern touch-based smartphones – although the phenomenon is quite visible with these devices. In the end, with this review, I want to ask the question: is it even possible to have interactive systems that rely on interfaces without any signifier-less design, or should we consider that signifier-less designs are an unavoidable aspect of human-computer interaction?

6.2.1 *Command-line interfaces*

Back to the time of command-line interfaces (Figure 6.1), the commands available were not signified permanently: users had to learn the list of commands recognized by the system, either from training or printed documentation. In the 70s, this documentation became digital and was included in the system so that users could access it on demand, and documentation for each command was provided when requested – both operations being realized with a dedicated command. To interact with a computer at that time, users had thus to memorize certain sets of abbreviations or hotkeys, or use cheat-sheets.

Command-line interfaces are still used today by some users for system administration, or for some activities available on all systems,

Figure 6.1: Example of command line interface.

and they still rely on the same mechanisms. Some programming and maintenance tasks may not have a graphical user interface and may still use a command line.

6.2.2 *Signifier-less designs in Direct Manipulation*

Graphical user interfaces had been proposed as an alternative to command line interfaces as early as the 60's and had been used as the primary interface in commercial products such as the Xerox Alto (1973) and Apple Lisa (1983), which defined the early forms of what has later been called WIMP interfaces (for Window - Icon - Menu - Pointer). According to Grudin [Gru06], the HCI community was skeptical at first about graphical interfaces and only embraced the idea that GUIs could be better than command-line interfaces after the commercial success of Microsoft Windows in 1985. Shneiderman's 1983 introduction of the concept of *Direct Manipulation*, based on the analysis of existing applications including text editors, spreadsheets, data analysis tools, and video games [Shn83] became then the main paradigm to study and design graphical interfaces and interaction techniques.

According to Shneiderman, the concept of direct manipulation follows four principles (quoted from [Shn83]):

> 1. Continuous representation of the object of interest.
> 2. Physical actions (movement and selection by mouse, joystick, touch screen, etc.) or labeled button presses instead of complex syntax.
> 3. Rapid, incremental, reversible operations whose impact on the object of interest is immediately visible.
> 4. Layered or spiral approach to learning that permits usage with minimal knowledge. Novices can learn a modest and useful set of commands, which they can exercise til they become an "expert" at level I of the system. After obtaining reinforcing feedback from successful operation, users can gracefully expand their knowledge of features and gain fluency.

The first and third of these principles clearly emphasize the need for visibility of the objects that can be manipulated and the need for visibility of the effects of these manipulations. Yet, none of the four principles of direct manipulation requires that action possibilities be signified to the user. This fact can be illustrated with the example of a drag-and-drop operation such as dragging a file icon on a trashcan icon to delete the corresponding file. This operation indeed follows the four principles of direct manipulation:

1. Both icons represent objects of interest that are *continuously represented*.

2. The drag gesture is a simple *physical action*.

3. This action is *rapid*, *incremental* (as the distance between the file icon and the trashcan icon is continuously reduced), *reversible* (by canceling the drag-and-drop or by putting back the file icon at its original location), and its impact is *immediately visible* (as the file icon is moving).

4. It uses a *layered approach to learning*, as this operation builds upon the knowledge of using drag-and-drop actions to move files in the file system and extends its meaning for the trashcan.

Yet, there is often no specific design element on the file icon itself that signifies it can be moved: at best, there is a visual feedback when the user places the mouse pointer over the icon. Users have to be taught this action possibility, or have to discover it themselves by exploiting a metaphor such as the desktop metaphor. Here, the drag gesture or the desktop metaphor belong to the basic layers of knowledge that users need to possess in the "layered or spiral approach to learning" that Shneiderman presents as the fourth principle above.

6.2.3 *WIMP interfaces*

The first systems relying on a graphical user interface in the 70s' and 80s' defined a style of interface called WIMP (for Window - Icon - Menu - Pointer). Since then, and still today, most desktop computer interfaces have followed the WIMP paradigm, which kept most of its identity as it evolved slightly toward modern desktop interfaces. In addition to their reliance on Shneiderman's principles of direct manipulation, these interfaces also present many features that are not signaled to the users (and are not always in the scope of direct manipulation).

CONTEXTUAL MENUS provide useful commands acting on some object of the interface, and are activated by clicking on this object with a secondary mouse button or while pressing a key modifier (Figure 6.2). While most objects give access to a dedicated contextual menu, this is not systematic and there is no specific sign on the objects that have a contextual menu.

HOTKEYS are combinations of keys that trigger some command, although there is often no indication that some combination of keys can be used at the time users would need it. Instead, users have to learn these combinations, either by consulting a manual or the hotkey customization interface (Figure 6.3), or by noticing and remembering the hotkey hints provided with the menu item that triggers the command needed (Figure 6.2).

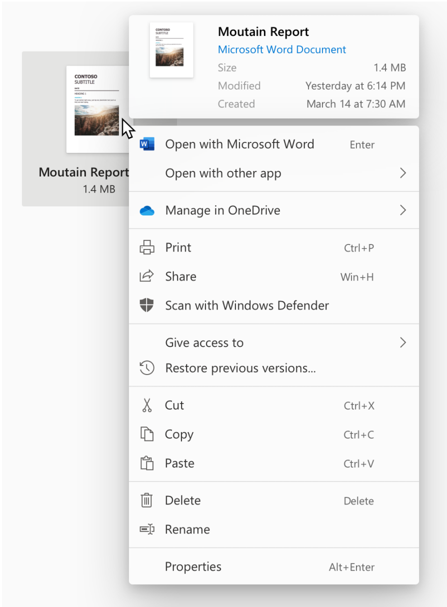


Figure 6.2: An example of contextual menu in Windows' File Explorer, displayed after the user right-clicked the file icon. Notice also how the menu provides hints for hotkeys on the right.

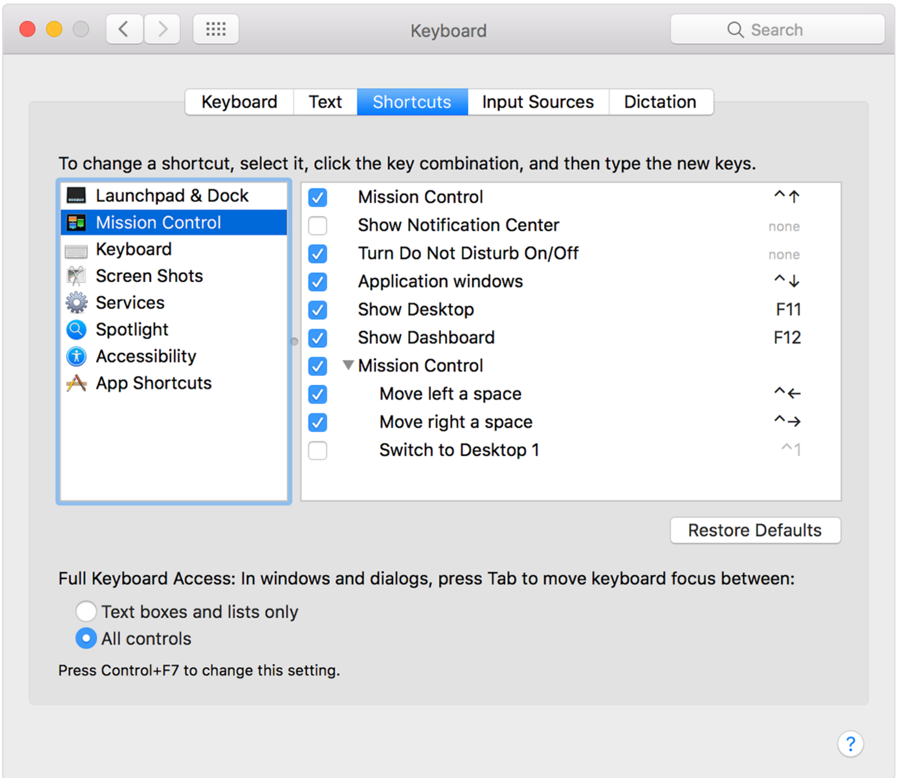


Figure 6.3: An example of system hotkeys on Mac OS X to navigate between windows and desktop spaces. These hotkeys are only hinted in this panel of the System Preferences application, where they can be customized.

The hints used in the later strategy may be thought of as signifiers, since users have a chance to see them when they select the command from the menu. However, from my point of view, this design is equivalent to a cheat sheet, as when the user selects the command in the menu, she does not need the hotkey anymore and she is likely not focusing her attention on the hint, and thus not learning the hotkey [Kur93, page 30]. It is more likely that the user decides to learn the hotkey at some point, and consults the menu to learn the key combination [GDB07]. Bailly *et al.* indeed observed that in the last uses of a command from a menu before transitioning to using hotkeys for this command, users spend more time using the menu than usually [Bai+18], which can be explained by a focus on the hotkey hint to memorize it.

In addition, the strategy of learning hotkeys from menu hints often suffers from two issues: first, not all hotkeys are hinted, even when the corresponding command is in a menu [Coc+14]. Second, some commands have no corresponding menu entries. In particular, selection and fast navigation keys are usually not shown in menus: the user may understand without hint that pressing the right arrow moves the caret to the next letter on its right in a text, but it requires more knowledge to understand that doing the same action while pressing the SHIFT key instead expands a selection to the right, or that using an ALT modifier key moves the caret to the next word instead of the next letter.

MOUSE POINTING Another non-signified interaction common in WIMP interfaces is the technique consisting in placing the mouse pointer in a specific area to trigger some commands or reveal some UI elements. This technique is used for instance in Mac OS to reveal the Dock, or the menu and window title bars, all three of which are hidden by default for an application in full-screen mode.

The hovering technique also allows to reveal tool-tips for specific widgets by hovering these widgets, usually with some dwelling time. Here, the visibility of the widget allows users to determinate the hover area, but as for contextual menus, nothing in the design of the widgets signifies to users that a specific widget has a tool-tip attached to it.

Other techniques rely on users placing the mouse in areas where they have no reason to expect something to happen if not told beforehand. For instance, Mac Os X and many Linux systems support "hot corners", a technique allowing to trigger some system-wide commands by sending quickly the mouse pointer in one of the four corners of the screen. In Windows 8, this technique reveals the Charms bar, an always-available tool that replaced the Start menu from earlier versions of this system.

MODAL INTERFACES *Modes* are different global states of the interface in which the system reacts differently to a same input. Modes can be considered problematic for interaction because users can forget in which mode the interface is, and the mode indicators are often not visible enough, as they are not under users' focus of attention. Yet, it is very hard to avoid the use of modes in interface design. For instance, the drag-and-drop operation described in Section 6.2.2 is modal: moving the mouse has not the same effect during the drag-and-drop (moving the file icon) than usually (simply moving the mouse pointer). For this reason, Raskin [Ras00] advocated *quasi-modes*, i.e. temporary modes that are only activated as long as the user actively maintains them, for instance by holding a button pressed. In that view, the information about the mode is put back "in the world" by user action, even if it was not provided by the system.

From the point of view of signifier-less designs, the problem with modes come from the fact that a same input can result in different commands according to the mode – or in other words, that a same handling affordance is connected to different effecter affordances through a mode that defines the active instrument. The lack of visibility of the mode can thus been understood as a problem resulting from the use of inefficient signifiers to signal the current mode, which is a concern that I discarded from my definition of signifier-less designs. However, it could also be argued that the mode only defines the set of active instruments, and that the problem is that these modal instruments are not signified directly: their instrumental affordance, which results from the connection of an handling affordance to a mode-dependent effecter affordance, is not signified directly but only through the signifiers of the mode. From this point of view, some modal interactions can be considered as signifier-less designs.

6.2.4 *Post-WIMP interfaces*

As WIMP interfaces knew a lasting success in commercial products, HCI research focused on establishing the concepts of so-called post-WIMP interfaces. In a seminal paper, Nielsen lists twelve dimensions across which new interfaces could diverge from WIMP [Nie93]. Among these 12 dimensions, two are particularly relevant for our concern: first, Nielsen advocates a change of *interface control* from being made "by user (i.e. interface is explicitly made visible)" to "by computer (since user does not worry about the interface as such)". Second, he advocates a change of *object visibility* from "essential for the use of direct manipulation" to "some objects may be implicit or hidden", as displaying everything prevents applications and systems to scale in size, and some objects will need to be managed by the system.

While Nielsen's second point concerns mainly objects of interest in an application, such as files in a file system browser, he adds in a follow-up paper [GN96] that to deal with the increase of interface controls, interfaces should use modes. He advocates to solve the problem of modes not by avoiding them, but by using richer cues, i.e., making the mode more visible.

What could now be considered as post-WIMP interfaces in commercial products are mainly the interfaces of smartphones and tablets, where Nielsen's concern of scaling the interface exists not because the interface has to contain more elements, but because it is displayed in a smaller screen. While these interfaces rely often on different panes or tabs for different types of activities and contents, and panes could technically be considered as modes, they usually have very different layouts that make the current mode quite obvious, as advised by Nielsen. However, smartphone and tablet interfaces often rely on non-signified interactions, notably navigation gestures and vocal interactions for digital assistants like Apple's Siri.

For navigation gestures, users may have to swipe to navigate between panes, to double-tap on an element to zoom on it, to pinch or spread two fingers to navigate in maps, as well as simply sliding a finger or performing a flick gesture to scroll. These gestures are believed to be part of a user's basic interaction vocabulary and to implement the system interface metaphor of manipulating physical objects, and are therefore usually not signified directly. There are sometimes indirect signifiers from which users can deduce an interaction possibility, such as little dots at the bottom of a pane to represent the different panes (one dot for each) and the active one (single dot with a different style). These signifiers are indirect in that they only represent the system state (see paragraphs on modes in Section 6.2.3), from which users can deduce the presence of an effecter affordance (to change pane), but nothing communicates the handling affordance (swiping), which users have to understand by recognizing a convention. Thus, users can learn to perceive such dots as signifiers for the navigation gestures, but the efficiency of such signifiers depends on user's expertise.

For voice interactions, a problem is similar to the one of command line interfaces in that users have no way to know what commands they can speak: these commands and their syntax are not signified. However, the problem is also different in that the assistant is supposed to be smart and understand free-form questions, so that users can use their natural communication skills to issue commands. Yet, from the point of view of instrumental affordances, the question of knowing how to communicate the command to the system is a question of handling affordance, and digital assistants also have hidden effecter affordances in that users are often unaware of what the assistant can do for them beside searching information on the web.

6.3 TYPES OF SIGNIFIER-LESS DESIGN

From the definition of signifier-less designs and the examples presented in the historical review of signifier-less design, we can observe different types of signifier-less designs.

Since I have defined signifier-less designs in Section 6.1 on the basis of a missing signifier for an instrumental affordance, and since instrumental affordances result from the connection of a handling affordance with an effecter affordance in an instrument, signifier-less designs can be either:

PUZZLING: The handling affordance lacks signifier.

HIDDEN: Both handling and effecter affordances lack signifiers.

MYSTERIOUS: The effecter affordance lacks signifier.

In *puzzling* signifier-less designs, the effecter instrument is signified to the user, but not the handling affordance. Users can thus know that a possibility to do something exists but they do not know how to do it. This is for instance the case of the handle-less cabinet mentioned by Norman in the quote introducing Chapter 1: it is clear that there is a cabinet with drawers, the drawers are even clearly visible, and can thus serve as signifiers of the effecter affordance, which is that they can be opened. But the drawers have no handle, so users do not know how to open them: the handling affordance, which consists here in pushing a drawer's front surface, is not signified. In a touch-based mobile user interface, the equivalent would be, for instance, a scrollview that can be scrolled directly by dragging its content with the finger. The content flowing out of the visible window would be a signifier that the view can be scrolled, and so the effecter affordance is signified (see Section 5.1.3). But the handling affordance, consisting in dragging the content, is not signified.[2]

Puzzling signifier affordances can lead users to face a wall, explore the interface in search of the handling affordance, or call for an external help. This can be a terrible experience, and users can blame themselves for their incompetence instead of blaming the design. But not always. Indeed, some people love challenges and figuring things out, and succeeding at doing so brings an intrinsic reward in the form of self-satisfaction. After the facts, it can also be an experience that one can share with their belongings and make fun of it.

In *hidden* signifier-less designs, neither the handling nor the effecter affordance are signified. Users may thus simply never get to know that an interaction possibility exists. To expand on the example from Norman's quote, hidden signifier-less design would now be the case

2 Note that it would not be a signifier-less design if the dragging was made with a mouse, as in some document viewers, and the mouse cursor changed for a hand that signify "dragging".

where "the cracks that signify the existence of a door [are] eliminated. The result can be a smooth expanse of gleaming material, with no sign of doors [. . .], let alone of how those doors [. . .] might be operated." Many hidden signifier-less design showed up in the previous section, such as command line interfaces, gestural interfaces, and vocal interfaces. *Swhidgets* are also hidden signifier-less designs, as neither the swipe gesture (handler affordance) nor the availability of further controls are signified (as for contextual menus, the effecter affordance of *swhidgets* consists in providing access to further controls).

The discovery of hidden signifier-less designs can cause mixed reactions: on one side there can be the feeling that the designers hid something useful to you. Users may thus blame the designers for missed opportunities. They may also question their competence with thoughts like "was I supposed to know that?" On the other hand, users might simply be happy to discover a new feature and appreciate how it might positively impact their life in the future. The design itself can sometime be thought of as clever, and users might also get a small boost in their feeling of being competent for discovering – and now, knowing – the hidden feature.

There is not always a clear distinction between hidden and puzzling signifier-less designs, as for some users it might be obvious that the interface must provide some feature, and so there must be a corresponding effecter affordance. Even if this effecter affordance is not signified in the interface, the application / product / system could sometime be considered as a signifier in itself. For instance, in a messaging application, you would expect to be able to send messages: there is an implied effecter affordance for sending messages, communicated to you even before you open (or even, install) the application.

Finally, in *mysterious* signifier-less designs, there is a signifier for the handling affordance but not for the effecter affordance. This situation is usually described in the HCI literature as a lack of *feedforward* (see Section 5.3.7). A typical (and exaggerated) example would be a big red button with no label on it and nothing around it: there is no way to know what it is for, except by trying to press it, with potentially dangerous results. A more common example would be an array of unlabeled light switches that control different lights: unless the switches and lights are configured to create a natural mapping (see Section 5.3.4), users cannot know which light is controlled by a given switch. However, in that case, it is also possible to consider that the placement and type of switches signify, by convention, that they control lights, and that the problem is not that the effecter affordance is not signified, but that it is imprecisely signified: we know that a switch has an effecter affordance of controlling a light, but we do not know which light. A related user interface example would be the three buttons at the top-left of every window in MacOS, which can be used respectively to close the window, minimize it, or put it in full screen: on recent versions of the OS, these buttons do not show the usual cross, minus sign, and diagonal double-arrows that used to represent the corresponding commands (they are only displayed when the mouse hovers the group of buttons), but they are still drawn respectively in red, orange and green. Although the placement and color of these buttons can be considered as signifiers for users with a minimum of knowledge about WIMP interfaces, they might as well be mysterious signifier-less designs for new users[3] – or when the function of the button change, such as when a new version of MacOS replaced the "maximize window" function with the current "enter full-screen mode" one.

This type of signifier-less design causes less usability problems than the two others, since it has the advantage of being discoverable: as users are aware of the handling affordance, they can try it to discover its effect. Yet, some users might be afraid of trying something they do not understand. If the unknown effecter affordance is an important feature for the user, she might never discover it because of that. In other cases, trying might not provide the required information, because the results are not visible (in addition to the problem of feedforward, there is a feedback problem).

3 E. g., a young child familiarized with user interfaces through touch-based devices and toys, but who has never used a desktop computer with a WIMP interface.

Part III

PROPOSITIONS FOR A MODEL OF DISCOVERY

7 DISCOVERING AND ADOPTING AFFORDANCES

7.1 ABOUT THIS CHAPTER

When an affordance[1]is not signified, users may not know about it, or may not think about using it even if they know it – simply because nothing is there to remind them about this possibility. Conversely, when a user do not employ a non-signified interaction, is it because she does not know about it, because she forgot about it this time, or because she prefers another way of achieving her goals? Although this question can be asked for any interaction technique, it is particularly relevant for signifier-less designs: since the knowledge about them is not "in the world" and has to be "in user's head", a major factor controlling their usage is thus the user's ability to "think" about them when she needs it – an ability that I will call the user's *proficiency* in this (set of) signifier-less design(s) to distinguish it from the mere notion of performance (indeed, the technique might actually be less efficient than alternatives) or usage of the technique (it may be perfectly known but not used).

This chapter concerns the mechanisms by which user proficiency in some interaction technique can improve, in general and for signifier-less designs in particular. To investigate this question, I propose a model of proficiency improvements that involves three components (as illustrated in Figure 7.1):

1. *Current knowledge and skills for the interface/task* are user attributes that affect her proficiency: the user's knowledge of the system (including the non-signified interaction possibilities); how skilled she is in using the different parts of the system; her habits and preferences; and ultimately, how she envision the tasks she's using the system for and what place she wants the system to have in her lifestyle.

2. *Design Means* by which a user can improve her proficiency by modifying some of her current knowledge and skills. This aspect of the model has already been described in Chapter 5 as consisting mostly in signifiers, and will thus only be discussed in this chapter in relation with the other two components.

1 I use "affordance" rather than "interaction possibility" in this chapter, because as we have seen in Chapter 3, affordances are not only interaction possibilities, they also have meaning as elements of a system of affordances, and this aspect might be important in their discovery.

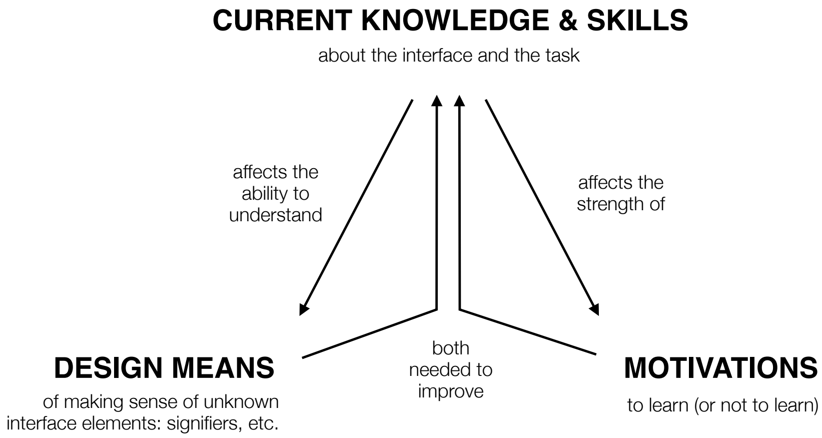


Figure 7.1: Proposed model of proficiency improvements, consisting in the three dimensions and their interactions (see the text for explanations).

3. *Causes and motivations* for such changes, which explain why a user would (or would not) improve her current knowledge and skills about the interface.

I will describe the state of the art for each of these three components in the following sections, then dedicate sections to their interactions. Finally, I will present two personal contributions in these areas: *degrees of knowledge*, a simple model of how well something is known and how this knowledge evolves; and *sources of knowledge*, a generalization of the concept of design means;

7.2 CURRENT KNOWLEDGE AND SKILLS

Defining exactly what a user knows about an interface and how skilled she is in using it is beyond the scope of HCI as it would rather be the domain of Psychology. However, HCI researchers have proposed simple models of user knowledge and skills that are generic enough to fit any type of interface, and have used such models to understand user behavior and provide design guidelines.

The simplest of these models, and probably the most used, is a distinction between *novice* and *expert* users. These common-sense notions are however rarely defined precisely, and while it allows to raise the question of how users transition from being novice toward being expert, answering this question requires slightly more precise notions (see [Coc+14] for an example and review).

Since I am interested in understanding how users can develop their knowledge and skills, I will first focus on how the interaction with an application or device requires knowledge and skills in a variety of domains, and on how these knowledge and skills, far from being independent from each other, are connected in structures similar to the *stratified hierarchies* already discussed for affordances in Section 3.2.1. I will thus discuss how users can exploit this structure to expand their knowledge, and develop usage habits on top of the newly acquired knowledge.

7.2.1 *Domain of the knowledge and skills*

This dimension describes what is concerned by the knowledge and skills that users need to possess to interact with a system. It is a *stratified hierarchy* inspired by some of the metrics for expertise defined by Grossman and Fitzmaurice [GF15], and particularly the metrics of *scope* (with three different sub-levels: UI, commands and task), *appropriateness of workflows*, and *domain knowledge*.[2]

GENERAL KNOWLEDGE AND SKILLS This level is about knowledge and skills developed by people to interact with the world in general – rather than for a specific interface or activity – and especially with the non-computerized aspects of their daily lives. For instance, the Reality-Based Interaction (RBI) framework [Jac+08] lists four "themes of reality" that interface designers can build upon:

- *naïve physics*: people have common sense knowledge about the physical world.
- *body awareness and skills*: people have an awareness of their own physical bodies and possess skills for controlling and coordinating their bodies.
- *environment awareness and skills*: people have a sense of their surroundings and possess skills for negotiating, manipulating, and navigating within their environment.
- *social awareness and skills*: people are generally aware of others in their environment and have skills for interacting with them.

Interfaces often rely on real-world metaphors so that users can reuse their general knowledge and skills. Some interfaces can also rely directly on some general knowledge and skills, as with the object manipulation skills involved in tangible interfaces.

UI LEVEL At this level, knowledge and skills concern the use of user interface elements like widgets and gestures, independently of their role in an application's support of some activity. This is the knowledge and skills that someone can use to make sense of the

2 Since Grossman and Fitzmaurice are interested in the measure of user expertise from the usage of the system, they distinguish low-level expertise measures (scope and its sub-levels) and high-level ones (appropriateness of workflow and domain knowledge), as the former only concern the expertise of the interface and the latter concern more generally the activity supported by the system.

interaction possibilities provided by an unknown application, without ever knowing what the application is for.

For instance, skills at that level would be typing with a keyboard, pointing with a mouse, using a scroll bar, or selecting a second-level item in a menu. An example of knowledge at that level would be knowing that clicking a menu label will open the corresponding sub-menu – independently of what the menu actually contains.

This level exists because applications generally reuse interface components provided by the operating system (or by libraries possibly shared with other applications), which in turn often rely on some similarities with the interfaces of other systems. It is therefore assumed that, to the exception of some widgets specific to an application (like Microsoft Word's Ribbon), users already possess knowledge and skills at the UI level before even facing an application that require this knowledge.

COMMAND LEVEL At this level, knowledge and skills concern the features provided by an application. Knowledge at this level would typically be the topic of application manuals or help files. For instance, knowing what font modifications (such as bold, italic and underlined) are available in a text editor and how to apply them to some text, or knowing what type of blur effects are provided by an image manipulation application and how to select the right blur parameters to obtain the desired effect. This level includes knowledge and skills that affect a user's awareness of the command and efficiency in triggering and applying the command. It would thus include the fact that the button labeled B sets the text in bold face, but would not include knowledge and skills about how to press a button, which are at the UI level. In other words, this is the knowledge about the mapping between UI elements and the effects they cause outside the UI – i.e. in the states of the world, of the documents manipulated in the application, or of the application itself – and the skills involved in manipulating the UI elements so that their effects match a desired result.

TASK LEVEL At this level are knowledge and skills about how to perform actual *tasks*, defined by Grossman and Fitzmaurice as a target goal that a user wishes to accomplish: "Typically the goal is to achieve a desired effect on the application content or environment, and the user must choose an appropriate method to accomplish this goal, made up of a sequence of actions" [GF15].[3]

Some tasks can be achieved by the use of a single command, in which case the distinction between the task level and the command level is blurred – although it stills exists: at the command level, the focus is on what the command do, while at the task level the focus

3 The reader will recognize in this definition the stages of Intention and Sequence of Actions from Norman's model (see Section 4.1).

is on how to achieve some result with the command. For instance, knowledge that a text can be made more visible by using a bold typeface would be at the task level, knowledge that the button labeled "B" (un)sets the selected text in bold face would be at the command level, and knowledge about how to press a button would be at the UI level.

Other tasks require the combination of multiple commands. For instance, a task like "printing the document edited" might involve many commands to choose what printer to print with, what portions of the document must be printed, the quality of the printing (ink, paper and speed), or options to save paper (two-sided, multiple pages per sheet of paper, etc).

Knowledge at the task level would typically be the topic of tutorials and applied lectures. It may consist in the awareness of different possible plans of action to reach the expected goal, such as using a menu entry or a shortcut to trigger a command. Skills may concern the ability to evaluate the efficiency of each plan of action in different contexts and the ability to choose the most efficient one depending on the current context. But knowledge and skills at the task level can also simply consist in the knowledge of procedures of the style "to achieve result X, do steps A, B, C" and the skills required to apply such procedures (memorizing the steps, recognizing the corresponding states of the application, etc).

appropriateness of workflows This level describes knowledge and cognitive skills involved in the decomposition of tasks into sub-tasks and ordering of tasks themselves.

For instance, a user can add five rows in a spreadsheet by repeating five times the task of adding one row, or she can know a way to add the five rows at once. Obviously, in this example, users can only choose the appropriate workflow if they have the task-level knowledge about how to add a single row and how to add multiple rows at once. However, even when they know how to add multiple rows at once, some users may prefer adding the rows one by one, or they might simply not think about using the more efficient method. There are thus awareness and skills involved in this choice, which is more about the appropriate way to think about the task than knowing the different ways to obtain a result.

Similarly, in their discussion of how applications can provide *support for advanced task strategies*, Cockburn *et al.* [Coc+14] provide the example of creating three identical objects made of the same parts A-B-C with a drawing application: a novice strategy might be to create the parts A, B, and then C of a first object then repeat these steps for the second object, and then for the third; but it can also be more efficient to draw three parts A, then three parts B, then three parts C, as it may require fewer changes of drawing tool; alternatively, a user who knows about grouping shapes may prefer to create the three parts of a first object, group them, and then copy/paste the group to create the two other objects. These three strategies illustrate three different ways to think about the task: they are not as much about getting the desired result (which would be at the task level), but about organizing the workflows by realizing that operations can be reordered to minimize tool changes, or by understanding how a specific command like "grouping" can be used.

DOMAIN KNOWLEDGE This is the general knowledge about the activity supported by an application rather than knowledge of the application itself. For instance, it could be knowledge about how to write a research paper, by opposition to the knowledge of a specific word processor used to do it. Domain knowledge implies the awareness of specific tasks (for instance, citing another paper, adding a figure) and it can suggest to users that an application should provide some commands to accomplish these tasks, but it does not include the specific way to use these commands to accomplish these tasks (these are knowledge and skills from the previous levels).

Domain knowledge can also induce the choice of appropriate workflows, although not being directly about the workflows. For instance, a user that receives a lot of emails may have developed specific knowledge and skills such as identifying quickly spams and e-mails that need to be replied with a higher priority. This user might adapt her workflow in order to exploit these knowledge and skills. For instance, she could delete spams and flag important mails directly from the list of newly received mails, based on the information and short extracts provided by this list, before reading the flagged mails individually. On the other hand, a less experienced user might prefer to read each mail individually before deciding to delete/ignore/reply it with the widgets provided in the mail view.

Similarly, when users know things like the relative frequencies of tasks in an activity, what tasks usually follow each others, and the type of materials each task requires, they can organize their work environment on the basis of that knowledge to augment their productivity: The structures resulting from such organizations have been called *taskonomies* by the anthropologists Janet Dougherty and Charles Keller [Nor06].

7.2.2 *Improving knowledge and skills in time*

According to the influential model of Fitts and Posner [FP67], people repeatedly engaged in an activity go through three phases of acquisition of the knowledge and skills required for this activity: cognitive, then associative, and then autonomous phases [Coc+14]. As Rasmussen remarks, other authors have proposed categorizations of human behaviors into three categories similar to the cognitive/associative/autonomous trichotomy [Ras83].

The first phase of skill improvement, the cognitive phase, is dominated by learning *what* has to be done. It involves a substantial amount of declarative knowledge (i.e., knowledge that can be expressed as sentences). It is slow, attention-demanding, serial in nature, and strongly *volitional* in that the activities can easily be avoided or interrupted.

The second phase of skill development, the associative phase, is dominated by learning *how* to do the activity and is characterized by improvements in the motor actions through subtle adjustments of execution, leading to slower but more stable performance gains than in the cognitive phase. The declarative knowledge acquired during the cognitive phase is largely unused and replaced with procedural knowledge ("know-how") that is unconscious an can hardly be expressed as sentences. For some activities (e.g., playing an instrument), this phase can last years.

Finally, the autonomous phrase, is characterized by *fast*, *not attention-demanding*, *parallel in nature*, and *not volitional* actions. The execution of tasks in this phase are also less demanding cognitively and physically.

The existence of these three phases of acquisition of knowledge and skills implies that user performance improves in time, and thus establishes a relation between a user's knowledge and skills and her performance. This relation will be analyzed in Section 7.3.1 as an interaction between a user's current knowledge and her ability to exploit design means, and in Section 7.5.1 as an interaction between user's current knowledge and performance-related motivations.

7.2.3 *Dependency across levels*

We can expect learning of the higher levels to build over the knowledge of lower levels. This is for instance a core concept of *direct manipulation* interfaces, as expressed in the concern of *layered learning* (see Section 6.2.2). This view integrates two concepts: the idea that users need to know an interaction technique before being able to use another one that relies on the former; and the idea that users can understand new interaction possibilities using their knowledge of other interaction techniques.

The example of deleting a file by drag-and-dropping it on the trashcan icon, already discussed in Section 6.2.2, illustrates this: First, it involves a *drag-and-drop* and this technique must be known before being able to perform a *drag-and-drop to the trashcan*. This is a case of dependency between two layers of knowledge, the knowledge of how to perform a drag-and-drop belonging to the UI level, while the knowledge about using a drag-and-drop to delete a file belongs to the command level. Second, it expands the domain of application of the drag-and-drop and thus allows users to re-frame their understanding of what a drag-and-drop is for, to see it not as a specific interaction technique to move files in the file system, but as a more general interaction technique to act on files. This generalization of the knowledge about the purpose of the drag-and-drop could then serve as an example for further generalizations (for instance, maybe drag-and-drop does not work only on file icons, but also on other types of data?).

A consequence of this second aspect of layered learning (i.e., understanding) is that users may have the knowledge of some possibility – because they understood it and remembered it – and may still not have integrated this knowledge into skills, strategies, or other forms of behavior that rely on this knowledge (i.e., they have not yet reached the associative or autonomous phases). In particular, for signifier-less designs, a problem is that users may know how to interact at the UI level despite the lack of signifier, but may not have integrated this knowledge in how to trigger some commands. A typical example is provided by hotkeys: people may know there is a hotkey or that there is likely one for some command, but not have the reflex of using the hotkey (or searching for it) when they want to trigger the command.

This is another problematic dimension of signifier-less designs: not only the lack of signifiers make them hard to discover, but it also make it hard to remember them when they are known. As a consequence, a model of users' knowledge of signifier-less designs such as the one I will propose in Section 7.6 needs to integrate the distinction between knowing an interaction technique and adopting it.

Conversely, the role of layered learning in allowing to understand other interactions can be bypassed: tutorials and other sources of knowledge about the interfaces sometime provide "recipes" to follow to reach some result, without explaining why they work – either from the point of view of the sub-interactions and interface concepts involved, or from the point of view of how this specific way of doing things is a best practice for the activity. Users can then have an incomplete view of the relation between one piece of knowledge and other pieces of knowledge in other levels, which, if they were knew, could help users adapt more easily to new situations or tasks. This is a problem that is well known in pedagogy, that has implications on signifier-less designs when their discovery depends on external sources of knowledge, but this concern will not be studied further in this thesis as the priority is to identify the possible external sources of knowledge and bring evidence of their use.

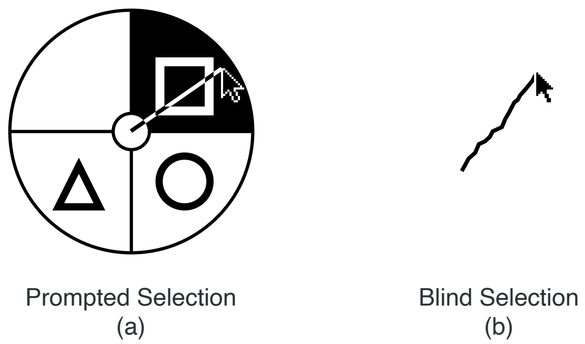


Figure 7.2: Kurtenbach's *marking menu* is a pie menu (a) that only appears around the mouse cursor after a short delay, providing an *expert mode* (b) for users who already know well the menu and can perform the corresponding gesture before the menu appears (in the pie menu, items occupy sectors of a disk and are triggered by leaving the sector at the disk's border). Adapted from [KB91].

7.3 INTERACTION BETWEEN CURRENT KNOWLEDGE AND DESIGN MEANS

7.3.1 *Current understanding affects the usefulness of means*

The efficiency of design means such as signifiers in helping users acquire new knowledge about the interface may depend on the user's current knowledge and skills themselves. Indeed, some design features can be used as clues to improve knowledge and skills only if the user already possess some knowledge that will allow her to interpret these features as signifiers. This phenomenon is already recognized implicitly in interaction design paradigms such as the layered approach to learning of *direct manipulation* (see Sections 6.2.2 and 7.2.3), but it is worth analyzing more in details some aspects of the phenomenon and how it can be exploited in design.

Transfer of knowledge and skills between applications can rely on this phenomenon: the knowledge of some interaction contexts allows users to recognize similar contexts, and this similarity becomes a new design mean to improve the general knowledge and skills of these users. It may be an important factor in the discovery of signifier-less designs such as the *item swhidgets* in the iOS Mail application (see Section 2.1), where the list layout can act as a signifier that items in the list may hide *swhidgets*, an interpretation that a user can only make if she already knows about *item swhidgets* in other list layouts.

Rasmussen has described how a user's current knowledge and skills affects the kind of clues that she will pick up in the interface to guide her behavior [Ras83]. His model is based on the distinction between the cognitive, associative and autonomous phases of acquisition of knowledge and skills (see Section 7.2.2), and is close in principle to the Seven Stages of Actions (see Chapter 4), which was proposed later and is simpler. In Rasmussen's model, there are three types of behaviors that he names *knowledge-based*, *rule-based*, and *skill-based*, each type of behavior being predominant in (respectively) the cognitive, associative, and autonomous phases. These behaviors and the corresponding usage of design means will be described below, taking the design of Kurtenbach's *marking menus* [KB91; Kur91; Kur93] as an example.

In *knowledge-based behaviors*, users rely on what Rasmussen calls *symbols*,[4] which "represent other information, variables, relations, and properties and can be formally processed. Symbols are abstract constructs related to and defined by a formal structure of relations and processes – which by conventions can be related to features of the external world." He provides the example of a dial's hand displaying an abnormal value (and thus, at an abnormal position), which users interpret as a need to understand why it happens using her mental model of the system (maybe it is a pressure gauge and there is a leak in the system?), eventually updating this mental model.

Such a knowledge-based behavior can be seen in how novice users of marking menus would understand them: the appearance of the menu, with its labels and sectors in a circle (see Figure 7.2-a), would not fit their current understanding of what a marking menu is or how to use it. They would thus need to interpret the circle and its label as menu items, the radius of the menu as a distance they need to travel with the mouse to trigger the selection of an item, and the direction of the mouse movement as the way to select the desired menu item by moving the mouse in the corresponding direction. Feedback and feedforward can help them create such a model by exposing respectively its dynamic behavior and its purpose (as Rasmussen discusses [Ras83]). Once they have this model of how a marking menu works, they can start using it, but even then, they still need to know what item to select according to their needs. To do so, they must interpret the menu labels as symbols related to commands, in order to find the one that matches their needs: this identification, although simple, is an example of the kind of problem-solving behaviors that characterize the cognitive phase. At that point, the user only has to execute the item selection, relying on feedback to be sure that she does it correctly.

For signifier-less designs, the lack of "symbols" advertising an affordance is an obstacle to knowledge-based behavior. It can however still be triggered by an accidental revelation of *swhidgets*, or in case users try to manipulate the interface elements in ways that are known

4 Rasmussen does not use the words *symbol* and *sign* in the same way as Peirce does (see Section 5.2). However, Rasmussen's distinction between signals, signs, and symbols can be understood as an instance of a Peircean trichotomy in Firstness, Secondness, and Thirdness respectively. Indeed, signals are interpreted directly into calls for action, "signs" are interpreted as referring to a rule telling what action should be taken, and "symbols" are interpreted as a need to come up with a mediating explanation before determining what action should be taken.

to be compatible with the physical metaphors used on the system (not only for *swhidgets*). In addition, knowledge-based behavior may be triggered by other factors than observing a symbol that does not fit users' current understanding. For instance, it could be triggered by realizations of an interaction need, such as "I need a command to do X" or "that would be helpful if there was a faster way to do Y". Exploring the interface next to find a way to satisfy these needs would be a knowledge-based behavior (see Section 4.2.4).

In *rule-based behaviors*, users rely on what Rasmussen calls *signs*, which "indicate a state in the environment with reference to certain conventions for acts. Signs are related to certain features in the environment and the connected conditions for action. Signs cannot be processed directly, they serve to activate stored patterns of behavior." He provides the example of the dial's hand now being interpreted differently depending on the state of the system as indicating a normal state of operation, or a need for adjustment, or a need for re-calibrating the system.

Rule-behavior can be seen in how users of marking menus with an intermediate level of expertise would use them: in their knowledge of the menu, they would have rules like "to draw a square, select the square icon" or "if the label is grayed, the command is not available". At this point, they have integrated the knowledge of the basic operation of the marking menu, but they still rely on the labels to be displayed to know how to interact. They can however simply recognize the label they need instead of trying to interpret it: the labels do not need to act as feedforward anymore, but only to act as a landmark guiding the user.

Because rule-based behavior is based on the recognition of "signs", and any aspect of the interface can be considered as a sign, it is perfectly possible to have rule-based behaviors for signifier-less designs. For instance, a list layout can be recognized as such, and activate the rule "to delete an item, attempt a leftward swipe on it". In that case, there is however a question of the effectiveness of the rule for predicting the availability of *swhidgets* on the items of the list, which is a question of consistency in the design of lists across applications.

In *skill-based behaviors*, users rely on *signals* from the interface, which "are sensory data representing time-space variables from a dynamical spatial configuration in the environment, and they can be processed by the organism as continuous variables". He provides the example of the deviation of a dial's hand relatively to the set point, which can be tracked continuously and maintained close to zero with a control loop relying only on simple psycho-motor skills.

Skill-based behavior can be seen in how expert users would use the marking menu: they would know in what direction they should move the mouse without even looking at the menu, and all they need is a graphical feedback to be sure their gesture has the right direction and length. As a consequence, they do not need to wait for the menu to be displayed and can do the "blind selection" of the expert mode shown in Figure 7.2-b.

Similar skill-based behaviors can be observed for signifier-less designs, in which continuous variables like the positions of animated elements and mouse pointers can be tracked continuously to control the precise timings and movements of a skilled manipulation. Indeed, at that point, users do not need the signifiers that are missing for novice users (which is why signifier-less designs are often considered as expert techniques, both by their designers and by users).

7.3.2 *Learnability of the interface*

The notion of interface *learnability* has been recognized as an important aspect of interface usability and can be defined broadly as "to what extent the interface supports learning (eventually for different types of users)". Learnability can thus be understood as a global measure of the efficiency of the design means. A 2009 survey about the usage of this term in the HCI community [GFA09] revealed that definitions of learnability focus on the efficiency of the design means but mostly ignore users' motivation for learning about the interface beside performance improvement (which will be discussed in Section 7.5.1). Many definitions acknowledge, however, that different populations of users with different levels of expertise have different needs in terms of learnability of the interface – which is a way of presenting the concern that actually encapsulate two aspects: 1) the fact that these populations need to learn different types of knowledge and skills, and 2) the fact that the efficiency of the design means depends on user's current knowledge and skills. Another way to say this is to observe that in the left part of Figure 7.1, there is a dependency loop between design means and current knowledge and skills, and that such a loop can create complex behaviors in terms of the time-related aspects of learnability.

The fact that learnability is a global measure of the efficiency of all the design means in the interface hides another subtlety, which is that the global learnability of the interface may be greater or lesser than the sum of the efficiencies of the individual design means it is made of. We have seen that when transfers of knowledge are possible, learning one interaction contexts can improve the learnability of another interaction context. Conversely, when multiple things must been learned, they compete for users' learning efforts (as we will discuss in Section 7.5.2). While there has been a great amount of works that concerned the efficiency of specific means, as we have seen in Chapter 5, there is much fewer works that provide insights on how to make different parts of the interface contribute together positively to the overall learnability of the interface. The guidelines in this domain seem to fall down to "take care of consistency", although consistency can also be considered harmful sometimes [Gru89].

Concerning the dynamics of learnability, Kurtenbach has provided three design principles to support learning and using gestures: revelation, guidance, and rehearsal [KMB]. *Revelation* is defined as "the system should interactively reveal information about what commands are available and how to invoke them". Revelation is illustrated in how novice users can understand the whole design of a marking menu, as we have discussed in the paragraph about knowledge-based behaviors. *Guidance* is defined as "the way in which revelation occurs should guide a user through the method for specifying the complete command in any specific situation". It is also illustrated in how novice and intermediate users can use the labels to know in what direction they should move the mouse to trigger the corresponding item, and in how the system provide feedback as they do so. Finally, *rehearsal* is defined as "the way guidance is provided should require a physical rehearsal of the way an expert would issue the command". Indeed, users can perform the same gesture to select an item in the expert mode than with the menu displayed. This form of rehearsal is believed to help users learn the most efficient way to make the gesture and is characteristic of the associative phase (Section 7.2.2).

Kurtenbach's design of marking menus exploits the fact that skilled behavior requires less design means than novice behavior (according to Rasmussen's model), so that visual feedforward can be removed for expert users. A popular approach to the design of interfaces does exactly the opposite, choosing to disclose complexity as the user progresses toward expertise: advanced features are hidden first until users have the necessary knowledge to understand the benefits of their use. This approach can be seen in applications with "simple" and "expert" interfaces, buttons labeled "advanced options" or "advanced search", and panels with extra commands that are hidden by default. Customization of the interface is also related to this approach, as far as the selection of a default configuration is concerned. Indeed, with a simplified interface, novice users can find unknown features more easily by problem-solving or exploration, simply because there are fewer possibilities to consider. However, this approach cannot be justified by considerations for learning alone, as it also has other benefits. For instance, it can help making the product less intimidating to new users and better expose how it is suited to their needs (see Section 7.4.3 on User Experience). It can also make the interface more efficient for novice users by providing less controls, but more efficient signifiers and controls (e.g., bigger buttons that are faster to click on, tool palettes that provide better sensory affordances – see Section 5.1.2) – although advanced users may then need ways to regain the loss of performance caused by switching to the advanced, less efficient interface, such as hotkeys or gestures.

7.4 MOTIVATIONS

Motivation is the experience of desire or aversion: a desire to reach or to possess certain things or status, or an aversion that push you to avoid something. Here, I address the motivation for users to improve their knowledge and skills related to the interaction with a system's interface and the activities it supports. I first address the difference between intrinsic and extrinsic motivation, and then present different psychological theory that highlight different types of rewards. The last subsection will address causes for learning that cannot be linked to any motivation.

7.4.1 *Extrinsic vs. intrinsic motivation*

According to Richard Ryan and Edward L. Deci, motivation can be separated into extrinsic and intrinsic ones [RD00b].

Extrinsic motivation comes from external sources: you are motivated to do something because of its consequences, including the possibility to get various forms of rewards or to avoid negative effects (treats, punishments, etc.) For instance, an extrinsic motivation to have a better knowledge of some tools could be that you believe it would allow you to be more efficient in your job, and later be in position to ask for a salary rise.

Intrinsic motivation comes from internal sources: you are motivated to do something because you value doing this thing in itself, not because of its consequences. For instance you could play a musical instrument not because you want to make music but because you appreciate doing so: the activity that you are motivated to do is its own reward. A particular case of intrinsic motivation is the natural, inherent drive to seek out challenges and new possibilities that Self-Determination Theory (see below) associates with cognitive and social development.

There can be both intrinsic and extrinsic motivations at the same time. For instance, reading a novel before sleeping might be valuable in itself (intrinsic motivation), but can also bring the reward of a peaceful mind for a better sleep (extrinsic motivation). Similarly, practicing some sport might be fun in itself, but can also be a mean to improve your health or physical appearance.

The distinction between intrinsic and extrinsic motivation has been used to show that offering an external reward for a task that was intrinsically motivated could actually undermine the motivation instead of reinforcing it, or to observe that the efficiency of extrinsic motivators tend to decrease in time faster than the efficiency of intrinsic motivators.

The distinction is based on the source of the motivation: what is the motivator, rather than how it motivates you. Other classifications of motivations exist on other bases, such as the distinction between push and pull motivations that are based on how you experience the motivation (it is a push if you feel that you have to push yourself toward the activity, but it is a pull if you feel attracted toward the activity or reward, such as when you act by curiosity or empathy). A complete account of all such classifications is out of the topic of this dissertation: I only presented the distinction between intrinsic and extrinsic motivations because it is pertinent in the context of learning and discovery, where external motivations such as performance improvement call for cost-benefits trade-off evaluations with consequences on the learning (as we will see in Section 7.5.1), while intrinsic motivations such as curiosity do not.

7.4.2 *Self-determination theory*

Ryan and Deci created the Self-Determination Theory [RD00a] (later simplified as SDT) in the 70s. It is a macro theory of human motivation and personality, which concerns people's inherent growth tendencies and innate psychological needs. SDT focuses on the degree to which an individual's behavior is self-motivated and self-determined. It introduced the intrinsic/extrinsic dichotomy of motivations and additionally considers three basic psychological needs, which lack of satisfaction causes negative consequences globally, while their satisfaction by an activity is associated with positive consequences of the activity:

- *Competence* is the need to seek to control the outcomes of one's actions and experience mastery.
- *Autonomy* is the need or desire to be causal agents of one's own life and act in harmony with one's integrated self.
- *Relatedness* is the need to interact with, be connected to, and experience caring for others.

Self-Determination Theory received a lot of attention in the HCI subfield of game design and gamification research [TM20], but received little attention from other subfields traditionally involved in usability and design, despite the recent availability of SDT-based questionnaires [Brü+18] and suggestions that it could help better understand the influence of input methods on users' experience with interactive products beyond games [Wat+13].

In the case of signifier-less design, I believe the SDT could help understand some aspects of the design. For instance, the needs for competency, autonomy and relatedness could cause extrinsic motivations such as: (competence) searching hidden widgets because they are efficient and will allow the user to become more competent, (autonomy) to find new ways to use the device in users' lives that better

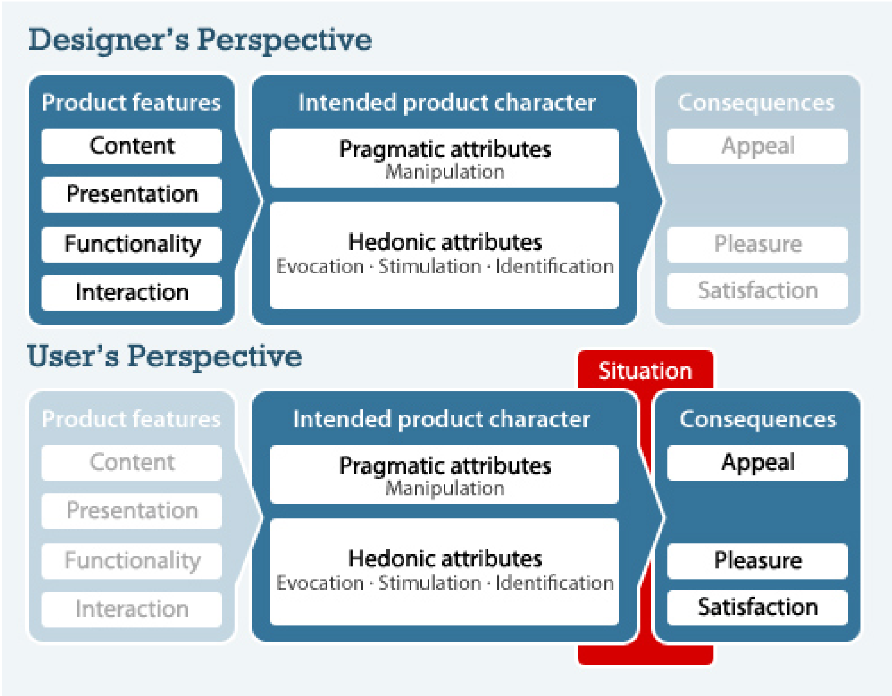


Figure 7.3: Hassenzahl's model of User Experience (UX), from [Has03].

match what they expect for their life (e.g., saving time by reading my e-mail in commutation with an interaction that is easy to perform in a crowded train), or (relatedness) be able to talk about signifier-less designs with others.

7.4.3 *Hassenzahl's model of UX*

Hassenzahl presented a model of the User Experience (UX) of a product from both designer's and user's perspective [Has03]. The model is based on the assumption that a product has some *features* that convey a particular *product character* to users, which leads to *consequences* on the user's emotions and behaviors (see Figure 7.3). Additionally, the model distinguishes between the designer's perspective and the user's perspective. From the point of view of the designer, she chooses and combines the product's features with the goal of creating an *intended character*. From the point of view of the user, she perceives the features of the product and construct her own *apparent product character* and reacts to it.[5] The *consequences* on the user are not always the same and depend on the actual *situation*, and can consists in a judgment about the product's appeal (e.g., "It is good/bad"), emotional conse-

5 The difference between the intended and apparent product characters is similar to Norman's distinction between the designer's mental model and the user's one [Nor88], although the notions of product character and mental model are quite different.

quences (e.g., pleasure, satisfaction) and behavioral consequences (e.g., increased time spent with the product).

The features of a product are: what it contains (*Content*), what it looks like (*Presentation*), how it works (*Functionality*), and how the interaction is suggested (*Interaction*).

These four types of features give the product its own character: a high-level description of the product, which function is to reduce cognitive complexity and to trigger particular strategies for handling the product. This character summarizes the attributes of the product, such as “novel”, “interesting”, “useful”, or “predictable”. Hassenzahl additionally distinguishes *hedonic* attributes from *pragmatic* ones [DH08].

Pragmatic attributes concern the practical aspect of the product, the achievement of behavioral goals: tasks such as checking messages, making a call, etc. Examples of pragmatic attributes are: “simple”, “clear”, “controllable”. They are always instrumental in the sense that they concern goals that are not an end in itself. As such they have been the main focus of usability-centered approaches to design and HCI research.

Hedonic attributes concern the different sensations that the product can bring to users and relate to users’ self – "Why I use this product?". Hedonic attributes, e.g., “innovative” or “exciting”, signal the fulfillment of superordinate, self-referential goals like personal growth or self-expression. There are three categories of hedonic attributes in Hassenzahl’s model: *evocation* concerns the memories that a product can bring to users; *stimulation* concerns a product’s ability to address the basic human needs of self improvement, novelty, and challenge; and finally, *identification* addresses the human need to express one’s self through products one owns.

A longitudinal study of the experience of users from one week before to four week after the purchase of an iPhone [Kar+09] revealed that the product attributes that provided positive initial experiences were not as crucial for motivating prolonged use. According to the authors of this study, “product adoption contained three distinct phases: an initial orientation to the product dominated by the qualities of stimulation and learnability, a subsequent incorporation of the product in daily routines where usefulness and long-term usability became more important, and finally, a phase of increased identification with the product as it participated in users’ personal and social experiences”.

The *stimulation* dimension of the model could be very relevant to describe the benefits of signifier-less interaction techniques. Indeed, the discovery of such techniques is a very stimulating experience, which may motivate users to explore the interface in hope of such stimulating experiences. It is also possible that the level of knowledge and skills required to discover signifier-less designs match the user’s entry in the second phase of UX evolution, where the product get incorporated in daily routines and its usefulness and long-term usability become relatively more important, and its learnability has become relatively less important.

7.4.4 *Non-motivational causes*

In addition to the motivational aspects of users' improvement of their knowledge and skills that I described above, I must mention the possibility of non-motivated causes for such improvements. In particular, improvements can result from *accidental* (i.e., unexpected) discoveries, or can be *incidental*, resulting from a practice of skills through rehearsal although the user's goal is not to become better (as it would be if the user engaged in some form of training).

7.5 INTERACTION BETWEEN CURRENT KNOWLEDGE AND MOTIVATION

7.5.1 *Knowledge motivated by performance*

POWER LAW OF PRACTICE A phenomenon called the *power law of practice* has been repeatedly observed when the performance of individual people involved in activities is aggregated over the group studied and plotted against time: it shows a smooth and continuous power function that starts quickly (broadly corresponding to the cognitive phase) before progressively reaching a plateau of maximal performance (corresponding to the autonomous phase) – although not all participants reach the maximal performance level and may require special training to do so.

INTRAMODAL PERFORMANCE IMPROVEMENTS Cockburn *et al.* use this law as a model of *intramodal performance improvement*, i.e., the performance improvement with one particular interactive method (such as pointing with the mouse) and one particular function (such as selecting the bold function in a text editor) [Coc+14]. In terms of domains of the knowledge and skills (see Section 7.2.1), it corresponds broadly to the improvement of specific UI-level knowledge and skills as support for specific command-level knowledge and skills.

INTERMODAL PERFORMANCE IMPROVEMENTS The notion of intramodal performance improvement is contrasted in Cockburn *et al.*'s work with *intermodal performance improvement*, *vocabulary extension*, and *task mapping at the interface* [Coc+14]. Intermodal performance improvement concerns the performance gain made possible by using a different interaction method to access a same function, when this alternative interaction method has a higher performance ceiling (i.e., the height of the plateau of maximal performance in a power law of practice model). This notion considers performance improvements as the only motivation for using a different method although there can be other types of motivation for doing so, as we have seen in Section 7.4 (for instance, it could be motivated by a need for competence with an argument like "expert users know how to use this feature, I want to be an expert user, so I have to learn it too", which is more a question of status than of actual performance).

Intermodal performance improvements can then be described in terms of domains of knowledge and skills as follows: first, to realize that a function can be accessed differently, there must be an improvement of the knowledge and skills for this function at the command or task level (depending on what is called a "function" of the interface); second, improvements of the knowledge and skills involved in the new interaction method must be made at the UI level to support the improvement of knowledge and skills for the function considered at the command or task level, as for intramodal performance. However, the first of these two steps is not always followed by the second. Conckburn *et al.* indeed consider two reasons why it could be the case: one is related to the first step, and the other is related to an opposition between short-term and long-term benefits.

SATISFICING AND COST-BENEFIT ANALYSIS The first reason for not switching to the new technique is an inaccurate perception of the costs and benefits of doing so, leading to a belief that it is not worthwhile. This is a problem that arises because the performance improvement is perceived as an extrinsic motivation for the learning, and might be alleviated if it was perceived as an intrinsic motivation. Users likely have an inaccurate perception of the costs and benefits of switching to the new technique, because they just learned about it and do not yet have a model of its operation that is clear enough to make such evaluations. They can thus easily overestimate the difficulty of learning the technique or underestimate its performance relatively to the "old" technique (which can also happen because she overestimates her current performance with the "old" technique). All three cases have been observed in studies (see section 5.1.2 of [Coc+14] for a review).

Investigating more deeply the question of the performance of a newly discovered technique could help users better evaluate the benefits of this technique in terms of performance (or according to other criteria than performance), but to engage in such an investigation, they must first believe that it is worth the time and effort. As a consequence, users rarely engage in such rigorous evaluations: they prefer faster, less demanding, but also less accurate evaluations of the costs and benefits of switching to the new technique, so that they can take a non-optimal but "satisficing" decision.

PERFORMANCE DIP The second reason to not switch to the new technique is a fear of a *performance dip:* users may understand that in the long-term, switching to the new technique may improve their performance, but in the short term, they will have to use the new technique in a way that is less efficient than the level of performance they can reach with the "old" technique. Users may then believe that now is not an appropriate time to learn the technique because they need to maintain their current level of performance. This phenomenon explains the interest of the HCI community for designs like Kurtenbach's marking menus (see Section 7.3.2) and how their implementation of the principle of *rehearsal* allows to reduce or even cancel completely the difference of performance between the two input methods, without relying on an external motivation of the user for doing so.

VOCABULARY EXTENSION Cockburn *et al.*'s notion of *vocabulary extension* concerns improvements of the overall knowledge of an interface in terms of the set of commands it provides. However, despite listing works that provide users with recommendations of commands to learn, and mentioning gamified approaches to learning, Cockburn *et al.* do not mention explicitly the problem of prioritizing learning that I will discuss in Section 7.5.2. Instead, they see vocabulary extension mainly as similar to intermodal extension in that "users need to be aware of commands before using them, and they need to understand the magnitude of improvement that they will enable".

TASK MAPPING Finally, the last of the four domains of interface performance improvement discussed by Cockburn *et al.* is *task mapping*. It "addresses higher level issues of the strategies that users adopt when seeking to complete their tasks with a UI. It concerns the coordination of functions to complete a task" [Coc+14]. In terms of the domains of knowledge and skills, task mapping thus concerns the improvement of user's knowledge and skills at the task, appropriateness of workflow, or domain object levels (depending on what exactly is meat by "complete a task"), and may or may not be supported by improvements of knowledge and skills at lower levels.

What these three levels have in common in that they involve entities (tasks, workflows, domain knowledge) that are not features of the interface. At such, the knowledge and skills involved at these levels are often user self-defined strategies, and the problem is that coming up with a strategy is a slightly different process than learning about something that pre-exists as an affordance in the interface. Adopting strategies goes through the same steps of 1) getting aware of a possibility, 2) evaluating its benefits and the costs of switching to it, and then 3) doing the switch, that we discussed for intermodal performance improvements. The difference is that, like for signifier-less designs, there is often no design means in the interface to expose a possible strategy to users. As a consequence, Cockburn *et al.* focus on design means to expose possible strategies. They also mention another approach suggested by Bhavnani and John [BJ00], which is to design interfaces that allow users exploiting efficient strategies that generalize across a variety of application domains (which would set them as belonging to the General Knowledge and skills level), such as the Detail-Aggregate-Manipulate strategy used in the drawing application example used to analyze the Appropriateness of Workflows level in section 7.2.1. In addition to raising user awareness of the strategies, this approach additionally help them evaluating the costs and benefits of adopting them, as they already have a clear understanding of the strategies.

7.5.2 *The Knowledge-motivation loop*

Because motivation is required to learn, but what one knows affects one's ability to foresee the consequences of engaging in some learning – as we have just seen for performance-motivated learning – complex behaviors can emerge from the interaction loop between current knowledge and skills and motivations (the right part of Figure 7.1). In this section I address some aspects of this phenomenon.

COMPETITION FOR LEARNING A third reason that could be provided to explain why intermodal performance improvements do not happen is by considering that a user may have many new techniques to learn and that these techniques are competing for the user's learning efforts (in addition to the non-learning tasks that motivate the use of the interface in the first place). As a consequence, users may have to prioritize the things they want to learn about the interface, and a particular interaction technique may never been learned simply because learning it never becomes a priority. Since this priority level is a part of what the user "knows" about the technique, it is also possible that a particular technique has been categorized early as "advanced, low-priority stuff", and that users stick to this classification later as they do not feel the need to update it.

RELEVANCE OF THE KNOWLEDGE AND SKILLS Some knowledge and skills may only become relevant for users after having understood how to exploit them at a higher level. For instance, users of an e-mail application who read each mail individually before deciding to delete it or reply it (see Section 7.2.1) may dismiss the deletion widgets in the thread list as not useful, only because they do not see the point of using them. It is the same thing for people who prefer to format their text with the bold/italic/underlined buttons rather than using styles: all the style-related commands would seem useless to them and they would not try to learn them.

Conversely, a user can discover these features and realize their potential to change the way they think about the activity supported by the application. In this case, the knowledge of these commands become relevant and can motivate users to learn them as well as the higher-level workflow and domain knowledge.

MOTIVATION FOR EXPLORATION Knowledge can also flow from an upper level to a lower one. Typically, a user's knowledge of an activity can lead her to explore the interface looking for a feature that she judges necessary for this activity, even if this feature is actually implemented with a non-signified design. This is typically the case when one looks for the power switch of a device, and other forms of *puzzling signifier-less designs* (see Section 6.3).

On the other hand, when users lack domain knowledge, they might rely on exploration of the interface in hope of acquiring such knowledge, as suggests Kirsh in his analysis of using an unknown image manipulation application to enhance the quality of a figure to be included in a research paper [Kir97] (see Section 4.2.3).

7.6 FIRST CONTRIBUTION: DEGREE OF KNOWLEDGE

Computer interfaces have become more and more complex to use nowadays, as some applications are designed to cover a very wide range of activities. As a consequence, on one side there can be many features of an application that are targeting activities that a user may not be interested in an do not need to know, and on the other side, there can be many different ways to interact with the application to achieve a same operation (for instance, to quit the application, users can use icons, hotkeys, or drop-down menu items).

It is then not straightforward to distinguish users as novices or experts of an application, because they may have good knowledge of the interaction possibilities in one specific area of activity but not in another area. For example, Photoshop users who know all the commands to edit photos, with the most efficient ways to trigger them, but know nothing about how to do animations in Photoshop: in some professional contexts they would be called "experts" but not in others.

Knowledge and skills about the interaction possibilities of an interface may change over time and evolve according to users' intentions, motivations, and life goals. But so do users' expectations about what they want to use an application (or system) for. More important than expertise levels, we need a measure of how close the knowledge and skills of a user are from those required to use the application "at its best". Such a measure can however not been defined, because it is too vague: how could we define what is the best way to use an application for specific users, when even them may not have a clear idea about it? But we can define measures of the knowledge and skills of users

Used

Known

DoK	Primary	Productive	Receptive	Unknown
1. Is the user aware of the existence and the basic operation of the technique? 2. Can the user recognise that the interaction technique provides a way to accomplish a task when asked?	✓	✓	✓	
1. Can the user perform the technique to accomplish a task if asked to do so? 2. Can the user think about the interaction technique to do the task by himself/herself?	✓	✓		
Does the user systematically use this technique in preference to others, or include the technique in usage strategies?	✓			

Figure 7.4: Practical definition of the *degrees of knowledge* in terms of how users can behave relatively to an interaction technique in different situations.

relatively to specific interaction possibilities, and see if there is room for improvement.

I propose *degrees of knowledge* (or DoK) as such a measure. Instead of trivially categorizing interaction techniques as "not known" or "known", DoKs acknowledge a continuum of knowledge and skills that provide users with different abilities to understand interaction techniques, recognize them, perform them, or include them in higher-level usage strategies and habits.

These interaction possibilities exist in terms of menu, keyboard shortcut, icon buttons, or even mouse clicks, etc. By reviewing the degrees of knowledge at which a user stands for a set of interaction techniques, we may see different patterns of how users' strategy of their interaction vocabulary / current knowledge, it may inform designers about how users deal with the interaction techniques, or about which interaction technique "works" for which users.

I thus propose to define the *degree of knowledge* of a user concerning an *interaction technique* with the following simple ordinal scale:

1. UNKNOWN, when the user is not aware of the existence of the interaction technique, and thus cannot recognize it nor perform it.

2. RECEPTIVE, when the user is aware of the interaction technique and can recognize it if performed in front of her, but cannot use it by herself.

3. PRODUCTIVE, when the user can use the interaction technique by herself, when asked to perform a task that can be accomplished with this interaction technique.

4. PRIMARY, when the user spontaneously prefers using the interaction technique over others when prompted to perform a task that can be accomplished with this interaction technique, or has built strategies about when to use this technique.

As seen in Figure 7.4, the degrees of knowledge can be identified experimentally by investigating some simple abilities of users – although the model is voluntarily simplified and merges some aspects of interaction that could be considered as separate degrees in more complex models: this limitation comes from the constraint of defining a practical measure that can be performed during experiments without spending to much efforts on trying to identify distinctions that may not be relevant.

As they get familiar with an interaction technique, users reach higher degrees of knowledge for this technique, by developing the abilities listed in Figure 7.4 for each degree. Conversely, if they do not use the technique they can forget about it and fall back to lower degrees of knowledge. To ease the discussion about users' skills, I also define USED for interaction techniques that can be used by the user independently of her strategies (i.e., either PRIMARY or PRODUCTIVE); and KNOWN for interaction techniques that the user can recognize, independently of their ability to perform it (i.e., any DoK but UNKNOWN).

7.7 SECOND CONTRIBUTION: SOURCES OF KNOWLEDGE

If designers hide some interaction techniques, they take the risk that users may not be able to find them at all; in this case, knowing the sources of users' knowledge of these techniques may provide insights for design improvements. An important concept for discussing signifier-less designs is thus the *sources of knowledge*, or SoK in short.[6] I propose two dimensions to categorize sources of knowledge: *distance* and *intentionality*, which are not necessarily orthogonal.

7.7.1 *Distance of the source of knowledge*

A first dimension to describe the sources of users' knowledge about the interface is how remote they are from the place and time this

6 Despite the proximity of the two acronyms DoK and SoK, used here for practical reasons, I invite the reader to pay attention to the fact that Degrees of Knowledge and Sources of Knowledge are very different concepts: the former is a definition of a way to quantify knowledge, while the latter is more a focus of interest that does not translate directly into a single quantification.

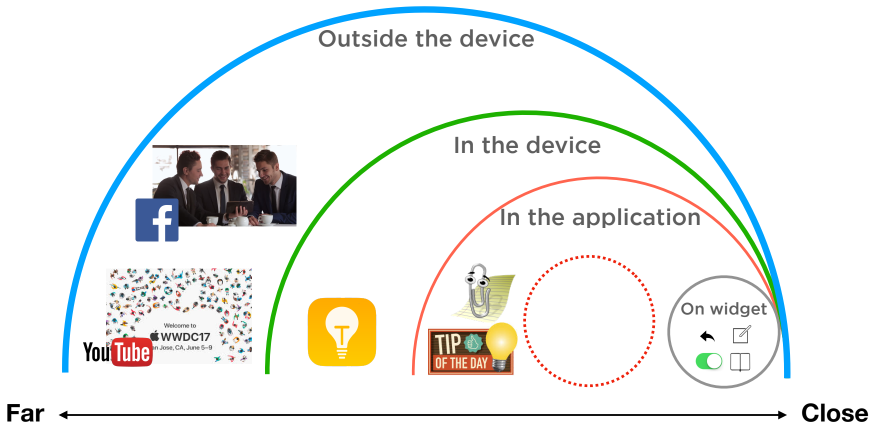


Figure 7.5: Distance between a source of knowledge and the usage of this knowledge in interaction.

knowledge is applied as users interact with the interface. There is an ill-defined but intuitive notion of distance between a source of knowledge and its application, which is depicted in Figure 7.5.

The ideal situation for users when interacting with an application is that they can get aware of its important functions by just looking at the interface, using perceived affordances, signifiers, and metaphors. When users see an icon of a trash bin, they may instantly obtain the knowledge that the function of this icon is to delete or to categorize some data as no longer useful: the interface itself is the source of this knowledge.

This is the shortest possible distance, which is epitomized by SoKs that are classical widgets and direct signifiers providing affordances and feed-forward. In this case, user's knowledge is gained by directly looking at the SoK and it happens at the same place and almost at the same time this knowledge will be used. The distance of the SoK is thus almost null, or very close to users. Sometimes, interaction possibilities are discovered by accidentally triggering them, or by explicitly trying them, which could be considered as the shortest possible distance – although it is unlikely that designers can rely on such SoKs as there is no certitude that users will indeed use such sources.

There are also SoKs that are more remote from the places the knowledge is needed, but still in the application. Examples would be how menus can provide hints for hotkeys, or Microsoft Word's *Clippy*. At the farthest, but still in the application, information boxes such as the *tips of the day* displayed at application's load time can convey some knowledge to users – although it is also usually possible to deactivate them. These sources being provided "on the way" leading to the place where the information they provide is useful (and sometimes, "in the way"), they can be considered as closer than the help pages provided by an application, because the later are only accessed explicitly by users and require them to interrupt their activity to perform a search of the relevant page, and are less likely to be accessed.

Knowledge of an application can also be communicated outside the application but still inside the device that hosts it. For instance, with Google's Drippler or Apple's Tip, which are applications dedicated to communicating useful knowledge about other applications shipped with the system. This intermediary distance is shown in the middle of Figure 7.5.

The largest distance for a SoK is out of the device itself. A frequent case, for example, is that we learn about an interaction possibility from a colleague during the coffee break, or when searching for video tutorials on the web.[7] The knowledge may also come from other external sources, such as news, social networking applications and websites, demonstrations in store, chatting between friends, or even from a glance at a stranger using her phone beside you in the subway train.

A simple categorization of distances thus considers whether the SoK is the widget or domain object itself, the application, the operating system / device, or the social environment of the user (mass media or word of mouth).

7.7.2 *Intentionality of the discovery*

There are three different situations / scenarios for users to acquaint knowledge of an interaction possibility and get in contact with the source of this knowledge, depending on the intentionality of the discovery:[8]

Through explicit research: users have a precise goal or problem to solve in using the application, and they search for information about it. This search can be performed in the interface itself (targeted exploration, use of help features provided by the application) or from more distant sources (web search, asking a friend, etc.)

Through random exploration and information: users may just want to explore the interface, led by curiosity. They may expect to find valuable

7 This includes the case where the tutorial video is watched on the device that is concerned by the tutorial: in that case, what matters is that the video, as a SoK, is external to the device, even if it is accessed through the device.

8 I have described earlier some aspects of this dimension as belonging to the motivational aspects of the model proposed in this chapter, which is true as long as the distinction between design means and motivations rely on a distinction between what belongs to the interface or user. The notion of source of knowledge transcends this distinction, as it is as much a feature of the world than a particular event involving the user. A benefit of moving the aspects of intention out of the domain of motivations is also that it removes the need to discuss non-motivational aspect as a subset of motivational aspects: the notion of motivation becomes more focused in scope.

information although they are not targeting a specific interaction need.[9]

Without intentions: the knowledge has been communicated to the user but it was not motivated by the user. For instance, knowledge can be *suggested* or *imposed* to users when they are shown a device in a physical or online store before buying it. Knowledge can also be deduced from habitual know-hows acquainted during past interaction with similar devices, systems, or applications. And finally, knowledge can also be discovered by accidentally triggering the function.

7.7.3 *Sources of Knowledge as an extension of Design Means*

The notion of a SoK can be understood as an extension of the concept of *design means* of conveying usage knowledge, such as signifiers, metaphors and constraints. It acknowledges that learning about an interface is an experience that transcends the pragmatical resolution of precise interaction problems, unlike what would suggest a model such as the Seven Stages of Action (see Section 4.2.3). The notion of a SoK also acknowledges that this experience of learning about the interface can be influenced by designers not only by the design of a product, but also by designing *around* the product (e.g., how the product is presented to customers). This understanding of design is already at the heart of design practices such as User Experience (UX) Design, and can be found in scientific approaches that take interaction as a *situated* phenomenon such as Beaudouin-Lafon's call for focusing on "interactions, not interfaces" [Bea04].

Introducing the notion of a SoK has the benefit of highlighting an important concern for designers – providing information about how to use the system – in a way that do not focus on a specific design solution (as would be terms such as signifiers or feedforward) but invite to consider all possible solutions in the design space. It also allows designers to question how their designs support behaviors like accidentally discovering an interaction technique or sharing the experience of such a discovery with relatives.

7.8 conclusion

In this chapter, I have proposed a model of interaction technique proficiency improvement based on three types of prerequisites: a user's current knowledge and skills concerning the use of the interface and the activity it supports, a user's motivations and non-motivational

9 The distinction between explicit research and exploration led by curiosity is not always easy to make. For instance, when a user discovers that swiping leftward an e-mail in the iOS Mail application reveals *swhidgets*, she can wonder if swiping rightward also does something. In that case, the exploration is not random but targeted, yet users are not motivated by solving a specific problem.

causes for improving the proficiency, and design means, the features of an interface that can help users improve their proficiency. Current knowledge and skills is the central element of this model, as it includes not only declarative knowledge about the interface and activity but also procedural knowledge about how to act, which affects directly the proficiency in an interaction technique depending on the stage at which the technique has been acquired (cognitive, associative, or autonomous). Design means and motivations can indeed be defined respectively as aspects of the interface and user's mind that are required to improve knowledge and skills.

The model however also reveals a more complex dynamic caused by the interactions between its three components, shown in Figure 7.1: the current knowledge and skills affects the efficiency of the two other components in that 1) knowledge is required to understand some design means but skilled behavior reduces the need for some design means, 2) the strength of motivations, especially extrinsic ones, is affected by users' knowledge that limits users' ability to clearly understand the consequences of their adoption of new usage patterns. As a consequence, complex behaviors emerge from the two dependency loops interconnected through the current knowledge and skills. I described shortly some of these complex behaviors, but the topic is too vast to provide a complete account of them in this dissertation.

In the last two sections, I have presented two additional contributions to the model. The first, the concept of degree of knowledge (DoK) is a measure of a user's proficiency in a technique based on four ordinal categories that attempt to capture the complexity of proficiency dynamics in a simple and low-detail model. This measure has been created to be easily integrated into experiments implying multiple interaction techniques – such as those that I have conducted on *swhidgets* which will be presented in Chapter 8.

Finally, the concept of sources of knowledge (SoK) has been introduced as an answer to the problematic of signifier-less designs, to account for the possibility of their discovery despite their lack of signifiers. As such, it provides a partial account for differences between the learnability of an interface and its actual knowledge by a population of users. It generalizes the notion of design means by acknowledging that the discovery can happen outside the interface at various levels of "distance" from the actual usage of the technique, and at various levels of intentionality. It can also be used in experiments as we will see in Chapter 8.

Part IV

STUDIES

8 TWO STUDIES ON USERS' KNOWLEDGE AND RECEPTION OF IOS SWHIDGETS

The state of the art of signifier-less designs that I presented in Chapter 6 and the analysis of *swhidgets* that I presented in Chapter 2 revealed that little is known regarding whether users rely on these controls or not, and how they discover the *swhidgets*. In that purpose, we conducted a first controlled experiment to observe which modalities 28 participants – all iPhone users – used when prompted to perform a set of operations, and had them filled out 2 questionnaires to learn the sources of their knowledge of *swhidgets*. Eager to know if this result can apply on general *swhidgets* usage, we tried to reach more participant by putting an online study with more detailed questions on how they feel about using *swhidgets*. The results of this online study provide even more interesting data about participants' appreciation of *swhidgets*.

In this chapter, I will present the rationale and objectives of the two experiments, detail their designs, analyze their results, and I will finally discuss and compare the results from these two experiments.[1]

8.1 THE LABORATORY AND ONLINE SURVEYS

8.1.1 *Rationale and objectives*

In order to analyze the "means" and "motivations" allowing users to use *swhidgets* to achieve tasks, it is necessary to find first how much users know about *swhidgets*. Therefore, my first goal in designing an experiment was to know what users currently do to solve their daily tasks with applications that contain *swhidgets*.

The first experiment I have conducted thus aims at providing answers to the research questions on *swhidgets* described in Section 2.7 by observing the participants and interacting directly with them. It took the form of a laboratory study in order to better understand how users interact with *swhidgets*. This study aims at investigating which *swhidgets* users are aware of and whether they would spontaneously use them when prompted to perform operations that can be completed by using *swhidgets*. The goals of this experiment are more specifically:

1. to observe if participants know about *swhidgets*;

2. to find if participants discover *swhidgets* by themselves or more often by another source;

1 This chapter has been partially published as a paper for the ISS'19 conference [PM19].

3. to have an idea of the possible sources of knowledge from which participants generally develop their interaction vocabulary;

4. to see if participants know *swhidgets* better when they are "tech-friendly" persons;

5. to know if participants think that *swhidgets* make their life easier/more convenient;

6. to learn what participants expect from *swhidgets*.

So the research questions that I try to answer with the laboratory study were first operationalized with the following questions:

1. How many interaction techniques do users know to achieve a task?

2. What is the order/strategy when users know more than one way?

3. Where do users learn their interaction techniques?

4. How many users know about *swhidgets*?

5. Do *swhidgets* remain the favorite way for users when they know about it?

6. Do users think these *swhidgets* improve their efficiency in smart phone tasks?

After the first laboratory experiment, I had collected data that drove me to plan the second experiment online. As an online questionnaire is a better way to reach more participants, the goals of this second experiment were to:

1. have more diverse professions of participants (rather than separate them into IT and non-IT group);

2. know more precisely how the usage frequency of an application or interaction technique affects the adoption of *swhidgets*;

3. know the exact source of knowledge if participants are aware of a *swhidget*;

4. know if participants find *swhidgets* difficult to discover;

5. know if participants are willing to adopt *swhidgets* if they don't know about them;

6. know if participants will rank *swhidget* as their favorite way to achieve the task;

7. know if participants build a logic of finding *swhidgets* in the application or in other applications;

8. know if participants are capable of transferring the knowledge of using *swhidgets*;

The expected insights of these surveys belong to different categories:

- Users' original interaction vocabularies of the system.
- From where they learn these interaction vocabularies.
- Do users keep using these *swhidgets* after having acquired this knowledge.

8.1.2 *Scope limitation*

These surveys have a limited scope and focus on the iOS operating system and its most used default applications. I introduced this scope limitation to:

1. Maximize the odds that participants had a chance to use and know the *swhidgets* in these applications before the study, so that the study indeed evaluates participants' current knowledge rather than their ability to learn the interface during the experiment.
2. Ensure the set of *swhidgets* studied is coherent and stable across application versions and device models, so that this evaluation of participants' knowledge applies to a large population of iPhone users and is not biased by participants' experience with a specific device.[2]
3. Maximize the coherence of the set of *swhidgets* studied: Since I am interested in the possibility of knowledge transfers across applications, I need to consider applications that follow the same guidelines and share a common logic, which is more likely with applications from a same developer.

I acknowledge that *Swhidgets* also exist in other touch-based operating systems such as Android and Windows Mobile, but I believe that iOS offers a consistent interaction environment with a variety of hidden controls in it that makes it well adapted for first studies on that topic.

---

2 The laboratory study ran from December 2017 to January 2018, and the online study ran in May-June 2019. The iPhone X was released between the two studies and introduced important changes in the iOS interface: it removed the physical main button below the screen, added a physical separation between the left and right halves of the top screen bezel, and changed the activation gesture for the two system *swhidgets*, as the Notification Center and Control Panel that were previously mapped respectively to the top and bottom of the screen became mapped respectively to the left and right parts of the top bezel. The action of pressing the physical button was replaced with a swipe across the bottom bezel, a gesture previously mapped to the Control Panel, which can be seen as a third system *swhidget* for the iPhone X.

8.1.3 *Applications and operations studied*

The two studies involve the same four "applications" from which 14 operations have been defined in total (see Table 8.1). Since each study uses a different subset of operations but some of them are common to both studies, I describe the four applications and all their respective operations in the next paragraphs, independently of the study in which they are used.

CONTROL PANEL OPERATIONS Although not technically an application, I group four operations in this category, as they all use the system-wide feature that is the Control Panel. The first operation, *revealCP*, allows to assess if participants know how to reveal the Control Panel by doing a upward swipe across the bottom bezel of the screen. By asking the participants to complete the three other operations, we can assess their knowledge of the features provided by the Control Panel. These operations are: turning on/off the Wi-Fi, turning on/off the flash light, and (un)locking the screen orientation in portrait mode. For *wifi* and *flashlight*, participants can also use a *Voice control* by asking Siri to complete the operation; and for *wifi*, they can also use *Navigation Controls* in the "Settings" application.

NOTIFICATION CENTER OPERATIONS Although not technically an application, I group three operations in this category, as they all use the system-wide feature that is the Notification Center. The first operation, *revealNC*, allows to assess if participants know how to reveal the Notification Center by doing a downward swipe across the top bezel of the screen. By asking the participants to complete the two other operations, we can assess their knowledge of the features provided by the Notification Center. These operations were checking the current weather and responding to missed notifications. The latter can be performed explicitly by touching the notifications listed in the Notification Center, or implicitly by opening the application associated to the notification and using *Navigation Controls* in this application to access the content notified. Participants could get the *weather* information with *Swhidgets* by reading it in the Information Panel, with *Navigation Controls* by opening the Weather application, or with *Voice Controls* by asking Siri. The Information Panel is a sub-panel of the Notification Center that can be accessed with a rightward swipe in the Notification Center, on the lock screen, or on the main panel of the Home Screen.[3]

---

3 Technically, this rightward swipe is not a *swhidget* but rather a gesture to navigate between panels. There is indeed a very subtle signifier for this gesture at the bottom of the screen, in the form of small dots – one for each panel, the current one being brighter.

Applications	Operations	Studies		Input Methods			Instructions
		lab	web	Swhidgets	Navigation Controls	Voice Controls	
Control Panel	revealCP		■	system	Ø		Do you know how to reveal the Control Panel on your touch-based device?
	flashlight		■	system	Ø	✓	Do you know how to turn on/off the Flashlight on your iOS touch-based device?
	lockPortrait		■	system	Ø		Do you know how to lock the orientation of the screen on your touch-based device?
	wifi	■	■	system	*Settings* application	✓	(lab) Can you turn off the Wi-Fi? (web) Do you know how to enable/disable Wifi on your iOS touch-based device?
Notification Center	revealNC		■	system	Ø		Do you know how to reveal the Notification Center on your touch-based device?
	weather	■		system	*Weather* application	✓	Can you tell me what is the current temperature outside?
	notification		■	system	directly open apps	✓	Do you know how to respond to the notifications that you missed on your touch-based device?
Messages	SMS.delete	■	■	item	edit mode (long press)		(lab) Can you delete this message, please? (web) Do you know how to delete a conversation from a specific person in the Message application?
	SMS.search	■	■	view	scroll to find	✓	(lab) Can you search the SMS chat with John? (web) Do you know how to search messages from a specific person in the Message application?
	SMS.time	■	■	view	read time		(lab) Can you tell me at what time this message was received? (web) Do you know how to see the timing of messages in a conversation in the Message application?
Mail	Mail.delete	■	■	item (+long)	edit mode (*edit* button)		(lab) Can you delete this e-mail? (web) Do you know how to delete e-mails in iOS Mail?
	Mail.search	■	■	view	scroll to find	✓	(lab) Can you search the e-mails from John? (web) Do you know how to search an e-mail on your touch-based device?
	Mail.markAs	■	■	item (+long)	*flag* button		(lab) Can you mark this e-mail as read? (web) Do you know how to mark an e-mail as read/unread in iOS Mail?
	Mail.reply	■		item	*reply* button		Can you reply this e-mail?

Table 8.1: List of the possible input methods to complete the operations that participants were asked to perform during the experiments. Depending on the operation, input methods are either a *Swhidget*, a *Navigation control*, or a *Voice control* (Ø indicates an input method that is not available). Tasks in blue and orange were respectively used in the laboratory study and online study.

MESSAGES OPERATIONS Three operations concern the Messages application, shipped with iOS to provide access to many messaging services, most notably SMS.

The first operation consists in deleting the entire conversation with someone. I included this operation in the studies because it represents a risk of data loss and its design might thus have benefited from special attention, and also because it is likely that many users never or rarely perform it. The operation can be done with an *item swhidget* by swiping leftward on the appropriate item in the conversation list to reveal buttons, then pressing the Delete button and finally confirming the deletion. Participants can also use *Navigation Controls* by performing a long press on any message in the conversation view, which displays a menu, then selecting the "More. . . " option in this menu to enter the edition mode, and finally press the "Delete All" button on the top-left of the screen.[4]

The second operation consists in searching for the messages sent by a person identified by her name. This is an operation that presents no risk and might be performed more or less frequently depending on the users' chat habits and number of contacts. Participants can use a *view swhidget* by performing a downward swipe to reveal a search bar at the top of the discussion list, they can use *Navigation Controls* by scrolling up or down the conversation list until they find the one searched, or they can use *Voice Controls* by asking Siri.

The third operation consists in looking for the time a specific message was sent. Participants could only get the exact information by using a *view swhidget*, swiping leftward from anywhere in the conversation view in order to reveal the information. Alternatively, they could get an approximate timing information with *Navigation Controls* by scrolling the conversation until they find the nearest time information displayed in the view between messages. Such timestamps are inserted in the discussion above a message only when the previous message was exchanged more than one hour earlier.

MAIL OPERATIONS Four operations concern the Mail application provided with iOS as the default application to read and send emails. I have included the deletion and search of mails so that comparisons between the two applications can be performed. I also included the two operations consisting in marking e-mails as read and replying to emails, as they use variants of a similar *swhidget* and differ by their frequency of use and users' way of managing e-mails.

Deleting e-mails can be performed with *Swhidgets* by performing a short leftward swipe on a mail in the thread list, then selecting the Delete button that it revealed. Interestingly, an "expert" version of this

4 There is actually a third way to do it, which is to press the "Edit" button on the top left of the conversation list, then select the conversation(s) to delete, and finally press the "Delete" button on the bottom-right corner of the screen. But nobody used it in the lab study.

method exists, consisting in performing a longer swipe, which will directly delete the e-mail. Participants could also use two *Navigation Controls*, as they could press the trashcan icons in the mail view or thread view, or they could enter the edition mode in the thread list by pressing the Edit button in the top-right corner, selecting the threads to delete, and finally press the Delete button in the bottom-right corner of the screen.

Searching emails can be performed with *Swhidgets*, as it is again possible to use the search bar that hides under the upper navigation bar, and that participant can reveal with a downward swipe in the thread list. It can also be performed with *Navigation Controls* by scrolling the thread list and opening threads until the searched one is found, or with *Voice Controls* by asking Siri.

Marking e-mails as (un)read can be done with a *Swhidget* by swiping rightward on a thread (in the thread list) or on a mail (in the thread view) to reveal an unique button representing a closed or open mail. Like for deleting mails, an expert mode exists for this operation with a longer swipe. Alternatively, participants could use *Navigation Controls*, by directly opening a thread view and scrolling – which marks mails as read as they appear into view, or by pressing the flag icon in the bottom-left of the thread view to open a menu containing a "Mark as Unread" option.

Finally, replying an email could be performed with *Swhidgets* by performing a short leftward swipe on a mail in the thread view (but not in the thread list), to reveal a blue button that give access to a menu containing a "Reply" option. Here, the alternative *Navigation Controls* seem more practical, as there is a leftward arrow icon at the bottom of the thread view and mail view that can be touched to reply to the mail (or whole thread).

8.2 LABORATORY STUDY

The laboratory study focuses on direct observation of and interaction with 28 participants, including notably phases in which we observe how they interact with a device in response to a request, and debriefing interviews.

8.2.1 *Factorial design of the experimental tasks*

The study follows a within-subject factorial design with two independent variables and two dependent variables. The two independent variables are the *input method* (three levels: *Swhidgets*, *navigation controls*, and *voice controls*) and the operation (9 levels, see Table 8.1). The two dependent variables are *primary-or-uncompleted* and *degree of knowledge* (their levels will be described in Section 8.2.6). The factorial design is complete if we ignore the voice controls: each participant faces exactly one *swhidget* and one *navigation control* for each level of *operation*, but also one *voice control* in some operations.

8.2.2 *Experimental tasks*

Participants are invited to accomplish different "operations" designed to test their knowledge of four "applications" (see Table 8.1): the control panel, the information panel, and hidden commands in the text messaging and email applications. All operations can be achieved with various *swhidgets* triggered by swipes, but other interaction possibilities are also available for the participants. For instance, participants can invoke Siri or use visible buttons in the navigation bar, or 3D-touch some interface elements. The four applications are divided in operations, which will be described in the next paragraphs.

By running through the operations of all four applications, we intend to evaluate the proportion of *swhidgets* used –or known but not used– by each participant, as well as to evaluate the proportion of users who use –or know but do not use– a specific control. To control learning and fatigue effects, the four applications are presented to participants in one of four possible orders according to a Latin square matrix.

8.2.3 *Participants*

Twenty-eight persons (sixteen males and twelve females) have participated in the laboratory survey. The average age is 39.5 year-old. As we focused on iOS *swhidgets* in this study, we only recruited iPhone users. Eighteen participants come from the *Inria Lille - Nord Europe research center* and from the University of Lille, although only 13 had a technical background. Elder persons and youngsters were recruited among some of our participants' relatives, although we could not recruit enough youngsters and did not use their data in the results reported here. A binary AgeGroup factor was created, distinguishing five participants aged 45 or more. We also considered a binary ComputerJob factor distinguishing the 13 participants who work in the IT industry or academic research at positions that require a technical background, from those who are not.

8.2.4 *Material and Apparatus*

The survey is equipped with one consent form, two questionnaires, one camera on tripod, pens to fill out the forms. We also provide an iPhone 6S running iOS 11, with a fictive account for all participants so that we exclude the risk of possible invasion concerning personal privacy in participants' own phone. To participate in this survey, participants are informed that the interaction section may be videotaped, if they agree, only with their hand movement when they interact with the iPhone. They have, therefore, an agreement to sign before beginning this survey. We set up a camera when the participants are ready to interact with the iPhone. The study obtained an agreement from Inria's Ethic Committee and Data Protection Officer.

8.2.5 *Protocol*

procedure Participants were invited to our team building in a dedicated room. As the main goal of the study was to assess usage of *swhidgets*, I only specified that the goal of the study was to observe interaction on smartphones without further details, to avoid making participants take the participation as a competition. Participants first read and signed a consent form, and answered various questions regarding their demographics and usage of touch-based devices. Then, they were presented with an iPhone in its default configuration and populated with artificial text messages and e-mails, before moving to the main phase of the experiment.

After the first questionnaire is filled out, we set up the camera and let the participant interact with the iPhone provided by Inria with fictive information such as mails and text messages. In the interest of collecting all the interactions that participants can do during the study, we also assure our participants that there is no pressure to be competitive, we may continuously ask if they know other possible way to do the task even if there are no other methods left. The four applications listed in Table 8.1 were administrated in a Latin Square order. After finishing all operations, participants are asked to fill out the second questionnaire. After the second questionnaire is filled out, we start debriefing all the interaction techniques that participants *did not* use in the study. The detail of important elements in this study are listed bellow:

first questionnaire In the first section, participants are asked to fill out the first questionnaire (provided in Appendix J), where we collect their age, professional and personal background. There are also questions aiming at knowing how the participants interact with their iOS system in different activities such as to play games, or to listen to music. There are also questions asking participants to estimate their knowledge of different smartphone operating systems on a Likert-type scale. At the end of this questionnaire, we also ask participants if they are interested in new technologies and to evaluate if they think themselves as a person who is highly interested in new technology and if they recognize themselves in eleven description with five-level Likert-type scales.

OPERATIONS SECTION is the moment I start to videotape participants and to observe their interaction vocabulary while dealing with given operations. Operations are divided into four different applications: Control Panel, Notification Panel, SMS and Mail, (see Table 8.1). The order of applications was counterbalanced across participants using a full Latin square, but the order of *Operations* in an application was the same for all participants. Participants were prompted to perform all the operations, which were grouped into *Applications*. For each operation, I set the smartphone to the home screen and asked the participant *Could you please perform <operation> ?* (for instance, *Could you please enable/disable WiFi?*). If participants did not know how to perform the operation, they were invited to reply *I do not know how to do it* and the operation was marked as *uncompleted*. If the participant was able to perform the operation, I logged which input method she used as *Primary*, and then reset the smartphone to the home screen and asked the participant *Do you know if this operation can be performed with an alternative method? If yes, could you please do it?* This could result in two new situations: either the participant knew an alternative and did it, or the participant replied *I do not know any alternative.* If participants knew an alternative method, I logged which alternative input method had been used as *Productive*, and then prompted participants again for alternative methods until the participants replied that they do not know any alternative, in which case the experiment moved to the next *operation*. Once the participant had completed all operations, I would pass to the second questionnaire.

THE SECOND QUESTIONNAIRE is provided in Appendix K. It was given after participants interacted with the iPhone. It aims at knowing firstly if participants think that these *swhidgets* help them in their daily tasks, then, to know the sources of knowledge from which participants learned about *swhidgets*.

DEBRIEFING SECTION Finally, the experiment concluded with a directed interview. In this section, I present all the input methods that the participant did not use during the experiment during the debriefing section, each time asking if the participant was aware of the method. An open discussion regarding the experiment ended the session, in which I investigate participants general feelings about *swhidgets*, if there are particular gestures that the participants suggest to trigger these *swhidgets*, and if participants have other expectations about what these *swhidgets* could do.

8.2.6 *Data Collection*

During the survey, participants could fail or succeed to complete an operation, and a unique PRIMARY input method was recorded for this operation only in case of success. We have thus computed for each *Operation* and participant a *primary-or-uncompleted* categorical dependent variable with four levels: *Swhidget*, *Navigation*, *Voice*, and *Uncompleted*. The within-subjects categorical independent variable is *Operation* (9 levels, see Table 8.1), and the factorial design is complete.

An ordinal *DoK* dependent variable was defined with values collected for each participant and interaction method (i. e., for each *Participant* × *Operation* × *Input method* triple), with four levels defined as follow (in decreasing order): The degree UNKNOWN was given when a participant had no idea that the interaction method could be used to accomplish the operation. The degree RECEPTIVE was given when the participant did not think about using the interaction method by herself, but acknowledged she knew about the method when asked after the operation. The degree USED was given when the participant effectively used the interaction method spontaneously or when she used it after the examiner asked her if she knew more ways to accomplish the operation. In addition, we recorded which interaction technique was used first to accomplish the operation (if any), providing data about the PRIMARY degree of knowledge. This last degree has however to be treated with care, since at most one interaction technique can have this degree for each operation-participant pair, and the observations for the different interaction techniques would thus not be independent. We thus exclude data relating to the PRIMARY degree when we want to compare interaction methods in a same operation, falling back to a three-level *DoK* ordinal dependent variable. We however use PRIMARY as a fourth ordinal level when comparing input methods across operations.

8.2.7 *Data processing*

Concerning the degrees of knowledge as ordinal data: I Followed the advice of Kaptein et al. [KNM10], and analyzed these data with Brunner and Puri's non-parametric tests for factorial plans [BP01] provided by the R package *nparLD* [Nog+12]. I report p-values computed with the ANOVA-type statistic, which is more appropriate for our number of participants (see [BP01]).

Concerning the *primary-or-uncompleted* categorical dependent variable, I analyze the dataset with two Cochran's Q test, one considering if the participants completed or failed each operation, and the other considering if the participants used *swhidgets* as PRIMARY or not. Since both tests are performed on the same dataset, I report the Bonferonni adjustments of the p-values.

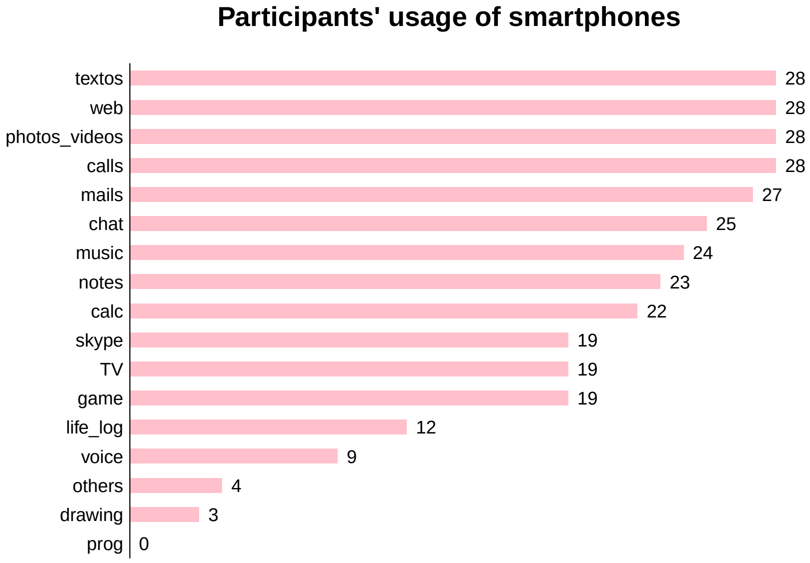


Figure 8.1: Proportion of the 28 participants of the laboratory study who declared to use their smartphone for various activities (See Appendix J for the corresponding questions).

8.3 RESULTS OF THE LABORATORY STUDY

8.3.1 *Participants*

EXPERIENCE WITH SMARTPHONES AND IOS Participants were relatively experienced in interacting with a touch-based devices: all 28 participants owned an iPhone for at least 3 months.

BACKGROUND Participants had various backgrounds although 13 had a professional activity in IT.

ACTIVITIES Figure 8.1 shows the activities that participants reported to practice with their smartphones. Four activities (text messaging, web browsing, taking photos/videos, and receiving calls) were reported to be practiced frequently by all the participants. Eight other activities (among which: checking mails, chatting, playing games and taking notes) were reported to be practiced frequently by at least 19 participants (67.9%). As expected, most participants use their iPhones for text messaging and checking e-mails, which are two activities implying operations included in the survey.

8.3.2 *Completion of Operations*

GENERAL OBSERVATIONS Average operation completion rate was 86.9% (33 operations uncompleted over the 252=9 operations $\times$ 28

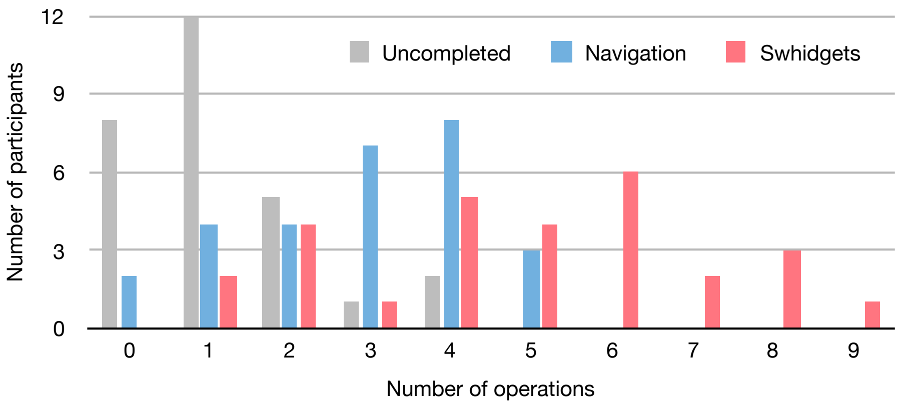


Figure 8.2: Distribution of participants by the number of operations in which they used *Navigation* or *Swhidgets* as primary, or that they did not complete.

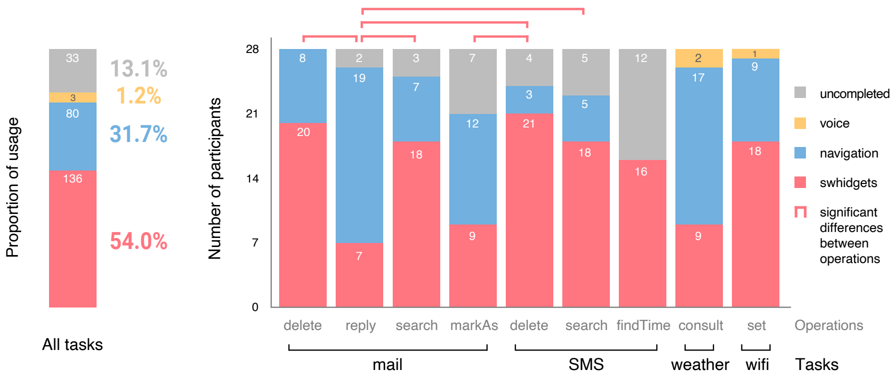


Figure 8.3: Distribution of primary answer across participants globally (left) and for each operation (right).

participants). However, the distribution of participants by the number of operations they did not complete, shown in gray in Figure 8.2, reveals that it was common for a participant to not be able to complete at least one operation. Indeed, only 8 out of 28 participants (28.6%) were able to complete all the operations. On the other hand of the spectrum, two participants (7.1%) could not complete as many as 4 of the 9 operations (44.4%). On average, each participant ended up with 1.18 operations uncompleted (13.1% of the 9 operations).

effects of operation on completion *Operation* had a statistically significant effect on *Uncompleted* frequency ($Q=40.5$, $df=8$, $p<10^{-5}$), as could be guessed from Figure 8.3. Post hoc pairwise comparisons between operations with Wilcoxon signed-rank tests revealed statistically significant differences in operation completion rate only between SMS.FindTime and each of mail.delete, mail.reply, weather.consult, and wifi.set. Three operations were completed by all participants: deleting mails, consulting the weather and setting the wifi. The least often completed operation was finding the time of an

Input method	Number of instances	Number of possible uses	% of completed operations	% of possible uses
Swhidgets	136	252	62.1%	54.0%
Navigation	80	252	36.5%	31.7%
Voice	3	84	1.4%	3.6%

Table 8.2: Distribution of primary usages by input method.

SMS (uncompleted by 42.9% of participants), followed by marking an e-mail as (un)read (25.0%).

8.3.3 PRIMARY *usage of input methods*

GENERAL OBSERVATIONS Table 8.2 gives the distribution of primary usages by input method and shows that *swhidgets* were globally the most frequent PRIMARY input method, with 136 of the 219 completed trials (62.1% of completed operations, 54.0% of the 252 possible usages).

The distribution of participants by the number of tasks they completed with a *swhidget* as PRIMARY, shown in blue in Figure 8.2, is rather wide and balanced. This suggests that knowledge of (or preference for) *swhidgets* was quite uneven among the participants.

EFFECTS OF OPERATION Figure 8.3 shows the distribution of PRIMARY input methods by operation. *Operation* had a statistically significant effect on the rate of PRIMARY usage of *swhidgets* over other input methods or non-completion of the operation ($Q=37.7$, $df=8$, $p<10^{-4}$). Post hoc pairwise comparisons between operations with Wilcoxon signed-rank tests revealed statistically significant differences only between 5 pairs of operations, which are shown in Figure 8.3.

8.3.4 *Degree of knowledge of Swhidgets vs. Navigation*

DATA PROCESSING AND ANALYSIS To investigate the difference in participants' knowledge of *swhidgets* and navigation controls, we perform a statistical analysis with two categorical within-subjects independent variables and one ordinal dependent variable in a complete factorial design. The independent variables are *Operation* (9 levels, see Table 8.1) and *Input method* (2 levels: *Swhidgets* and *Navigation*, since we are not interested in *Voice*, here). The ordinal dependent variable is the *degree of knowledge*, with only three levels (PRODUCTIVE, RECEPTIVE, and UNKNOWN) since we treat PRIMARY as simply PRODUCTIVE to ensure the hypothesis of independence between the two levels of

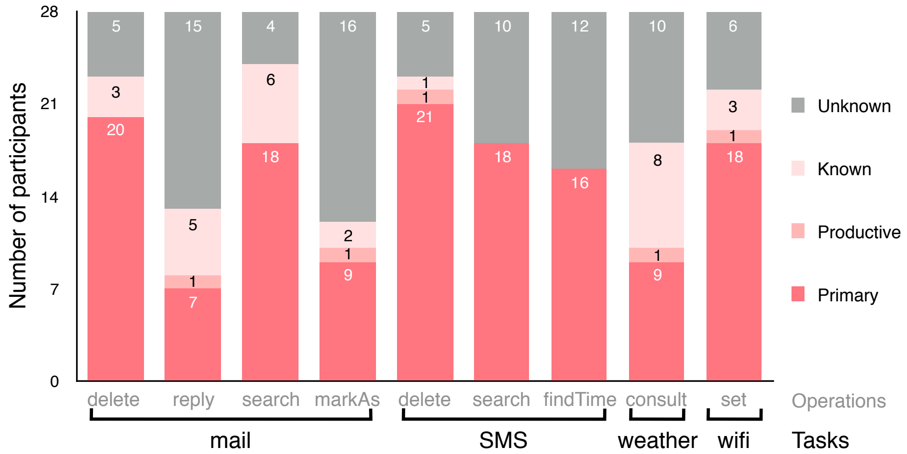


Figure 8.4: Degree of knowledge of *swhidgets* by operation, with an additional level to differentiate the use of a *swhidget* as the interaction method spontaneously used by the participant from a later use.

Input method (for a participant and task, there can only be one input method labeled PRIMARY).

Following Kaptein et *al*. recommendation to the HCI community [KNM10], statistical analysis was performed with Brunner' and Puri's non-parametric test LD.F2 from the *nparLD* R package [BP01; Nog+12]. We report test results for the ANOVA-type statistic ATS_{df}, which can be approximated by a $F_{df,\infty}$ distribution for small and moderate sample sizes [BP01, p. 36].

FACTORS AFFECTING OVERALL DEGREE OF KNOWLEDGE Figure 8.4 shows the degrees of knowledge reached by participants for *Swhidgets* in each *Operation*. Statistically significant effects of *Operation* on *degree-of-knowledge* were observed ($ATS_{5.89}=7.87$, $p<10^{-7}$), but no effect of *Input method* could be found ($ATS_1=1.73$, $p=.18$). However, interaction between these two factors was statistically significant ($ATS_{6.13}=9.443$, $p<10^{-9}$), suggesting that *Input method* had different effect directions depending on the *Operation*. Post-hoc tests conducted with the LD.F1 test and Holm-Bonferroni adjustments indeed revealed a statistically significant effect of *Input method* for SMS.findTime ($ATS_1=18.5$, $p<10^{-4}$, $p^\star<10^{-3}$), SMS.delete ($ATS_1=18.4$, $p<10^{-4}$, $p^\star<10^{-3}$), mail.reply ($ATS_1=16.1$, $p<10^{-4}$, $p^\star<10^{-3}$), and weather.consult ($ATS_1=16.1$, $p<10^{-4}$, $p^\star<10^{-3}$). Direction of effect was different in these four operations, with a better knowledge of *swhidgets* for SMS.findTime and SMS.delete, but the opposite for mail.reply and weather.consult.

DEMOGRAPHIC FACTORS We could not observe any statistically significant effect of TASKORDER, GENDER, AGEGROUP, or – more surprisingly – COMPUTERJOB on degrees of knowledge in general (see Section 8.2.3 for the definition of these groups). We could neither

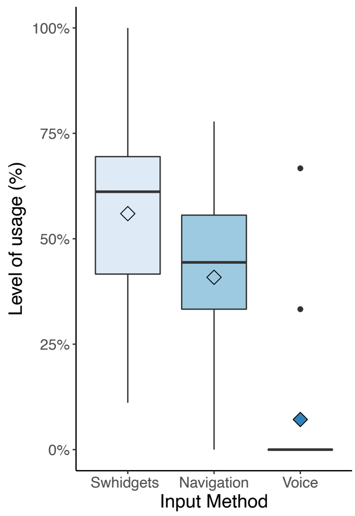


Figure 8.5: Distribution of levels of usage by input method.

find a summary statistics of participant's interest in new technologies (based on the questions from the first questionnaire, see Appendix J) that would correlate with participants' degrees of knowledge.

8.3.5 PRODUCTIVE *knowledge*

GLOBAL PRODUCTIVE KNOWLEDGE I define as a *level of usage* the proportion of input methods that were USED by a participant during the experiment, out of the available input methods of a given type. On average, the *global* level of usage (proportion of input methods USED out of the 21 available) of our participants was of 42.5% (sd=28.8), which means that they used less than half of all the available input methods.

METHODOLOGY I investigate the effects of *input method* and *operation* on participants' ability to effectively use an input method to perform an operation, by merging the RECEPTIVE and UNKNOWN levels of the dependent variable *degree of knowledge* into a single UNUSED. When performing statistical tests for this analysis, I ignore the *Voice* input methods – since they were not available in all operations – and thus consider only exactly one *Swhidden* and one *Navigation* input method by *operation*.

EFFECTS OF INPUT METHOD The level of usage was of 56.0% (sd=24.7%) for *Swhidgets*, 40.9% (sd=16.9%) for *Navigation* and 7.1% (sd=18.9%) for *Voice* (Figure 8.5). Figure 8.6 shows the distribution of productive knowledge across participants and input methods, suggesting that usage of *swhidgets* (median=5.6 operations) is better known than usage of navigation controls (median=3.8 operations). Statistically significant effects were observed for *Input method* ($ATS_1 = 4.70$,

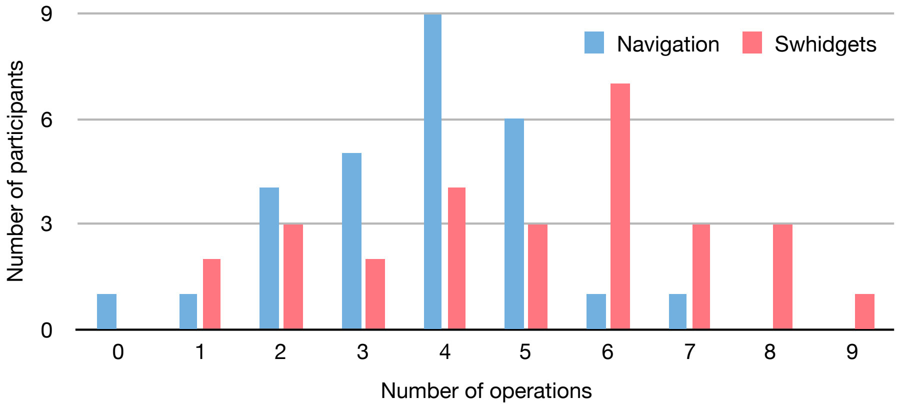


Figure 8.6: Distribution of participants by the number of operations in which they used *swhidgets* and navigation controls (USED).

$p = .030$) in direction of a better productive knowledge of *swhidgets* than for navigation controls.

EFFECTS OF OPERATION Statistically significant effect of *Operation* ($ATS_{6.19} = 5.44$, $p < 10^{-5}$) and of the interaction *Operation* × *Input method* ($ATS_{6.16} = 10.42$, $p < 10^{-11}$) were also observed. Post hoc tests confirmed the effect direction and statistical significance of *Input method* for the four operations identified in section 8.3.4, and also revealed an effect toward *Swhidgets* for SMS.search ($ATS_1 = 8.24$, $p = .0041$, $p^\star = .021$).

This result suggests that participants tend to use in priority *swhidgets* (assuming they know them) over navigation controls, which is not so surprising considering that *swhidgets* often acts as shortcuts (e.g. for Wifi or for deleting an e-mail) or provide a more detailed information (typically for finding the time a sms message was sent at). Moreover, 22 of the 28 participants (78,6%) also reported during the post-experiment interview that *swhidgets* improve their everyday interaction, which can also explain why participants preferably use *swhidgets* when they know about them. While the priority of usage of Voice control is extremely low, it is important to note that this result is artificially amplified by our experimental design since only 3 operations could be performed using voice control. Therefore, it was virtually impossible for a participant to systematically use voice control as Primary, except if that participant completed only the 3 tasks where voice control was available. That being said, we must stress that 26 of our 28 participants never used voice control as primary.

8.3.6 *Awareness and* RECEPTIVE *knowledge*

METHODOLOGY I investigate the effects of *Input method* and *Operation* on participants' awareness of the existence of an input method to

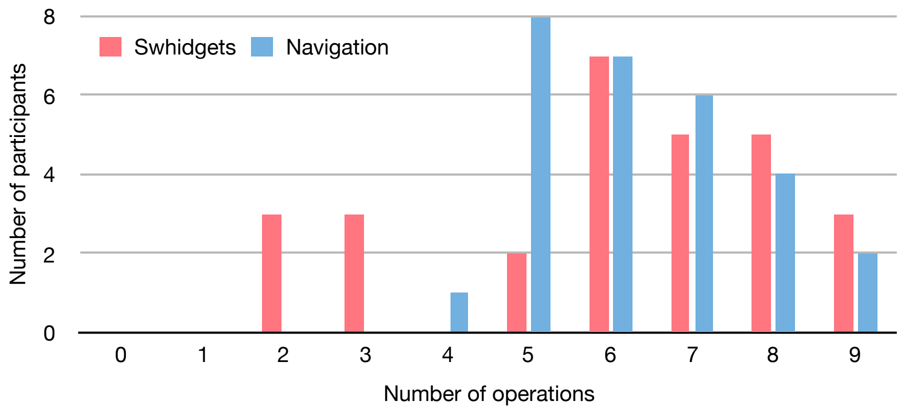


Figure 8.7: Distribution of participants by the number of operations in which they were aware they could use *swhidgets* and navigation controls (KNOWN).

perform an operation, by merging the PRODUCTIVE and RECEPTIVE levels of the dependent variable *degree of knowledge* into a single KNOWN.

GLOBAL RECEPTIVE KNOWLEDGE On average, 20.6% of all input methods were classified as RECEPTIVE by our participants (121 out of the $28 \times 21 = 588$ input methods). In other words, there were around one fifth of the input methods demonstrated to them that participants could recognize but would not think about using. Together with the 42.5% of input methods classified as USED, it means that participants were aware of 63.1% of all the input methods considered in this study.

EFFECTS OF INPUT METHOD Input methods classified as RECEPTIVE make 11.1% of all *Swhidgets*, 29.8% of all *Navigation* controls, and 21.4% of all *Voice* controls. However, these differences seem to disappear at the KNOWN level: Figure 8.7 shows the distribution of participants by the number of *Swhidgets* and *Navigation* input methods they are aware of, suggesting that awareness of *swhidgets* (median=6.50 operations) is similar to the awareness of *Navigation* controls (median=6.07 operations). Indeed, no statistically significant effect of *Input method* on KNOWN could be found ($ATS_1 = .70$, $p = .40$)

EFFECTS OF OPERATION Statistically significant effect of *Operation* ($ATS_{5.54} = 8.02$, $p < 10^{-7}$) and of the interaction *Operation* × *Input method* ($ATS_{6.07} = 6.13$, $p < 10^{-5}$) were observed using the methodology described above. Post hoc tests confirmed the effect direction and statistical significance of *Input method* found in section 8.3.4 for mail.reply, SMS.findTime, and weather.consult.

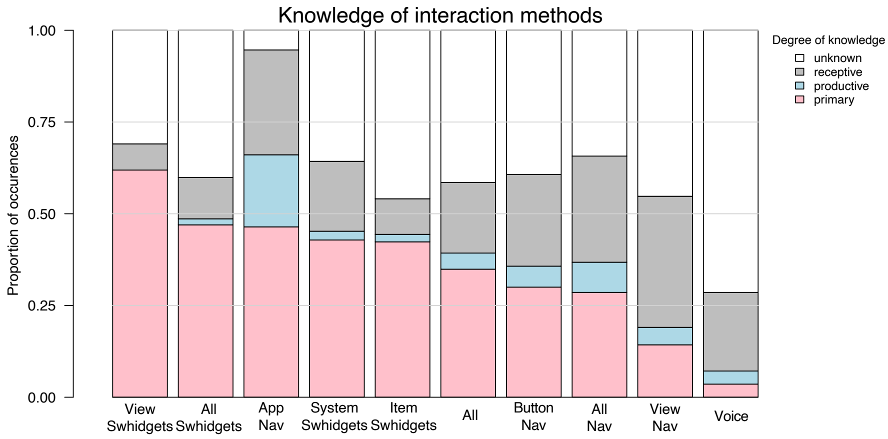


Figure 8.8: Distribution of Degrees of Knowledge by interaction method, sorted by decreasing proportion of primary usage.

8.3.7 *Preference for swhidgets*

When participants knew how to use a *swhidget* to complete an operation (i.e., at the USED level), they generally used it first (i.e., at the PRIMARY level). Indeed, the ratio of PRIMARY instances over USED instances is 96.5% (136/141) for *Swhidgets*, 77.7% (80/103) for *Navigation* controls, and 50% (3/6) for *Voice* – although there is not enough data in this last case to draw a conclusion. Figure 8.8 suggests that this preference for *swhidgets* holds for all three types of *swhidgets*, but do not hold for any of the other input methods.

8.3.8 *How participants discovered swhidgets*

Fifteen participants reported that at least one of the *swhidgets* they used during the experiment had been discovered while consulting an on-line resource (either a Website, a Blog or a video tutorial). Participants also reported to heavily rely on their social network to discover *swhidgets*, as 23 of our 28 participants reported to have discovered at least one *swhidget* following a demonstration by a friend or colleague. Finally, the main knowledge source that 27 of our 28 participants reported to rely on is through self-discovery while interacting with the device, either by accident, or by trying to apply gestures that they know from a different touch-based operating system (typically Android in this case). Figure 8.9 shows for each possible source of knowledge the proportion of participants who acknowledged that they used such a source, with 95%-level confidence intervals.

It is important to note that these results report only *how* participants discovered at least one *swhidget* they used in the experiment, but this must still be put in perspective with our participants' use of *swhidgets*. On average, participants still ignored more than 4 *swhidgets* out of 9, motivating the importance to help users discovering these widgets,

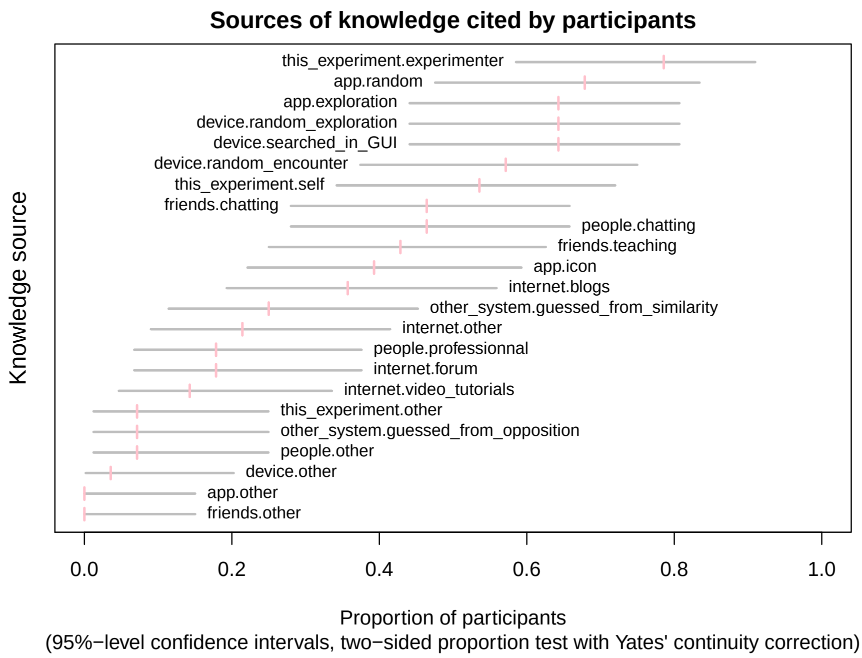


Figure 8.9: Proportion of participants who cited each source of knowledge, with 95%-level confidence intervals, sorted by increasing proportion.

either through alternative graphical design or by fostering exploration and social exchanges about touch-based user interfaces.

Although the large overlapping of the confidence intervals does not allow to conclude with certitude, the trend seems to be that the applications and devices used are the most frequent sources of knowledge, above other people and Internet. The latter sources of knowledge can however not be neglected, as e.g. blogs are cited by at least 19.3% of the participants. The high rank of "this_experiment.experimenter" is another hint that participants have a low level of knowledge for the interaction methods used in this experiment.

8.3.9 *Participants' reaction about swhidgets*

In the final part of the survey, 22 of the 28 participants (78,6%) reported that *swhidgets* "improve their everyday interactions", which can also explain why participants preferably use hidden controls when they know about them. The 95%-level confidence interval for this proportion is [58.5,91.0], which is clearly above the status-quo, but also leaves some room for a non-negligible proportion of people who disagree with this statement. During the post-experiment interview, these 22 participants shared their intention to not only keep using the *swhidgets* they already knew before the survey, but to also consider using the ones they learned at the end of the survey.

8.4 DISCUSSION OF THE RESULTS OF THE LABORATORY STUDY

8.4.1 *Reflection on the study*

Summary. I conducted a controlled experiment investigating the usage of *swhidgets* on iOS by prompting 28 participants in majority from a IT background to perform various operations on the smartphone, and observing which input method they used. This experiments showed that participants have a moderate level of knowledge of *swhidgets* as they used on average slightly more than half of the *swhidgets* available in our experiment. The results also suggest that users tend to prioritize *swhidgets* over alternative input methods (assuming that they are aware of them).

Usage of Navigation controls. One surprising result of this experiment was the relatively low level of usage of Navigation controls, which was significantly lower than for *swhidgets*. While surprising at first, it can be explained by the fact that three operations (*search SMS, find time SMS, and Search Mail*) were perceived as unclear by the participants. Indeed, unlike the other operations, the *Navigation control* alternative for these operations consisted in browsing the corresponding SMS/E-mails rather than activating a command by tapping on specific widgets. Therefore, *find time SMS* was the least completed operation, which in the end also resulted in participants reporting significantly more *Navigation* controls than *swhidgets* as RECEPTIVE.

Experimental design decisions. I investigated the problem of *swhidgets* usage with a controlled experiment because I believe this methodology offered the best trade-off between applicability and validity of results. Ideally, *swhidgets* usage would be investigated by instrumenting users' device in order to monitor everyday interaction and precisely quantify the proportion of *swhidget* usage. Unfortunately, accessing that type of information on the iOS operating system is not possible without asking the participant to *jailbreak* his or her device. An alternative method would be to record the screen of the device of the user and post-annotate her interaction but it would raise obvious privacy issues.

8.4.2 *When participants did not use swhidgets*

A large majority of participants (78.6%) agreed that *swhidgets* do "help"[5] them with their everyday tasks on touch-based devices, which is understandable considering that most *swhidgets* provide shortcuts to frequently used features, make it possible to achieve faster a daily task, or simply allow to access information that would not be accessible otherwise (typically, the exact time a text message was received). That being said, this result also means that 21.4% did not consider that

5 The wording of the question was voluntarily vague, in order to stimulate discussion about how *swhidgets* are useful or not in the open interview after the questionnaire.

swhidgets are helpful for their everyday interaction. I review here some reasons given by the participants not to use *swhidgets*.

Obviously, *swhidgets* can bring benefits if users know how to use them. The results of this survey show that even if most participants knew at least a few interaction methods based on *swhidgets*, the knowledge of *swhidgets* is not evenly spread. In particular, the knowledge of one interaction method does not seem to transfer well to similar interaction methods in other contexts. It often occurred, in the debriefing part of the survey, that after being introduced with such a hidden-control-based interaction method, participants reacted with a reflection such as "I've heard about it from somewhere but I didn't knew I can also use it here" or "I wasn't aware of that, but I will do it this way from now."

When users don't know how to use *swhidgets*, it does not only affect their overall performance: it can actually bring confusion to users. Several participants reported that because some types of *swhidgets* cannot be used everywhere the design would technically allow it (e.g. there is no swipe from the left in the SMS application), they had no clue about when to use them or not, so they rather sticked to using the navigation controls (visible widgets) instead.

Another interesting observation is that users can have a wrong perception of the performance gain provided by *swhidgets*. For instance, several participants who knew how to delete emails with a hidden button still preferred to tap on the visible button, because they believed it is faster to select all the emails that they want to delete and then put them into the trashcan together.

8.4.3 *Task-based vs. System-based Interaction Models*

The analysis of the interaction vocabularies for *swhidgets*, shown in Figure 8.7, reveals that it is not a bimodal distribution with peaks at the extreme values, which should be the case if users were perfectly able to transfer their knowledge from one interaction method to another of the same type: they would know all of them or none of them. A possible interpretation would be that users tend to build interaction models for each task, rather than build global interaction models to match the system's overall design: They try to find interaction possibilities on the basis of the task they want to achieve rather than on general interaction models that they have learned. The rather low score of "other systems" as a source of knowledge in figure 8.9 tends to support this hypothesis. If this interpretation holds, it means that the phenomenon of "interface layouts" that we have observed in the analysis of *swhidgets* in Chapter 6 is not successful as a way to ease transfers of knowledge between applications.

8.5 study 2: online survey on awareness, usage and discovery of swhidgets

The results of our online study suggest that users have an uneven knowledge of *swhidgets*, and that this input method tend to be preferred to other types of interactions. We then conducted a second, larger scale, online study in order to confront the results of the previous study in terms of awareness and usage of *swhidgets*, but also gain more insights on their perceived efficiency, as well as better understand how users discover them.

8.5.1 *Design and Procedure*

We designed an online survey in order to investigate usage of various *swhidgets*. It started by asking demographic questions, and general questions regarding smartphone/tablet usage and interest in new technologies. Then, for each operation, the respondent was first asked whether she knows how to perform it or not, and if yes, she was asked how frequently this operation was performed. Then, for each associated input method, she was first shown a video demonstrating this input method and then asked whether this input method was new to her. If the input method was unknown, she was asked to report on a 1-7 scale if the input method looked *easy to discover*, *more convenient than the way usually used to perform the operation*, and if she *would consider adopting this input method in the near future*. If she already knew the input method, she was asked if using it is part of her everyday interaction, if the technique was *easy to discover* (1-7 scale), and if she remembered how she discovered it (for *swhidgets* only). Finally, regardless of the input method, we asked if the respondent would have considered any reasons not to adopt it. This sequence of video and questions was then repeated for each input method corresponding to this operation. Before to move on to the next operation, the respondent was asked to rank all possible input methods by order of preference. Having run through all the *operations*, respondents reach the final part of the survey asking various questions regarding *swhidgets*, among others about their perceived performance and assessing if the logic behind the physical metaphor was understood.

The survey investigated 13 *operations* across 4 *applications* (see Table 8.1, orange). Order of the application was counterbalanced across participants, and order of operations was randomized within an application with two exceptions[6]. Order of input methods was the same for all operations. Similarly to study 1, *Voice* control was included for ecological validity but often ignored in the following analysis.

6 *Reveal* for Notification Center and Control Panel which were always presented first as they are needed to achieve the other operations.

8.5.2 *Apparatus and Participants*

The survey was created using the LimeSurvey platform [19] and the RemoteControl 2 API. 144 participants were recruited via calls for participation on social networks. All were daily iPhone users, with an average age of 36 year old (sd=9.54). All data were collected anonymously with a limitation of legal age, participation was entirely voluntary and not compensated. The study obtained an agreement from Inria's Ethic Committee and Data Protection Officer.

8.6 RESULTS OF ONLINE STUDY

8.6.1 *Participants Background and Experience*

The survey was completed by 144 respondents (114 females, 29 males, 1 other), aged 18-75 (mean=36.4, SD=10.0). They were relatively educated, with 82.6% having at least an undergraduate-level degree, 74.3% having graduated, and 22.9% owning a master's degree or higher. Participants had various backgrounds, as they described their main working industry using 27 of the 30 different industrial categories. Overall, participants reported to be interested in "collecting the latest technology news", the median response on a 1-7 scale[7] being 5 (Q1=4, Q3=6), with 59.0% of respondents answering 5 or more, and only 4.9% answering 3 or less. Participants reported to be quite experienced with iOS devices, as 93% used one for at least two years (over 6 years for 48.6%). 93.1% of them declared spending at least one hour a day on an iOS device, 67.4% over three hours, and 43.8% over five hours. On a scale from 1="I know nothing" to 7="I know everything", they all evaluated their knowledge of how to interact with their iOS device at 3 or higher, strictly above 3 for 93.1%, and at 5 or more for 74.8% of them (Q1=4, median=5, Q3=6). This score positively correlates with the agreement to "I am highly interested in collecting the latest technology news" ($\rho = .486$, $p < 10^{-9}$), which suggests that following the technology news contributes to participants' knowledge of iOS, or at least to their confidence in knowing it.

8.6.2 *General Usage*

Overall knowledge of input methods. Of all 28 input methods presented in this study, participants were aware of 19.7 (70.4%) in average (Q1=15, median=21, Q3=25), knew how to use 18.1 (64.6%) (Q1=14, median=19, Q3=23), and reported to effectively use 13.9 (49.6%) (Q1=10, median=15, Q3=17).

Application and task completion. The four applications used in this study were well known, evaluated by the awareness of at least one

7 with 1 being "totally disagree" and 7 "totally agree"

input operation to complete them: 96.5% of participants for the Control Panel, 92.4% for Notification Center (92.4%), 91.0% for Messages (91.0%) and 84.0% for Mail (84.0%). Among all 12 tasks, searching a mail is the one with the smallest proportion of participants who know at least one input method to do it (67.4%), while activating Wi-Fi is the most frequently known task (95.8%). Lack of knowledge of the tasks can be explained only partially by a lack of knowledge of the applications: the proportions of participants who know at least one input method to perform the task among those who know the application range from 77.9% for seeing SMS timings to 99.3% for setting the Wi-Fi (Q1=84.1%, median=92.3%, Q3=94.7%). The 12 tasks had statistically significant differences in reported usage frequency (Skillings-Mack statistic = 307.1, $p < 10^{-6}$). Based on reported frequency, the most frequent tasks were all notification-related tasks and revealing the Control Center (several times a day), followed by turning the Wi-Fi on or off, (un)locking the screen orientation, and all Mail-related tasks (several times a week), and finally, flashlight and Messages-related tasks (once a week or less).

8.6.3 *Awareness and Usage of Swhidgets*

Awareness. We evaluate user awareness of a *swhidget* by the proportion of participants stating that the input method was not new to them. Awareness rates of *swhidgets* range from 35.4% (*see SMS times*) to 95.1% (*set Wi-Fi*), with a median of 80.2% (Q1=64.0%, Q3=89.1%). Considering only participants who actually knew how to complete the operation, awareness rates of *swhidgets* ranges from 50% to 100% with a median of 95.0% (Q1=89.8%, Q3=99.8%). A repeated measures logistic regression was used to model participants' odds of being aware of an input method (when they know the task it solves), and revealed statistically significant effects of task ($p < 10^{-15}$), input method ($p < 10^{-15}$), and interactions between task and input method type ($p < 10^{-12}$). Post-hoc Tukey tests show statistically significant differences between all types of input methods, with average awareness rates for voice (64.4%) being lower ($p < .001$) than for *swhidgets* (88.6%), themselves lower ($p = .020$) than for navigation controls (96.6%). A logistic regression on the *swhidgets* awareness odds alone showed that awareness rates for the three types of *swhidgets* were different ($p < 10^{-15}$). Post-hoc Tukey tests show that awareness rates for view *swhidgets* (78.4%) are lower ($p = .009$) than those for item *swhidgets* (80.8%), which are themselves lower ($p < .001$) than those for system *swhidgets* (98.3%).

Usage. We evaluate users' usage of a *swhidget* by the proportion of participants stating that they use the corresponding input method or used it in the past. Usage rates are high for most *swhidgets*, ranging from 28.5% (*see SMS times*) to 84.7% (set Wi-Fi), with a median of

66.0% (Q1=54.3%, Q3=81.6%)[8]. When only considering users who were aware of the *swhidget*, the proportion ranges from 66.3% to 94.6% with a median of 83.1% (Q1=81.2%, Q3=89.4%). A repeated measures logistic regression modeling participants' odds of knowing an input method they are aware of revealed statistically significant effects of task ($p<10^{-15}$), input method type ($p<10^{-15}$), and interactions between task and input method ($p<10^{-4}$). Post-hoc Tukey tests revealed that usage of voice control (29.8%) is significantly lower ($p<10^{-5}$ for both) than navigation controls (83.4%) and *swhidgets* (86.5%), but the difference between *swhidgets* and navigation control is not significant. A logistic regression on the *swhidgets* alone showed that usage rates for the three types of *swhidgets* were different ($p<10^{-15}$). Post-hoc Tukey tests show that usage rates for view *swhidgets* (85.1%) are lower ($p=.0084$) than those for item *swhidgets* (80.7%), which are themselves lower ($p<.001$) than those for system *swhidgets* (90.5%).[9]

Preferred input method. Unlike in previous study, we could not query which input method is spontaneously used first for performing an operation. Instead, we asked participants to rank by order of preference all possible input methods for each operation. The proportions of participants aware of a *swhidget* for an operation who ranked it as their preferred input method range from 39.5% (delete mails) to 91.5% (flashlight), with a median of 68.6% (Q1=53.6%, Q3=72.3%). These results suggest that users tend to prefer *swhidgets* when they are aware of them, but that it also depends on the operation. For instance, several respondents reported to use conventional buttons for deleting e-mails because it is more convenient, as they often delete them right after reading and using the *swhidget* would require to come back to the list first. Note that among respondents who were not aware of the *swhidget* for an operation, proportions range from 18.2% to 85.7% (median=40.0%, Q1=25.0%, Q3=53.6%), showing that participants could also be highly interested in discovering some of the *swhidget* they do not know. As illustrated on Figure 8.10-Left, participants preferred *swhidgets* for 6 tasks out of 9. It was preferred over *Navigation* methods 5 times out of 8 when both input methods were available. *Voice* was by far the less preferred input method, which is not surprising given that respondents were also much less aware of it.

8 Note that for 22 respondents out of 144, a database issue prevented us to retrieve the answer to this question for the *swhidget* of the Messages search operation.

9 These proportions are not in the order revealed by the logistic regression because the latter accounts for variations in the knowledge of participants, and therefore reduces the weight of those who know more input methods. On the other hand, the proportions given are biased toward the behavior of the most knowledgeable participants.

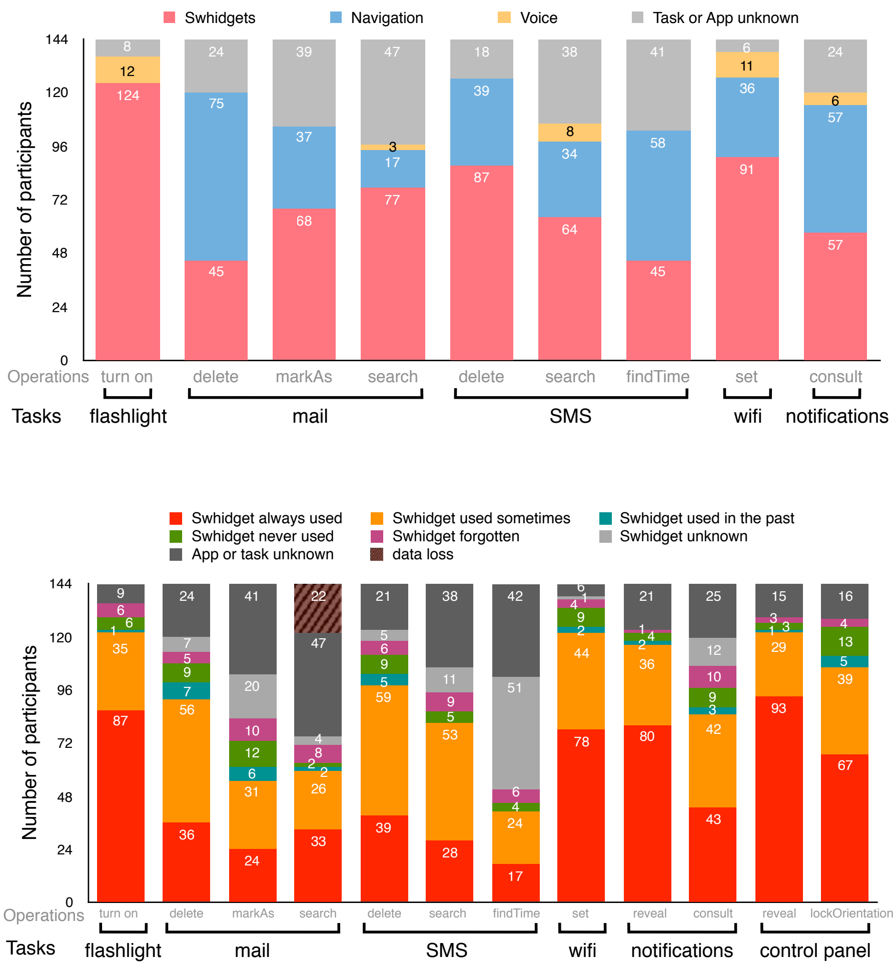


Figure 8.10: Top: Distribution of preferred input techniques to perform operations, when alternatives to *swhidgets* exist;
Bottom: Distribution of usage frequency of *swhidgets*.

8.6.4 *Discovery of Swhidgets*

SOURCES OF KNOWLEDGE On all participants and operations, exploration of the interface was the most frequently reported source of discovery of *swhidgets* (42.6%), followed by accidental activation (15.9%), transfer of knowledge from another application or system (8.8%), explicitly looking or asking for it (8.6%), informing herself generally about the interface or application (6.3%), being demonstrated the input method (4.9%), remembering seeing someone else using it (3.1%), and others (1.0%). Surprisingly, respondents replied "I cannot remember" for only 8.8% of the *swhidgets*.

The three most cited sources of knowledge (exploration, accidental discovery, and transfer of knowledge) all concern knowledge acquired from the usage of the interface and systems themselves rather than from external sources, and account together for 65.3% of participants' answers, almost three time more than for external sources of knowledge – which still account for 22.9% or answers.

That being said, it would be presumptuous to over-interpret these results, as it remains uncertain that participants remember well how they learned an input method, especially since our participants have a long experience of iOS and probably learned the *swhidgets* a long time ago. As such, it is likely that they assumed how they discovered them, rather than how they actually did discover them.

SWHIDGETS' PHYSICAL METAPHOR There was generally no consensus on the questions I7 "I can tell from the interface which on-screen objects could be moved to reveal *swhidgets*" (median=4, Q1=2, Q3=5) and I8 "I can tell from the interface in which direction I should move objects to reveal *swhidgets*" (median=4, Q1=3, Q3=5). These two ratings were strongly correlated ($\rho = .709$, $p < 10^{-22}$), which suggests that the design of *swhidgets* in iOS allows users to integrate these two aspects of the interaction in a single concern. The consistence of *swhidgets'* design in iOS interactions was better acknowledged, however, with 61.1% of participants agreeing (scored 5-7) and only 8.3% disagreeing (scored 1-3) on I9.

Although there was no consensus on the statement I6 "I do not expect controls to hide under interface elements or bezels", 45.8% respondents disagreed (scored 1-3) whereas 22.2% agreed (scored 5-7). Agreement with this statement was slightly and negatively correlated with the interest for technologies ($\rho = -.226$, $p = .0064$), which suggest that despite being rooted in real world physics, additional knowledge about technological product interfaces might be necessary to grasp this aspect of the metaphor.

TRANSFER OF KNOWLEDGE THROUGH APPLICATIONS There was a clear consensus among participants to agree with the statement

Please tell us how you feel about using swipe gestures to reveal *swhidgets*.								
I1	I enjoy manipulating *swhidgets*.	2	0	1	18	21	36	65
I2	When I interact with *swhidgets*, the interface responds to my inputs in a way that feels satisfactory.	1	0	2	14	23	36	67
I3	Swipe gestures are difficult to perform.	65	39	14	11	5	5	4
I4	The system often misinterprets my swipe gestures.	46	26	20	29	10	8	4
I5	The animation of *swhidgets* is sometime unpredictable.	43	31	15	35	6	9	4
Swhidgets behave like "real-world" objects that you physically manipulate: they slide on flat surfaces, they are bound to other objects that gets dragged or pushed with them, and they can hide under other objects. Please tell us how you feel about the use of this metaphor in the design of *swhidgets*.								
I6	I do not expect controls to hide under interface elements or bezels.	33	16	17	45	7	8	17
I7	I can tell from the interface which on-screen objects could be moved to reveal *swhidgets*.	23	21	9	45	17	10	18
I8	I can tell from the interface in which direction I should move objects to reveal *swhidgets*.	19	14	21	47	11	12	19
I9	This design is coherent with the design of other iOS interactions.	1	3	3	48	23	30	35
Now, we would like to know about the possibility of leveraging the knowledge of specific *swhidgets* to discover interaction possibilities in other contexts. Please complete the sentence bellow with the following statements and evaluate your agreement to them: *When I know a specific swhidget in an application, I...*								
I10	...suspect other places with a similar screen layout to provide a similar *swhidget*.	5	2	3	29	21	28	55
I11	...do not expect I can swipe it in another direction to reveal other functions.	25	20	10	42	21	11	14
I12	...play with it to find all the possible ways to use it.	7	3	10	36	19	24	44

Table 8.3: Questions in the last part of the questionnaire, with histograms of responses from totally disagree (left) to totally agree (right) and box pots showing quartiles.

We will now consider how the placement and type of *swhidgets* relate to the functions they provide: can you find this logic and make it yours?

I13	I usually understand the logic prevailing to the different types of *swhidgets* and their placement in the interface.	3	11	5	31	28	34	31
I14	I can use this logic to guide my search for a *swhidget* with the function I need.	1	6	8	28	31	35	34
I15	I sometime feel that different parts of an application failed to follow a common logic for their *swhidgets*.	1	11	9	42	20	34	26
I16	I noticed *swhidgets* that follow a similar logic in different applications or different systems.	4	6	9	53	19	26	26
I17	Once I've learned how to do something with a *swhidget*, I cannot think anymore about another way to do it.	5	7	8	40	22	27	34
I18	I sometimes forget that there are *swhidgets* that I can use to do a task.	6	11	8	39	29	29	21
I19	I sometimes need to be reminded where are the *swhidgets*.	13	17	11	33	26	19	24

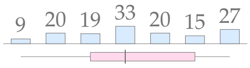
Can you use your knowledge of how to accomplish tasks with *swhidgets* to find other situations you could possibly use them?
Is your knowledge of how to accomplish tasks with *swhidgets* transferable to other situations?
Please evaluate your agreement with the following statements.

I20	If I often use a *swhidget* for some function in one application, I expect similar functions to be provided by similar *swhidgets* elsewhere.	0	2	0	31	20	33	57
I21	When I'm searching for a way to achieve a task or a better way than the one I know, I think about looking for *swhidgets*.	2	0	2	34	28	27	50

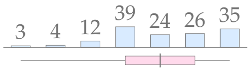
Table 8.4: (Continuation of Table 8.3) Questions in the last part of the questionnaire, with histograms of responses from totally disagree (left) to totally agree (right) and box pots showing quartiles.

Does using *swhidgets* in your daily tasks make you feel in control of the situation? Please evaluate your agreement to the following statements.

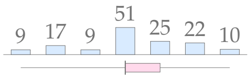
I22 I sometimes worry about triggering *swhidgets* by mistake.



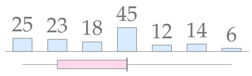
I23 I use *swhidgets* only when I am sure of what I am doing.



I24 I swipe when I want to use *swhidgets*, but the commands it reveals are not always the ones I thought.

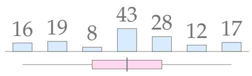


I25 Using *swhidgets* requires some attention and distracts me from my tasks.

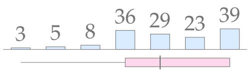


Does using *swhidgets* in your daily tasks make you feel like an advanced user? Please evaluate your agreement to the following statements.

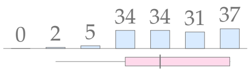
I26 I cannot remember all these *swhidgets*.



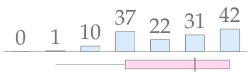
I27 I consider *swhidgets* as advanced techniques.



I28 In general, I know when it is possible to use a *swhidget* to achieve a task I want to do.

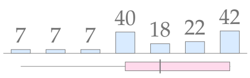


I29 *Swhidgets* are the fastest way for me to do my tasks.

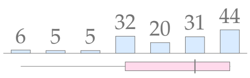


Finally, we would like to know if *swhidgets* have an impact on your ability to organize your life so as to be in harmony with your values.
Please evaluate your agreement to the following statements.

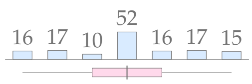
I30 There are tasks I could not or would not do without *swhidgets*.



I31 There are situations I could not or would not use my phone to do some task, without *swhidgets*.



I32 I do not want to remember so many *swhidgets*.



I33 Knowing many *swhidgets* is a legitimate source of pride.

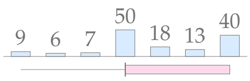


Table 8.5: (Continuation of Table 8.4) Questions related to the Self-Determination Theory in the last part of the questionnaire, with histograms of responses from totally disagree (left) to totally agree (right) and box pots showing quartiles.

"When I know a specific *swhidget* in an application, I suspect other places with a similar screen layout to provide a similar *swhidget*." (I10: Q1=4, median=6, Q3=7).

Most participants agreed that when they discover a *swhidget*, they "play with it to find all the possible ways to use it" (I12: Q1=4 median=5, Q3=7). Answers to this item were also well correlated to the answers to the previous item about other places with a similar *swhidget* ($\rho = .540$, $p < 10^{-11}$) and negatively correlated with participants' age ($\rho = -.231$, $p = .0053$). Self-reported knowledge of iOS interaction was slightly correlated to this item's answers ($\rho = .283$, $p < .001$), which suggests that users who "play" with *swhidgets* to explore interaction possibilities have more chance to know about iOS interaction or, at least, to feel more confident about their knowledge.

Participants are divided on the idea that having discovered a *swhidget*, they should expect it can be swiped in another direction to reveal other functions (I11: 38.2% disagree, 31.9% disagree, median=4, Q1=2, Q3=5). This is not necessarily surprising considering that *swhidgets* are not systematically symmetric. Typically, lists in many applications have a delete *swhidget* on the right of the items but do not necessarily have *swhidgets* on the left.

MENTAL MODELS AND THE LOGIC OF SWHIDGETS We asked participants to evaluate their agreement with seven statements concerning "how the placement and type of *swhidgets* relate to the functions they provide [and if participants could] find this logic and make it [theirs]". There was a global agreement (Q1=4, median=5, Q3=6) for the following six statements: I13 participants understand the logic prevailing to the different types of *swhidgets* and their placement in the interface; I14 they can use this logic to guide their search for a *swhidget* with the function they need; I15 they sometime feel that different parts of an application failed to follow a common logic for their *swhidgets*; I16 they notice *swhidgets* that follow a similar logic in different applications or systems; I17 once they learned how to do something with a *swhidget*, they cannot think anymore about another way to do it; I18 they sometimes forget that there are *swhidgets* they can use to do a task. As one can expect, there is a strong correlation between items I13 and I14 ($\rho = .787$, $p < 10^{-30}$), which were both correlated with participants' self-estimated knowledge of iOS ($\rho = .279$ and .217, $p^{\star} = .0043$ and .047 respectively) and interest for technology news ($\rho = .268$ and .308, $p^{\star} = .0012$ and .0002 respectively). There was however no agreement on I19 "I sometimes need to be reminded where are the *swhidgets*".

8.6.5 *General perception of Swhidgets*

INTERACTING WITH SWHIDGETS Participants appreciate interacting with *swhidgets*, reporting that they enjoy manipulating them (I1:

Q1=5, median=6, Q3=7, 85.3% scored 5-7) and that the interface responds in a way that feels satisfactory (I2: Q1=5, median=6, Q3=7, 88.1% scored 5-7). Answers to these two items are well correlated (Spearman's $\rho = .75$, $p < 10^{-25}$), and also correlate with participants' interest in technology news ($\rho = .306$, $p < 10^{-3}$, and $\rho = .231$, $p = .0054$ respectively).

Participants globally rejected the difficulty of manipulating *swhidgets*, at 82.5% for I3 "Swipe gestures are difficult to perform", 64.3% for I4 "the system often misinterprets my swipe gestures", and 62.2% for I5 "the animation of *swhidgets* is sometime unpredictable". There was nonetheless respectively 9.8%, 15.4% and 13.3% of participants who agreed with these statements. These three items can be regarded as measuring the perception of difficulties in the manipulation of *swhidgets* with good internal consistency (Cronbach's $\alpha = 0.88$).

REASONS NOT TO ADOPT SWHIDGETS The main reason reported for not adopting a *swhidget* was *habits* ("It is unadapted to my habits and/or preferences", 38.6% of all evaluations), followed by *context* ("It is unadapted to my contexts of use", 26.6%), *perform* ("It is difficult to perform", 14.5%), *worth* ("It is not worth changing my habits", 11.2%), and *learn* ("It is difficult to learn", 5.4%). Distinguishing the three types of *swhidgets* revealed a statistically significant difference for *context* ($p = .043$), with system *swhidgets* having respectively 1.37, and 1.38 times greater odds than view *swhidgets* ($p = .044$), and item *swhidgets* ($p = .010$). The few open comments regarding reasons not to adopt varied considerably from one respondent to another. As an example, one respondent reported that she had difficulties revealing the Notification Center, as her swipe was often not recognized by the system, which changes her preference of this *swhidget*, whereas as seen above, the statement "the system often misinterprets my swipe gestures" was in majority rejected by respondents. Interestingly, even though it was not asked in the survey, several participants insisted for some *swhidgets* that they did not see any reason not to adopt them (reported by 13 respondents for *flashlight* and 11 for locking orientation through the control panel).

8.6.6 *Satisfaction of basic needs*

Table 8.5 shows the results of the last questions in the final questionnaire of the study, which were inspired by the Self-Determination Theory (See Section 7.4.2) and targeted participants' feelings of Competence (through its two components control and mastery) and Autonomy.

FEELING OF BEING IN CONTROL There was no clear consensus on the two questions related to participants' confidence in using *swhidgets* in a way that match their intents. For I22 "I sometimes worry about triggering *swhidgets* by mistake", 33.6% of participants disagreed and 43.4% agreed with the statement (Q1=3, median=4, Q3=6). For I24 "I swipe when I want to use *swhidgets*, but the commands it reveals are not always the ones I thought", 24.5% disagreed but 39.9% agreed (Q1=4, median=4, Q3=5).

There was however a relative consensus on the two other items in favor of a positive feeling of being in control when manipulating *swhidgets*: agreement with "I use *swhidgets* only when I am sure of what I am doing" (I23: Q1=4, median=5, Q3=6, 59.4% agreed, 13.3% disagreed), and disagreement with "Using *swhidgets* requires some attention and distracts me from my tasks" (I25: Q1=2, median=4, Q3=4, 46.2% disagreeing and 22.4% agreeing).

FEELING OF MASTERY There was a clear consensus among participants for a positive answer to the question "I consider *swhidgets* as advanced techniques" (I27: Q1=4, median=5, Q3=7, 11.2% disagree, 63.6% agree). There was also a clear consensus among participants that they express mastery of *swhidgets* and the interface in general, as manifested by their knowledge with "In general, I know when it is possible to use a *swhidget* to achieve a task I want to do" (I28: Q1=4, median=5, Q3=7, 4.9% disagree, 71.3% agree), and by their performance with "*Swhidgets* are the fastest way for me to do my tasks" (I29: Q1=4, median=6, Q3=7, 7.7% disagree, 66.4% agree). On the other side, there was no consensus on participants' ability to memorize all *swhidgets*: "I cannot remember all these *swhidgets*" (I26: Q1=3, median=4, Q3=5, 30.1% disagree, 39.9% agree).

FEELING OF AUTONOMY Participants expressed a clear consensus on the fact that *swhidgets* enable them to do some tasks with their devices generally or in particular situations: "There are tasks I could not or would not do without *swhidgets*" (I30: Q1=4, median=5, Q3=7, 14.7% disagree, 57.3% agree). "There are situations I could not or would not use my phone to do some task, without *swhidgets*" (I31: Q1=4, median=6, Q3=7, 11.2% disagree, 66.4% agree).

There is also a relative consensus on the fact that a good knowledge of *swhidgets* is a quality that is or should be socially recognized: "Knowing many *swhidgets* is a legitimate source of pride" (I33: Q1=4, median=4, Q3=7, 15.4% disagree, 49.7% agree).

On the other hand, participants expressed different opinions on their interest in learning *swhidgets per se*, as shown by the lack of consensus on "I do not want to remember so many *swhidgets*" (I32: Q1=3, median=4, Q3=5, 30.1% disagree, 33.6% agree).

8.7 DISCUSSION OF THE RESULTS OF THE TWO STUDIES

I reported on the results of the first two studies conducted on current usage of *swhidgets*. The first study conducted in a laboratory setting investigated which *swhidgets* were spontaneously used by participants when prompted to perform certain operations on an iOS device. The second study conducted via an online survey platform, investigated which *swhidgets* users reported to know and use. Combined, our studies provide the following main insights on awareness, usage and discovery of *swhidgets* by middle-aged and technology-friendly smartphone users.

8.7.1 *Awareness of Swhidgets*

Awareness of *swhidgets* was of 64.1% in the laboratory study and 80.2% in the online study. While unsurprisingly lower than awareness of navigation controls, awareness rate of our participants was relatively high, yet slightly different between both studies. One possible explanation of this difference may be the fact that it was *observed* in study 1 and *self-reported* in study 2. Regardless, these results suggest that the integration of *swhidgets* in touch-based operating systems, despite the lack of signifier to inform users of their availability, is well executed and that users are still able to discover their existence.

Interestingly though, awareness rate also happened to vary depending on the operation. Notably, 42.9% and 50% of participants in study 1 and 2, respectively, were not aware of the view *swhidget* for revealing the time at which a SMS was sent in Message. It is of particular interest since there is no real alternative to obtain this information, and that messaging was one of the main activity reported to be conducted by our participants with their smartphones. Unaware participants provided mixed feedbacks about this feature when this item *swhidget* was demonstrated to them, either commenting that they were simply not aware of it where others commented that they never need to know the exact time of a message. Awareness also depends on the type of *swhidgets*, with users significantly more aware of *system* than *item* and *view swhidgets*.

In the end, similarly to hotkeys that are more often learned for frequent commands [Lan+05; Mal+13], it is likely given their complete lack of signifier that users are more aware of *swhidgets* they feel the need to use, which probably motivates their discovery in the first place.

8.7.2 *Usage of (And Reasons Not to Use) Swhidgets*

Usage of *swhidgets* was on average of 56.0% and 66.0%, in study 1 and 2, respectively (once again, usage rate was lower for *swhidgets* than for navigation controls, as one could expect). Overall usage rates are relatively high, especially considering awareness rates as they show that our participants tend to use a high proportion of the *swhidgets* they are aware of (median usage rate of 83.1% in study 2, up to 94.6%). Even more interestingly, we observed that *swhidgets* are extremely often favored over alternative inputs. In study 1, 62.1% of all completed trials were performed using first a *swhidget*, with 36% only for navigation. In study 2, *swhidgets* were preferred on average by 57.8% of participants who knew how to complete the operation, against 39.2% only for navigation controls (*swhidgets* favored over navigation for 5 operations). This result can once again be explained by the fact that *swhidgets* may appear as a *shortcut* alternative to navigation controls in many applications, and as such, are favored over techniques that would take longer or would be more tedious to use. However, proportion of primary or preferred usage once again depends on the operation performed. Typically, in study 2, only 37.5% of participants who completed the *delete e-mail* operation preferred the *swhidget* over the navigation methods. Similarly, in the laboratory study, only 36% preferred the *swhidget* for replying to an e-mail. The main reasons reported were that these operations are often performed right after reading the e-mail, where conventional navigation controls are ready-to-hand, whereas using the *swhidget* would require to come back to the e-mail list. Once again, these results suggests that usage is also likely to be higher for *swhidgets* that appear as shortcut alternative to conventional input methods.

In the laboratory study, 21.4% of participants did not consider that *swhidgets* help them with their everyday tasks. We further investigated the reasons behind this in the online study, where 38.6% of respondents reported that the main reason not to adopt a *swhidget* would be because it is unfit to their habits, and 26.6% because it is unsuitable in their context of use, confirming the assumptions made above about *swhidgets* considered mostly useful as shortcuts. However, it is known that users have difficulties to estimate the actual costs and benefits of switching to a theoretically more efficient alternative [Coc+14; Mal+13]. This was confirmed by our observations, where several users had a wrong perception of the performance gain provided by *swhidgets*. For instance, in the laboratory study, several participants who knew how to delete emails with an item *swhidgets* still preferred to tap on the visible button, because they believe it is faster to select all the emails that they want to delete and then put them into the trashcan together. Although this is only our perception that *swhidgets* would be faster in this case, and that they are indeed designed to be faster, an implication for design would be to increase the visibility of this fact so that the speed difference is more accurately perceived by users.

8.7.3 *Discovery of Swhidgets*

Study 2 highlighted the role of self-exploration of the interface in the discovery of *swhidgets*, where we asked how users discovered each *swhidgets* they were aware of. 42.6% of the answers were that the *swhidget* was discovered by exploring the interface, and 15.9% that it was from accidental activation. These results are in line with the assumptions made by Schramm et al. in their work on performance of hidden toolbars, that users are becoming more familiar with touch-based devices and *swhidgets*, and tend to explore the interface to discover more of them [SGC16].

Schramm et al.'s assumption is also in line with the results collected in the online study regarding the understanding of the physical metaphor of *swhidgets*. While we found no consensus regarding a complete understanding of which objects could be moved to reveal *swhidgets*, respondents clearly reported that they suspect to find *swhidgets* they know at similar location in different applications, which probably encourages interface exploration, and therefore, the discovery of *swhidgets* that way. That being said, our results suggest that knowledge of *swhidgets* transfers only in limited amounts from one application to another, between *swhidgets* that rely on similar interface layouts. For instance, the similitude of participants' knowledge about deleting mails and SMS with *swhidgets* (in both studies), as well as the similitude of PRIMARY usage of *swhidgets* for searching mails and SMS in study 1 are compatible with such a transfer. However, the significant difference between PRIMARY usage of SMS.delete and mail.markAs – which use basically the same kind of *swhidgets* – suggests that the transfer of knowledge does not rely only on the type of *swhidgets* involved, but also on the nature of the task and direction of the swipe gestures (participants of study 2 differed in opinions about expecting different swipe directions).

Finally, the low numbers of *swhidgets* discovered following a demonstration of the input method (4.9%) or remembering seeing someone else using it (3.1%) was however surprising. Indeed, social interaction had been found to play a major role in disseminating useful system capabilities [Per+04]. Similarly to usage of *swhidgets*, it is possible that users have a wrong estimation of the performance benefits of a *swhidget* when witnessing someone using it. In addition, it is possible that our respondents discovered the *swhidgets* months or years before completing the survey, therefore it remains uncertain that participants remembered perfectly how they did discover them. Even though a "I do not remember" option was available, it is likely that users sometimes preferred to assume how they discovered them rather than answering they did not remember. Finally, it is also possible that participants who learned about a *swhidget* by watching someone do it still considered that they discovered it "by exploring the interface" when they tried to reproduce what they saw on their own device.

8.7.4 *Future work*

While informative, our studies could not explore all smartphone ecosystems, as well as focus on specific populations. As such, it leaves interesting opportunities for future work.

First, we chose to focus our studies on the iOS ecosystem for several reasons: it is a coherent and consistent OS, whereas Android allows manufacturer to create software overlays that change the look and feel of the OS. Because of this, default applications, advanced input methods (including *swhidgets*), hardware inputs, etc. are likely to differ much more between Android devices than between iOS devices. That being said, Apple recent smartphones and tablets are not equipped with a hardware home button anymore. The home feature is now performed with an upward swipe from the bottom of the display, which interferes with the previously control panel *swhidget*. Consequently, on these devices, downward swipes from the top edge are now used both for revealing Control Panel and Notification Center, depending on which corner they are performed from. Our studies did not investigate these novel gesture mappings for this system *swhidget*, but remain valid for all other *swhidgets* tested. Future work should therefore focus on investigating *swhidgets* on other platforms, and possibly investigating the discovery and awareness of the localized *swhidgets* on recent iOS devices.

Second, even though we recruited participants of various ages (18 to 75 years old), both of our studies were conducted in majority with middle-aged adults familiarized with novel technologies. As such, it is likely that our participants, while representative of many smartphone users, correspond to a relatively high limit in term of knowledge of interactive capabilities of their device and interest in latest technologies. It is unlikely that this limited pool of participants would diminish in any way the validity of our results, since many smartphone users are likely to correspond to our stereotypical participant. Yet, future work should investigate different populations. While we informally tested the effect of age on awareness and usage of *swhidgets* but did not find any strong effect (possibly due to the low number of participants over 65), we still believe that future work should investigate usage of input features on smartphones with different age groups. Indeed, seniors and teenagers have significantly different background with technologies and it would be interesting to investigate how these difference in technology exposure may impact.

8.7.5 *Dataset*

Data collected for this work has been anonymized and is available to the community through our institution address[10]. We collected a significant amount of information in order to compare interaction with *swhidgets* with navigation and voice control, but focused our analysis on *swhidgets* which is the core of this thesis. That being said, we remind that this data set contains information regarding many of the points discussed (typically regarding discovery, reasons to adopt, etc.) for navigation and voice controls as well.

10 http://ns.inria.fr/discovery/swhidgets/

Part V

PERSPECTIVE AND CONCLUSION

9 PERSPECTIVES AND CONCLUSION

In Section 9.1, I will discuss the perspectives for future works following the studies I did in this thesis and the theoretical models that I presented. In Section 9.2, I will shortly present an ongoing design work to improve the discovery of *swhidgets* in order to show how the theoretical models that I have introduced can point to unexplored areas of the design space of *swhidgets*. Finally, I will conclude this thesis in Section 9.3.

9.1 FROM RESULTS BACK TO MODELS

In this dissertation, for the clarity of the discourse and because it is a first work on *swhidgets*, I chose to start with the definitions of affordances, signifiers, and signifier-less designs, so that I could introduce a model of discovery and proficiency improvement in Chapter 7, which would bring useful concepts for the design of the studies presented in Chapter 8 as well as for the discussion of their results. In other words, I presented the concepts that allowed to define signifier-less designs and to devise experiments beyond the simple measure of what proportion of the population knew about them. In doing so, my goal was to provide foundations not only for my own work but also for future works on *swhidgets* and signifier-less designs. I will now discuss how these concepts may be the building blocks of further studies.

Assuming they generalize to the whole population of iOS users, the results of the two experiments bring a few observations: users have a moderately high awareness of iOS *swhidgets*, use most of those they are aware of, and often prefer *swhidgets* over alternative input methods when their performance benefits are clear. While relying on a signifier-less design, *swhidgets* are relatively well discovered by users through exploration of the interface.

These results need to be confirmed by other experiments and extended to larger populations of touchscreen-based devices, but they provide early observations that raise new questions. I will address some of these questions in the next sections as perspective for future research, discussing how the model I proposed in Chapter 7 as well as the notions of affordances, signifiers, and semiotics can bring light to the questions or suggest ways to approach them.

9.1.1 *Transfer of Knowledge*

What role plays the transfer of knowledge between applications in the discovery of swhidgets in particular, and interaction techniques in general?

This question was identified as one of the general research questions related to *swhidgets* under the concern of *consistency with other interactions* (Section 2.6.1), and as one possible answer to a question investigated in this thesis, *how can users know swhidgets despite their lack of signifiers?* (Section 2.7.3). The two studies I conducted during this thesis were thus designed to approach this question but did not focus on it, and provided mixed results: On one hand, transfer of knowledge seems to be a common way to discover *swhidgets*. On the other hand, it does not seem to depend only on the design of the interface but also on users' ability to relate the two applications – which was apparently not the case for the conversation list in the Messages application and the thread list in the Mail application, despite similar layouts.

These studies thus call for additional experiments addressing this question with a narrower focus, to determine aspects of signifier-less designs that affect user's ability to transfer their knowledge to other contexts of use. However, experiments with a narrow focus need precise hypotheses, which require detailed models and theories. I will thus now review how the notions I discussed in this thesis can suggest topics of studies investigating the phenomenon of discovering signifier-less designs with a transfer of knowledge.

The analysis of signifiers that I have done in the framework of Peirce's semiotics (see Chapter 5) suggests that transfer of knowledge can sometime be modeled as recognizing a specific layout or other features of the interface and treat them as iconic signifiers, which objects are the affordances to be discovered – or, if not directly the affordances, at least generic interface patterns that include them, such as "a list view with *item swhidgets*" (See also Section 7.3.1).

In this model, users first need to understand that there is a rule in the interface connecting the affordance to the layout or feature that serves as a signifier. Such a rule would be knowledge at the *UI level* (Section 7.2.1). It can be taught explicitly by tutorials, or will more likely be discovered by users through an inductive process – essentially by noticing the pattern after having been exposed to some number of its instances. Is there something in the pattern that allows users to notice it more easily? Could the noticeability of the pattern be improved by means of careful design? To notice the pattern, users need to make generalizations at the *UI level* from pieces of knowledge at the *command level* (the instances of the pattern) in a form of dependency across levels (see Section 7.2.3), which may first need to acknowledge the relevance of these pieces of knowledge (see Section 7.5.2).

Are design means such as interface layouts the main factor affecting the likeliness of users noticing the pattern, or do users actually notice patterns that encompass design means but also knowledge improvement aspects? For instance, users might recognize that a problem they have is similar to one they already had, and thus look for a similar solution. In that case, the similarity of design means could help recognize the similarity of problem contexts, but the design means may not be recognized as similar if the user was not facing a similar problem. Such a phenomenon could explain why there seems to be little transfer of knowledge between the Mail *item swhidgets* and the Message ones, since deleting a mail or thread might be a frequent operation while deleting a full conversation with a contact might only happen rarely. If a type of problem is frequent enough, users may form strategies to deal with it, and shift from what Rasmussen describes as knowledge-based behaviors treating design means as symbols (see Section 7.3.1) to rule-based behaviors treating design means as signs (with strategies such as "If I need to delete an item in a list, test if *item swhidgets* are available").

Beside the noticeabilty of the pattern, effects of (a lack of) reliability on their adoption also need to be studied – i. e., will the recognized design mean of the pattern always be associated with the affordances it conveys? Indeed, comments from the participants of the two studies I did revealed that for some users, a lack of consistency in the use of *swhidgets* across applications was a reason not to adopt *swhidgets*. This behavior can be explained in the framework that I presented in Chapter 7, and especially in Section 7.5.1 about the interactions between knowledge and performance-oriented motivations. Indeed, when users believe a *swhidget*-hinting pattern lacks of consistency, they may conclude they will have to learn on a case by case basis which applications provide *swhidgets* and which ones do not. In other words, the cost of learning to use the interface in a way that relies on *swhidgets* may seem higher for users, which can lead them to discard *swhidgets* as globally demanding too much effort to learn. The results of the online study (See Section 8.6.4) suggest that difficulty of learning *swhidgets* can be a rare (5.4%) reason not to adopt them, despite overall results suggesting *swhidgets* are well understood by users and recognized across applications.

On the other hand, this negative effect of the lack of consistency on adoption of *swhidgets* can also not happen, being replaced with a positive effect on user experience when users decide to playfully explore the diverse applications to find which ones provide *swhidgets* (results of the online study suggest users tend to play with a *swhidget* after they have discovered it). In that case, the discovery of a *swhidget* can be considered as a random reward, and there is a great amount of literature in Psychology about the role of such rewards in conditioning human and animal behavior – in the field of video game design, it is often referred to by designers and critics with expressions like "Skinner boxes" or "loot boxes" (e. g., [Sci11; Blo10; Ext12; LWW12]).

As a conclusion on the topic of transfer of knowledge, I would like to first recall how important the topic is for HCI in general, but even more for the study of signifier-less designs like *swhidgets*, since their discovery rely more heavily on external sources of knowledge (22.9% in the online survey, see Section 8.6.4) and transfer of knowledge from other areas of the interface (8.8%). However, as I have discussed in this section, the topic is complex and there are many aspects of the interaction and interface that affect transfers of knowledge, such as the recognize-ability and notice-ability of user interface patterns as signifiers, analogical reasoning in goal-oriented interaction problem resolution, user strategies, consistency of patterns across applications and views, users' ability to correctly evaluate this consistency, playful user behaviors and behavior conditioning. The model that I have presented in Chapter 7 and the concepts of affordances and signifiers are useful tools to discuss these questions theoretically and to identify precise hypotheses on the mechanisms of transfer of knowledge that can be tested experimentally. In addition, integrating the above list of concerns into more pragmatic research tools like questionnaires or usability evaluation methods is an area of research that, in my mind, deserves more attention. I created the questionnaire for the online study with these concerns in mind, knowing that, having not been formally validated before the study, the questionnaire could only suggest user behavior trends rather than provide conclusive observations – which was acceptable because it was used in a preliminary study that had no such high-level objectives and also intended to test the usefulness of such a questionnaire as a research tool. In my mind, this questionnaire did bring interesting results and should be refined to develop a well-validated research tool that could be instrumented in other studies.

9.1.2 *Hedonic aspects of Swhidgets*

Can pragmatic benefits justify to not use signifiers in a design?

The concept of transfer of knowledge – and more generally the idea of sources of knowledge that are outside the application concerned by a piece of knowledge – helps understanding how users can learn about *swhidgets* despite their lack of signifiers. It thus explains how this design can possibly work for users, providing a possible answer to my research question *how can users know swhidgets despite their lack of signifiers?* (Section 2.7.3), but it does not explain the benefits of not including signifiers in the design of *swhidgets*, which was the topic of my last my research question (Section 2.7.4).

I have discussed in this thesis a few (rare) plausible reasons for not providing signifiers that rely on pragmatical expected benefits for users, such as making interfaces easier to read and improving the consistency of swipe-based interaction techniques in the system. While such benefits cannot be dismissed, it can also be doubted that they are strong enough to justify not providing signifiers at all (rather than simply minimizing their visual salience). Indeed, it seems that these benefits would be overcome by the improved discoverability of useful *swhidgets* that could be obtained by adding signifiers – although validating this hypothesis would require a dedicated study, which would itself require means to quantify these benefits and measure them.

I have thus developed during this thesis the conviction that to justify the signifier-less design of *swhidgets*, which is questionable from the point of view of pragmatics, considerations for the hedonic aspects of interaction (see Section 7.4.3) were at least equally important than considerations for the pragmatic aspects. Hassenzahl comments that pragmatic and hedonic concerns can be in conflict: "Strikingly, taking the need for novelty and change into account might unavoidably imply a reduction of usability. Usability and joy of use might be partially incompatible, because the former requires consistency and simplicity, whereas the latter requires surprise and a certain amount of complexity" [HBB01]. Although research on UX is conscious of this possible trade-off between pragmatics and hedonic qualities of an interface or product, I believe *swhidgets* are a good case study to better understand the consequences of favoring hedonic qualities over pragmatic ones. In the remainder of this section, I discuss how this general research question on UX-oriented designs can be informed by the study of *swhidgets*.

What types of benefits can we expect, based on the hedonic aspects of user experience?

Discovery of advanced interaction techniques can create a strong stimulation, and as we have seen in the previous section, such a stimulation can even incite users to engage into playful searches for *swhidgets* – finding hidden things is after all the core task of many games and playful activities, from hide-and-seek to adventure video games. The stimulation created by a discovery can also reinforce the emotional attachment to a device or system: our relation to things become more personal when we have discovered one of their aspects that was hidden.

Conversely, as the online study showed for *swhidgets* (see Section 8.6.6), we easily get proud of our knowledge of "tricks" to interact with electronic devices or everyday objects. We also like to share it with our relatives. This knowledge is thus instrumented to satisfy our innate needs for both Competence and Relatedness (see Section 7.4.2 about the Self-Determination Theory).

The results of the online study (Section 8.6.6) also show that participants attributed to *swhidgets* their ability to do some tasks with their device or to use the device in specific situations: the discovery of new affordances can indeed bring new opportunities to change the way we organize our lives so that it better matches with our values and who we want to be, satisfying our needs for Autonomy – although the relationship between theses higher goals and the affordance discovered might be quite indirect. For instance, one could realize that with Mail's *item swhidgets*, it becomes actually not too unpractical to manage a great amount of mails on the small screen of the phone, and could decide to do this activity during the commute to work in subway instead of doing it in the office, saving time either to focus on other work at the office, or to leave for work later and enjoy sharing a breakfast with beloved ones – or simply to distract from the unpleasant aspects of the commute.

Does accounting for the hedonic aspects of user experience require a shift in how we understand the roles of signifiers and discovery of affordances?

Historically, there has been a transition from computers as professional tools evaluated on their capacity to effectively increase workers' productivity at reasonable costs, to the personal and always-available smartphones that we have today and use not only for work but also for many other aspects of our lives. This transition naturally causes a shift in the considerations about what are the benefits of a device or interaction technique, from pragmatic concerns to hedonic ones: it is less about what I can do with the computer and more about how it supports me in being who I want to be. In this context, the focus of design may shift from designing products that communicate clearly what they are for and how to use them (typically using signifiers) to designing products that satisfy consumers' needs for self-improvement.

Discovering new affordances and developing mastery have always been a concern of user interface design, but it was mostly valued for its consequences on user performance as extrinsic reward rather than for the user experience of doing so, valued for itself as an intrinsic reward. Providing users with opportunities to experience the discovery of new affordances and of developing mastery might have become a new design concern that could justify the signifier-less design of *swhidgets*.

Such an approach to design suggests new ways to formulate usability concerns. For instance, a question like "is this affordance signified clearly to the user?" might be replaced by the more general question "does the design invite users to engage in an exploration that will lead them to discover this affordance?" Some designers may already have started more or less consciously to ask such questions, but the consequences of such approaches on users and on the quality of designed products are hard to foresee, and should thus be studied quickly.

There seems to be currently little evidence that user interfaces are designed with such a question in mind. It has however become frequent in the design of video games, as the player experience of discovering an unknown world and the way it works is a primary focus of game design and there are more opportunities to orient the player experience through the design of a world and the challenges it provides than for a user interface. As the studies of video game design tend to be more common in HCI, it is likely that HCI practitioners and researchers will get inspiration from video game design philosophies and focus more on this type of questions in the future.

How can we study such approaches to design?

Investigating such approaches to design requires new conceptual tools (both for design and HCI research) [HBB01]. Although there already exist conceptual models such as Hassenzahl's model of UX (see Section 7.4.3) and evaluation tools like the AttrakDiff questionnaire [HBK03; Lal+15], such models and tools are not always adapted to studies on user adoption of interaction techniques. For instance, the AttrakDiff questionnaire evaluates how users perceive a product or interface and is usually used to investigate users' perception of a product after their first contact with it. Although the questionnaire can be instrumented to investigate how this perception evolves in time with repeated measures [Kar+09], it is not made to investigate how a design contributes in the long term to how a product fits in its users' lives and satisfies their evolving needs. It can however bring important information, and I plan to use it in a study under preparation, in order to complement and confirm some aspects of the interaction with *swhidgets* that have been suggested by the online study, such as the fact that participants enjoyed interacting with *swhidgets*.

I have myself attempted to integrate questions inspired by the Self-Determination Theory at the end of the online study final questionnaire (see Table 8.5), although constraints on the length of the study did not allow me to go as deep as would have been necessary. I also faced the issue that existing SDT-based questionnaires – such as the *Intrinsic Motivation Inventory* (IMI) [RRP06] – tend to evaluate participants' satisfaction of needs during an activity without targeting the aspects of the activity (or of the design of artifacts involved in the activity) that are believed to cause such satisfaction of needs. Although there exist such targeted questionnaires in specific areas of design – such as the *Player Experience of Need Satisfaction* (PENS) questionnaire for video games with specific questions on topics such as presence in the game world and intuitive controls [RRP06] – I am not aware of any questionnaire that would be suitable for the analysis of *swhidgets*. Since some of the questions I integrated in the questionnaire brought clear and interesting results despite their limitations, I firmly believe that the creation and validation of a SDT-based questionnaire to evaluate signifier-less designs is an important research perspective.

What are the consequences of designing for the hedonic aspects of user experience?

A perspective for future works around this approach of design is to study its implications from the point of view of ethics. When usability is decreased to create opportunities for stimulation and fulfill users' need for personal growth, don't we artificially instrument a product to satisfy a basic need instead of truly giving users a way to self-improve? Indeed, the increase of knowledge concerns the knowledge of the product itself instead of concerning knowledge that may be more important to the user. If the design focus is more concerned by "who I want to be" than by "what I can do with the product", maybe the answer to the question "who do I want to be?" should not be "an expert in this particular product". This ethical question is connected to Hassenzahl's distinction between "experiences mediated through a product versus experience of a product" [Has18].

9.2 IMPROVING SWHIDGETS

9.2.1 *Revealing Swhidgets with Animated Transitions*

From the analysis of *swhidgets* conducted in Chapter 2 and the results of the experiment presented in Chapter 8, I derive the need to improve on some aspects of the design of *swhidgets* with the following considerations:

1. Obviously, improved designs should foster the discoverability of *swhidgets*;

2. I do not want to change the affordances and interactive behaviors of *swhidgets* themselves, nor to create new interaction patterns for users. My goal is therefore not to create new interactions or to redesign interactions with *swhidgets*,[1] but to build upon their design;

3. I want to preserve the benefits brought by the absence of signifiers, and will thus avoid the obvious solution of adding permanent signifiers. In other words, I stay in the paradigm of signifier-less designs defined in Chapter 6.

My approach is to explore the design space for sources of knowledge that have the smallest possible distance (as defined in Section 7.7.1)

1 In a first attempt in the direction of the work on animation exposed later in this section, I thought about simulating gravity so that the *item swhidgets* in a list view like Mail's thread list would be revealed by tilting slightly the device (the items of the list sliding sideways a little under the effect of gravity). However, informal tests with users quickly revealed they thought it was a new way to interact with the *swhidgets* rather than a way to inform about their presence, and spent a considerable time trying to exploit this new affordance.

without being direct signifiers (as defined in Section 6.1), and that foster discovery of *swhidgets* out of problem-solving contexts (as defined in Section 7.7.2).

To keep a small *distance* without using direct signifiers, designers may use temporary signifiers that would be displayed only when 1) the *swhidgets* concerned by these signifiers are available (or about to be) and potentially useful for users,[2] but 2) the user is unlikely to interact with these *swhidgets* at that moment, and 3) the temporary signifiers would not distract users in their tasks. Points 1 and 2 precise the idea of a small distance while point 3 extends the concerns of an uncluttered interface to temporary signifiers.

I therefore propose to use *animated transitions* as a way to give *swhidgets*-unaware users a chance to discover the *swhidgets* as they interact with the device. The goal is to raise users' awareness of interface features they do not know and stimulate their curiosity, in order to motivate them to perform the sliding gestures corresponding to these *swhidgets*. Animations are already used in many places in mobile touch-based systems and applications as a way to "increase understanding".[3] Their use is believed to facilitate the understanding of changes [CU93] and has been repeatedly demonstrated as helping to understand the spatial relationship between views (see [Che+10] for a brief review).

9.2.2 *Considerations for the Design of Animated Transitions*

ELIGIBLE TRANSITIONS Animated transitions are often used when a new view is *loaded* (a UI view replaces another one), or *updated* (an element in this view is added, removed, or otherwise changed at the user's or system's initiative). Since *swhidgets* are usually hidden outside of the view's boundary, I believe that user-triggered animated transitions between views could expose the space around the views and let users glimpse at the *swhidgets* lying there before they get hidden. I advocate to focus on animated transitions triggered by user events, as opposed to transitions triggered by external events that change the state of an element (e.g., when a message transitions from the *sent* state to the *read by recipient* state). Indeed, in addition to the

2 This point excludes the situation in which the user triggers a command provided by a *swhidget* through alternative methods of activation, which is a case similar to the hotkey hints provided in menus, as discussed in Section 6.2.3. There is however room for using animations in such situations too. For instance, deleting mails with the Edit button rather than *swhidgets* already uses an animation to remove the corresponding list items. This animation is also used when deleting emails with an *item swhidget*, although in that case the sws are still displayed in the item removed. The animation of item removal with the Edit button could be extended to show the *swhidget* as if they had been used, to expose these *swhidgets* and suggest the appropriate gesture to reveal them.

3 https://developer.apple.com/design/human-interface-guidelines/ios/visual-design/animation/

risk that users interact with the animated elements by mistake because they did not see them move or could not stop an engaged gesture in time, understanding such animations may also have a higher cognitive load because it requires users to understand why something changes in a first place. Moreover, such on-update animations will most often be redundant with other animated transitions between views.

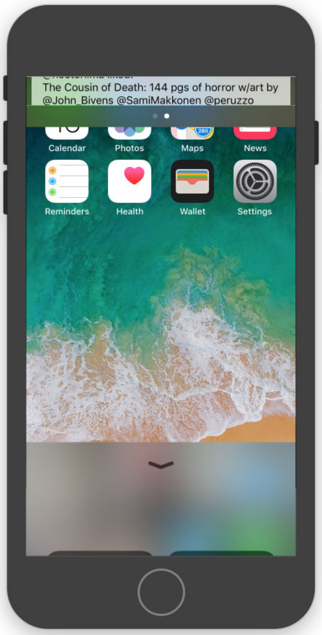
CONFLICTING PURPOSES Animated transitions serve a primary purpose, which can broadly be described as "helping the user understanding what is happening". In the case of animated view transitions, it is "helping the user understanding that the interface is switching to a new view". When using animated transitions to expose *swhidgets* and give users a chance to discover them, we introduce a secondary purpose of the animation, and we should then make sure it does not take precedence over the primary one or impede it. A good rule of thumb seems therefore to minimize the amount of information provided to the user for this secondary purpose only, by both reducing the visual salience of the *swhidgets* during the animation, and by avoiding redundancy (multiple animations exposing the same widgets). Indeed, as we have seen in Section 7.3.1, users' informational needs and abilities to process the information conveyed by the animation evolve as they gain more experience with the interface, and can shift from the primary purpose of the animation to the secondary one (although they can also learn to discard information that seem irrelevant for them).

COMPLEXITY OF ANIMATIONS When not used judiciously though, animations and motion effects can potentially be distractive or slowing down interaction [BWC03], which is why we should constrain new designs to animations that are relatively short and do not involve complex movement patterns (elements moving in different directions simultaneously, non-linear trajectories, etc.)

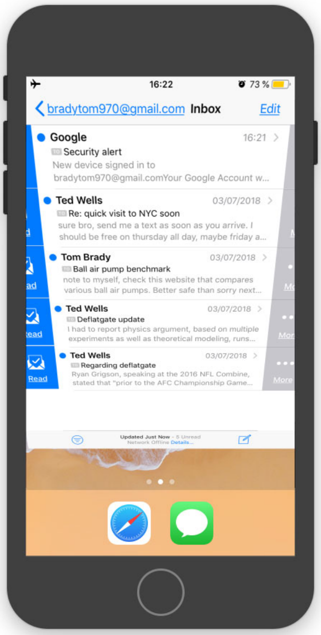
INTEGRATION IN USERS' WORKFLOWS Attention should also be payed not to impact users' workflows. For instance, in some applications the most important content can be expected to be in certain areas of the view (e.g., newest mails are on top of the list in the Mail application), and I then consider animations where contents move less in these areas or stop moving earlier than in other areas, so that users can read them as soon as possible.

9.2.3 *Proposed Animated Transitions*

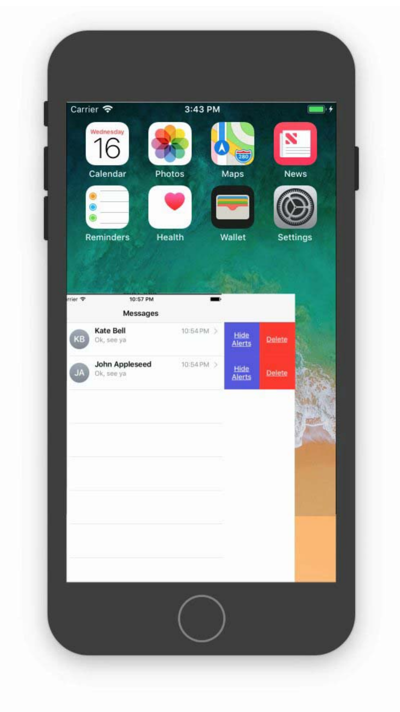
The interface of touch-based systems like iOS and Android already makes great use of animated transitions. To leverage the existing transitions in these systems to help users discovering *swhidgets*, we need to distinguish the different types of animations they use. For



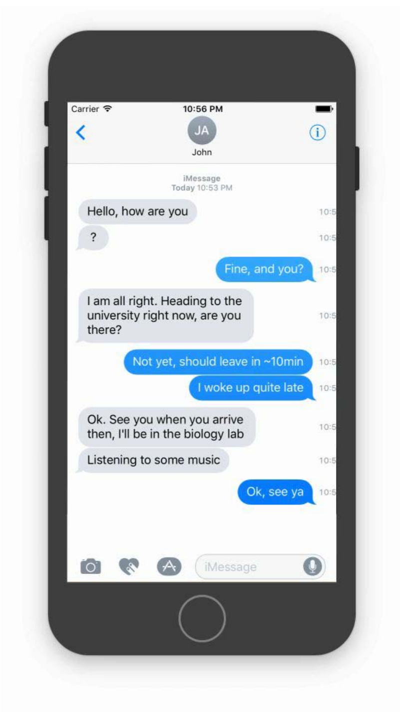
(a) Slide-in Panels



(b) Perspective in Mailbox



(c) Scale-in SMS list



(d) Slide-in SMS Discussion

Figure 9.1: Examples of animated transitions that can facilitate the discovery of *swhidgets*.

instance, iOS uses animations based on three types of geometrical transformations: Scale variations (when opening an application or unlocking the screen), 2D translations (when navigating within an application) and perspective rotations (when selecting an element in a *carousel* view, for instance to change the current tab in Safari). I propose the following three types of slightly modified versions of these transformations in order to increase the discoverability of *swhidgets*:

- *Scale-in* progressively scales the new view but differs from the iOS built-in transition in that it does not clip the view's content during the transition, thus giving the opportunity to see during the animation the *swhidgets* that will end hidden outside the screen (Figure 9.1c).
- *Slide-in* applies additional 2D translations to the new view and *swhidgets* during the animated transition to let users see behind which bezel of the display the *swhidgets* will finally hide (Figures 9.1a and 9.1d).
- *Perspective* applies a perspective rotation but, similarly to scale-in, does not clip the view's content to reveal the *swhidgets* during the animation (Figure 9.1b).

Scale-in, *slide-in* and *perspective* could be instantiated for all the tasks that were tested in the first study (listed in Table 8.1) whenever possible (scale-in and perspective-based cannot be applied to the System unlock scenario without significantly changing the metaphor currently employed in iOS). I detail only a subset of these instances below.

SYSTEM UNLOCK SLIDE-IN On unlock, iOS currently translates-out the lock-screen above the top edge of the display, which is extremely rapid (and hard to perceive). By simply sliding-out the Control and Information panels at the same time, but at a relatively lower speed, users could notice the panels progressively translating below the top and bottom phone bezels (Figure 9.1a). Using a simple easing-out timing function, the information and control panels quickly move in the vicinity of the edges of the display, thus not hiding the main view, but slowing down before getting completely out of view so that users can still perceive this motion.

MAIL APP PERSPECTIVE When an application is launched from the home screen, iOS scales-in the application's view from its icon to full-screen. By changing this animation for a perspective rotation without clipping the view's content at the border of the email list (Figure 9.1b), users have the opportunity to see the *item swhidgets* located on both the left and right sides of each e-mail in the list. The small perspective distortion at the top of the screen also allows them to start reading the first (most recent) mail before the transition ends.

MESSAGE APP SCALE-IN AND SLIDE-IN iOS scales the view of an application when it is launched, but uses translations to animate view transitions inside a same application. I propose to apply scale-in to the discussion list when the Message application is launched, so that users can notice the associated *item swhidgets* (Figure 9.1c). I also propose to increase the amplitude of the existing slide-in transition when a discussion is opened, to create a kind of "overshoot" that temporarily reveals the message time *view swhidgets* (Figure 9.1d), before they slowly slide back past the right edge of the display – giving users an opportunity to discover these *view swhidgets* and suggesting the leftward sliding gesture that brings them back into the view.

9.2.4 *Testing the animations*

Mock-ups of the proposed animations have been realized in HTML5 and JavaScript, along with *swhidgets* that reproduce as precisely as possible the iOS lock screen interface and Mail and Message applications – although the data displayed are not real data and the commands triggered with the *swhidgets* have no actual effect.

While these mock-ups allow to test the animation and fine-tune their parameters, an intended use was also to actually compare the new animations with the current design of *swhidgets*, in controlled studies. These studies could unfortunately not be realized during my PhD due to a lack of time, but also due to methodology issues. I thus plan to experimentally validate the efficiency of these prototypes in fostering the discovery of *swhidgets* as future work.

Such a validation is a however difficult task, as no ready-to use validation procedure has been proposed in the literature to test the discovery of features from commercial systems. Indeed, the discovery of system features and input methods is something that should be assessed on relatively long term and can hardly be evaluated with a laboratory experiment. It requires fully functional applications similar to the ones they are compared to. Moreover, it would require to first measure the level of knowledge of participants regarding the tested input methods, while avoiding that participants understand what the experiment is about to prevent any demand characteristics issue [Orn09]. It might alternatively require to find participants that have currently no experience with touch-based devices, but such people are getting rare and might not be representative of other users.

9.3 CONCLUSION

During this thesis, I studied the question of why designers would create interfaces that do not clearly expose some of their interaction possibilities, taking the case of *swhidgets* as an example and focus of inquiry. My contributions to this research are:

- An analysis of the fundamental notions required to define precisely this research question, including affordances, signifiers and semiotics.
- A definition of *signifier-less designs* and the observation that this type of design is common in user interfaces and deserves more attention from HCI research.
- A model of user discovery and adoption of interaction techniques, relying on the categorizations of three dimensions and their relationships: user's current knowledge and skills, user's motivations, and the design means of informing users provided by the interface.
- The notions of Sources of Knowledge and Degrees of Knowledge derived from this model, that can be instrumented in experiments.
- Two studies on *swhidgets* that investigated how well users known them, how they discovered them, how they feel about them, and how they integrate them in the way they think about the interaction. These studies revealed that *swhidgets* were globally appreciated and relatively well known by users, although there is still room for improvement, notably for some specific *swhidgets*.

These contributions are however limited in that the object of this study – i.e., interfaces that do not clearly expose some of their interaction possibilities – has traditionally been seen in HCI as a failure of design that was acceptable only for interaction possibilities targeting expert users. I therefore faced a lack of conceptual tools to study such designs beyond denouncing their flaws. As such, an important aspect of the work done during my PhD has been to identify and unify the conceptual tools needed for the study of *swhidgets* and *signifier-less designs*. As a result from this focus on identifying conceptual tools and research questions, the studies that I have done only provide preliminary and incomplete answers to the main research question of this PhD and to the question of the suitability of the conceptual tools that I have identified for conducting such a research. Further research is required to confirm these preliminary results, to extend them to other categories of systems, users, and signifier-less designs, and to better understand the design space of *swhidgets*. The research presented in this PhD however opens a few new perspectives for original research, as I have discussed at the beginning of this chapter.

Part VI

APPENDIX

J FIRST QUESTIONNAIRE OF THE LABORATORY STUDY

Participant ID 2018

Welcome and thank you for taking this test with us!

Please tell us about you:

- What is your age:

- What is your gender:

❑ man	❑ woman	❑ other/don't want to answer

- What is your education level?

- What is your professional domain and current activity?

- Have you worked in a different domain before? If so, in which domain(s)?

- What is your preferred mobile phone operating system?

❑ Android	❑ Windows
❑ iOS	❑ Others (please precise)
❑ No preference	

- How familiar are you with the following mobile phone systems?

System	not familiar				very familiar
❑ Android, ________________	1	2	3	4	5
❑ iOS ____________________	1	2	3	4	5
❑ Windows ________________	1	2	3	4	5
❑ Others (please precise) __________	1	2	3	4	5
❑ Others (please precise) __________	1	2	3	4	5

- Do you also own a tablet?

❑ Yes	❑ No

- If you do, what is the system of your tablet?

❑ Android	❑ Windows
❑ iOS	❑ Others (please precise)

- How familiar are you with the following systems on tablets?

System	not familiar				very familiar
❑ Android ________________	1	2	3	4	5
❑ iOS ____________________	1	2	3	4	5
❑ Windows _______________	1	2	3	4	5
❑ Others (please precise) ________	1	2	3	4	5

- What activities do you perform on your touch-based devices?

❑ accepting phone calls	❑ drawing
❑ taking pictures/videos	❑ calculating
❑ writing/taking note	❑ programming
❑ recording voice/singing	❑ listening to music
❑ checking mails	❑ watching tv/video
❑ browsing the web	❑ text messaging
❑ chatting	❑ video conferencing
❑ playing games	❑ lifelogging
❑ other activities	

- How much time do you spend daily on your touch-based devices?

❑ within 30 minutes	❑ between 3 to 5 hours
❑ between 30 mins to 1 hr	❑ more than 5 hours
❑ between 1 to 3 hours	

Just a few last questions!

We are almost there!

- How many smartphones did you own since 2008?

- Which platform and which version were they running on?

- Do you consider yourself as a person who is highly interested in new technology ?

No, I don't		I don't know		Yes, I do
1	2	3	4	5

- Do these statements describe you?

Description	Totally disagree				Totally agree
• I feel that new technology improves my life	1	2	3	4	5
• I always want to know the latest news on technology	1	2	3	4	5
• I have close friends who keep me informed on new technology	1	2	3	4	5

• I am always the one who tells others about the latest technology news	1	2	3	4	5
• I frequently search for news on technology	1	2	3	4	5
• I follow technical web blogs	1	2	3	4	5
• I always want to have the latest version of devices	1	2	3	4	5
• I sometime support innovative projects	1	2	3	4	5
• I am willing to try on all types of upcoming applications/devices	1	2	3	4	5
• When I get a new device/feature, I try/check almost every new feature	1	2	3	4	5
• In discussions, talking about technology is the best way to get my attention	1	2	3	4	5

K SECOND QUESTIONNAIRE OF THE LABORATORY STUDY

Participant ID 2018

This is the final section of the interview!

Congratulations for reaching so far :-)
Here are some questions that require some imagination from you:

- Do you think the existing hidden controls help you on your daily tasks?

☐ Yes ☐ No

- Where does your knowledge of these hidden features come from? (Check all items that apply)

- From the Internet
  - ☐ From bloggers' reviews
  - ☐ From video tutorials
  - ☐ From a forum
  - ☐ Other : ____________________
- From other people
  - ☐ I asked for help from professionals
  - ☐ I heard people talking about it
  - ☐ Other : ____________________
- From friends
  - ☐ I have knowledgeable friends who can teach me
  - ☐ My friends were talking about it
  - ☐ Other : ____________________
- From the device itself
  - ☐ I discovered it myself by searching the feature in the interface
  - ☐ I discovered it myself by randomly exploring the device
  - ☐ I discovered it myself by accident

- ☐ Other : ____
- From another system
  - ☐ I can guess it because I use another system ___________
  - ☐ I can guess it because it's just opposite from Android
- From the application itself
  - ☐ I just discovered it on my own by exploring
  - ☐ I discovered it on my own by accident
  - ☐ I discovered it with help from an icon or other hint
  - ☐ Other : ____
- During this interview
  - ☐ By myself (I observed it on my own)
  - ☐ From the investigator (Nicole taught you)
  - ☐ Other : ____

- If you are a designer interested in adding this kind of hidden features to your design, please tell us:
  - on what platform/device?
  - with which kind of application/software?
  - what kind of gesture is used to reveal the feature?

Thank you so much for your participation!

BIBLIOGRAPHY

Alzayat, Ayman, Mark Hancock, and Miguel A. Nacenta (2017). "Measuring Readiness-to-Hand through Differences in Attention to the Task vs. Attention to the Tool." In: *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. ISS '17. Brighton, United Kingdom: Association for Computing Machinery, pp. 42–51. ISBN: 9781450346917. DOI: 10.1145/3132272.3134114.

Andersen, Peter Bøgh (1991). *A Theory of Computer Semiotics: Semiotic Approaches to Construction and Assessment of Computer Systems*. USA: Cambridge University Press. ISBN: 0521393361.

— (Dec. 2001). "What Semiotics Can and Cannot Do for HCI." In: *Know.-Based Syst.* 14.8, pp. 419–424. ISSN: 0950-7051. DOI: 10.1016/S0950-7051(01)00134-4.

Apple (2019). *Apple Human Interface Guidelines*. Retrieved July 8, 2019. URL: https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/.

Avery, Jeff and Edward Lank (2016). "Surveying Expert-Level Gesture Use and Adoption on Multi-Touch Tablets." In: *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. DIS '16. Brisbane, QLD, Australia: ACM, pp. 577–581. ISBN: 978-1-4503-4031-1. DOI: 10.1145/2901790.2901895.

Bærentsen, Klaus B. and Johan Trettvik (2002). "An Activity Theory Approach to Affordance." In: *Proceedings of the Second Nordic Conference on Human-Computer Interaction*. NordiCHI '02. Aarhus, Denmark: Association for Computing Machinery, pp. 51–60. ISBN: 1581136161. DOI: 10.1145/572020.572028.

Bailly, Gilles, Emmanouil Giannisakis, Marion Morel, and Catherine Achard (2018). "Characterize the Transition from Menus to Hotkeys." In: *Proceedings of the 30th Conference on l'Interaction Homme-Machine*. IHM '18. Brest, France: Association for Computing Machinery, pp. 30–41. ISBN: 9781450360784. DOI: 10.1145/3286689.3286699.

Barr, Pippin, Robert Biddle, and James Noble (2004). "A Semiotic Model of User-Interface Metaphor." In: *Virtual, Distributed and Flexible Organisations: Studies in Organisational Semiotics*. Ed. by Kecheng Liu. Dordrecht: Springer Netherlands, pp. 189–215. ISBN: 978-1-4020-2162-6. DOI: 10.1007/1-4020-2162-3_13.

Bartram, Lyn, Colin Ware, and Tom Calvert (2003). "Moticons: detection, distraction and task." In: *International Journal of Human-Computer Studies* 58.5, pp. 515–545.

Bau, Olivier and Wendy E. Mackay (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. Monterey, CA, USA: ACM, pp. 37–46. ISBN: 978-1-59593-975-3. DOI: 10.1145/1449715.1449724.

Beaudouin-Lafon, Michel (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. The Hague, The Netherlands: Association for Computing Machinery, pp. 446–453. ISBN: 1581132166. DOI: 10.1145/332040.332473.

— (2004). "Designing Interaction, Not Interfaces." In: *Proceedings of the Working Conference on Advanced Visual Interfaces*. AVI '04. Gallipoli, Italy: ACM, pp. 15–22. ISBN: 1-58113-867-9. DOI: 10.1145/989863.989865.

Benyon, D. (1994). "A Functional Model of Interacting Systems: A Semiotic Approach." In: *CSCW and Artificial Intelligence*. Ed. by John H. Connolly and Ernest A. Edmonds. London: Springer London, pp. 105–125. ISBN: 978-1-4471-2035-3. DOI: 10.1007/978-1-4471-2035-3_8.

Bérard, François and Amélie Rochet-Capellan (2015). "The Transfer of Learning as HCI Similarity: Towards an Objective Assessment of the Sensory-Motor Basis of Naturalness." In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. Seoul, Republic of Korea: Association for Computing Machinery, pp. 1315–1324. ISBN: 9781450331456. DOI: 10.1145/2702123.2702359.

Bhavnani, Suresh K. and Bonnie E. John (2000). "The Strategic Use of Complex Computer Systems." In: *Human–Computer Interaction* 15.2–3, pp. 107–137. DOI: 10.1207/S15327051HCI1523_3.

Blow, Jonathan (2010). *Video Games and the Human Condition*. blog post: http://the-witness.net/news/2010/10/a-new-lecture-about-game-design/, video of the lecture: https://youtu.be/SqFu5O-oPmU. Retrieved Jully 25, 2020.

Bonderup Dohn, Nina (June 2009). "Affordances revisited: Articulating a Merleau-Pontian view." In: *International Journal of Computer-Supported Collaborative Learning* 4.2, pp. 151–170. DOI: 10.1007/s11412-009-9062-z.

Brühlmann, Florian, Beat Vollenwyder, Klaus Opwis, and Elisa D. Mekler (2018). "Measuring the "Why" of Interaction: Development and Validation of the User Motivation Inventory (UMI)." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, pp. 1–13. ISBN: 9781450356206. DOI: 10.1145/3173574.3173680.

Brunner, Edgar and Madan L. Puri (Jan. 2001). "Nonparametric methods in factorial designs." In: *Statistical Papers* 42.1, pp. 1–52. ISSN: 1613-9798. DOI: 10.1007/s003620000039.

Card, Stuart K., Allen Newell, and Thomas P. Moran (1983). *The Psychology of Human-Computer Interaction*. USA: L. Erlbaum Associates Inc. ISBN: 0898592437.

Carroll, John M and Mary Beth Rosson (1987). *Paradox of the active user.* The MIT Press.

Chang, Bay-Wei and David Ungar (1993). "Animation: From Cartoons to the User Interface." In: *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology*. UIST '93. Atlanta, Georgia, USA: ACM, pp. 45–55. ISBN: 0-89791-628-X. DOI: 10.1145/168642.168647.

Chevalier, Fanny, Pierre Dragicevic, Anastasia Bezerianos, and Jean-Daniel Fekete (2010). "Using Text Animated Transitions to Support Navigation in Document Histories." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '10. Atlanta, Georgia, USA: ACM, pp. 683–692. ISBN: 978-1-60558-929-9. DOI: 10.1145/1753326.1753427.

Cockburn, Andy, Carl Gutwin, Joey Scarr, and Sylvain Malacria (Nov. 2014). "Supporting Novice to Expert Transitions in User Interfaces." In: *ACM Computing Surveys* 47.2, 31:1–31:36. ISSN: 0360-0300. DOI: 10.1145/2659796.

Cooper, Alan, Robert Reimann, David Cronin, and Christopher Noessel (2014). *About Face: The Essentials of Interaction Design*. John Wiley & Sons.

Coppers, Sven, Kris Luyten, Davy Vanacken, David Navarre, Philippe Palanque, and Christine Gris (June 2019). "Fortunettes: Feedforward about the Future State of GUI Widgets." In: *Proc. ACM Hum.-Comput. Interact.* 3.EICS. DOI: 10.1145/3331162.

De Bruijn, Oscar and Robert Spence (May 2008). "A New Framework for Theory-Based Interaction Design Applied to Serendipitous Information Retrieval." In: *ACM Trans. Comput.-Hum. Interact.* 15.1. ISSN: 1073-0516. DOI: 10.1145/1352782.1352787.

deSouza, Clarisse Sieckenius (2005). *The Semiotic Engineering of Human-Computer Interaction (Acting with Technology)*. The MIT Press. ISBN: 0262042207.

Diefenbach, Sarah and Marc Hassenzahl (2008). "Give Me a Reason: Hedonic Product Choice and Justification." In: *CHI '08 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '08. Florence, Italy: Association for Computing Machinery, pp. 3051–3056. ISBN: 9781605580128. DOI: 10.1145/1358628.1358806.

Djajadiningrat, Tom, Kees Overbeeke, and Stephan Wensveen (2002). "But How, Donald, Tell Us How? On the Creation of Meaning in Interaction Design through Feedforward and Inherent Feedback." In: *Proceedings of the 4th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*. DIS '02. London, England: Association for Computing Machinery, pp. 285–291. ISBN: 1581135157. DOI: 10.1145/778712.778752.

Djajadiningrat, Tom, Stephan Wensveen, Joep Frens, and Kees Overbeeke (Sept. 2004). "Tangible Products: Redressing the Balance between Appearance and Action." In: *Personal Ubiquitous Comput.* 8.5, pp. 294–309. ISSN: 1617-4909. DOI: 10.1007/s00779-004-0293-8.

ExtraCredits, ed. (2012). *The Skinner Box – How Games Condition People to Play More*. Youtube. Retrieved Jully 25, 2020. URL: https://youtu.be/tWtvrPTbQ_c.

Fitts, Paul M and Michael I Posner (1967). "Human Performance. Brooks." In: *Cole, Belmont, CA* 5, pp. 7–16.

Gaver, William W. (Mar. 1989). "The SonicFinder: An Interface That Uses Auditory Icons." In: *Hum.-Comput. Interact.* 4.1, pp. 67–94. ISSN: 0737-0024. DOI: 10.1207/s15327051hci0401_3.

— (1991). "Technology Affordances." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '91. New Orleans, Louisiana, USA: Association for Computing Machinery, pp. 79–84. ISBN: 0897913833. DOI: 10.1145/108844.108856.

Gentner, Don and Jakob Nielsen (Aug. 1996). "The Anti-Mac Interface." In: *Commun. ACM* 39.8, pp. 70–82. ISSN: 0001-0782. DOI: 10.1145/232014.232032.

Gibson, James J (1977). "The Theory of Affordances." In: *Perceiving, Acting, and Knowing: Toward an Ecological Psychology. Hillsdale*. Lawrence Erlbaum, pp. 67–82.

— (1979). *The ecological approach to visual perception. Boston, MA, US*. Houghton, Mifflin and Company.

Grossman, Tovi, Pierre Dragicevic, and Ravin Balakrishnan (2007). "Strategies for Accelerating On-Line Learning of Hotkeys." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. San Jose, California, USA: Association for Computing Machinery, pp. 1591–1600. ISBN: 9781595935939. DOI: 10.1145/1240624.1240865.

Grossman, Tovi and George Fitzmaurice (2015). "An Investigation of Metrics for the In Situ Detection of Software Expertise." In: *Human–Computer Interaction* 30.1, pp. 64–102. DOI: 10.1080/07370024.2014.881668.

Grossman, Tovi, George Fitzmaurice, and Ramtin Attar (2009). "A Survey of Software Learnability: Metrics, Methodologies and Guidelines." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA: Association for Computing Machinery, pp. 649–658. ISBN: 9781605582467. DOI: 10.1145/1518701.1518803.

Grudin, Jonathan (Oct. 1989). "The Case against User Interface Consistency." In: *Commun. ACM* 32.10, pp. 1164–1173. ISSN: 0001-0782. DOI: 10.1145/67933.67934.

— (Mar. 2006). "The GUI Shock: Computer Graphics and Human-Computer Interaction." In: *Interactions* 13.2, pp. 45–47, 55. ISSN: 1072-5520. DOI: 10.1145/1116715.1116751.

Guillon, Maxime, François Leitner, and Laurence Nigay (2015). "Investigating Visual Feedforward for Target Expansion Techniques." In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. Seoul, Republic of Korea: Association for Computing Machinery, pp. 2777–2786. ISBN: 9781450331456. DOI: 10.1145/2702123.2702375.

Halasz, Frank and Thomas P. Moran (1982). "Analogy Considered Harmful." In: *Proceedings of the 1982 Conference on Human Factors in Computing Systems*. CHI '82. Gaithersburg, Maryland, USA: Association for Computing Machinery, pp. 383–386. ISBN: 9781450373890. DOI: 10.1145/800049.801816.

Hartson, Rex (2003). "Cognitive, physical, sensory, and functional affordances in interaction design." In: *Behaviour & Information Technology* 22.5, pp. 315–338. DOI: 10.1080/01449290310001592587.

Hassenzahl, Marc (2003). "The Thing and I: Understanding the Relationship Between User and Product." In: *Funology: From Usability to Enjoyment*. Ed. by Mark A. Blythe, Kees Overbeeke, Andrew F. Monk, and Peter C. Wright. Dordrecht: Springer Netherlands, pp. 31–42. ISBN: 978-1-4020-2967-7. DOI: 10.1007/1-4020-2967-5_4.

— (2018). "The Thing and I: (Summer of '17 Remix)." In: *Funology 2: From Usability to Enjoyment*. Ed. by Mark Blythe and Andrew Monk. Cham: Springer International Publishing, pp. 17–31. ISBN: 978-3-319-68213-6. DOI: 10.1007/978-3-319-68213-6_19.

Hassenzahl, Marc, Andreas Beu, and Michael Burmester (Jan. 2001). "Engineering Joy." In: *IEEE Softw.* 18.1, pp. 70–76. ISSN: 0740-7459. DOI: 10.1109/52.903170.

Hassenzahl, Marc, Michael Burmester, and Franz Koller (2003). "AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität." In: *Mensch & Computer 2003: Interaktion in Bewegung*. Ed. by Gerd Szwillus and Jürgen Ziegler. Wiesbaden: Vieweg+Teubner Verlag, pp. 187–196. ISBN: 978-3-322-80058-9. DOI: 10.1007/978-3-322-80058-9_19.

Hofmeester, Kay and Dennis Wixon (2010). "Using Metaphors to Create a Natural User Interface for Microsoft Surface." In: *CHI '10 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '10. Atlanta, Georgia, USA: Association for Computing Machinery, pp. 4629–4644. ISBN: 9781605589305. DOI: 10.1145/1753846.1754204.

Hope, Computer (2019). *Definition of Charms*. Retrieved July 8, 2019. URL: https://www.computerhope.com/jargon/c/charms.htm.

Horn, Michael S. (2013). "The Role of Cultural Forms in Tangible Interaction Design." In: *Proceedings of the 7th International Conference* *on Tangible, Embedded and Embodied Interaction*. TEI '13. Barcelona, Spain: Association for Computing Machinery, pp. 117–124. ISBN: 9781450318983. DOI: 10.1145/2460625.2460643.

Hornbæk, Kasper and Antti Oulasvirta (2017). "What Is Interaction?" In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, pp. 5040–5052. ISBN: 9781450346559. DOI: 10.1145/3025453.3025765.

Hutchins, Edwin L., James D. Hollan, and Donald A. Norman (1985). "Direct Manipulation Interfaces." In: *Human–Computer Interaction* 1.4, pp. 311–338. DOI: 10.1207/s15327051hci0104_2.

Jacob, Robert J.K., Audrey Girouard, Leanne M. Hirshfield, Michael S. Horn, Orit Shaer, Erin Treacy Solovey, and Jamie Zigelbaum (2008). "Reality-based Interaction: A Framework for post-WIMP Interfaces." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy: ACM, pp. 201–210. ISBN: 978-1-60558-011-1. DOI: 10.1145/1357054.1357089.

Jain, Mohit and Ravin Balakrishnan (2012). "User Learning and Performance with Bezel Menus." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: ACM, pp. 2221–2230. ISBN: 978-1-4503-1015-4. DOI: 10.1145/2207676.2208376.

Janlert, Lars-Erik (2014). "The ubiquitous button." In: *Interactions* 21.3, pp. 26–33. URL: https://interactions.acm.org/archive/view/may-june-2014/the-ubiquitous-button.

Johnson, Jeff (2014). *Designing with the Mind in Mind, Second Edition: Simple Guide to Understanding User Interface Design Guidelines*. 2nd. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 0124079148.

Kaptein, Maurits Clemens, Clifford Nass, and Panos Markopoulos (2010). "Powerful and Consistent Analysis of Likert-type Ratingscales." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '10. Atlanta, Georgia, USA: ACM, pp. 2391–2394. ISBN: 978-1-60558-929-9. DOI: 10.1145/1753326.1753686.

Kaptelinin, Victor (2014). *Affordances and Design*. DNK: The Interaction Design Foundation. ISBN: 8792964125.

Kaptelinin, Victor and Bonnie Nardi (2012). "Affordances in HCI: Toward a Mediated Action Perspective." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: Association for Computing Machinery, pp. 967–976. ISBN: 9781450310154. DOI: 10.1145/2207676.2208541.

Karapanos, Evangelos, John Zimmerman, Jodi Forlizzi, and Jean-Bernard Martens (2009). "User Experience over Time: An Initial Framework." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA: Association for Computing Machinery, pp. 729–738. ISBN: 9781605582467. DOI: 10.1145/1518701.1518814.

Kirsh, David (1997). "Interactivity and multimedia interfaces." In: *Instructional Science* 25.2, pp. 79–96. ISSN: 00204277, 15731952. DOI: https://doi.org/10.1023/A:1002915430871.

Kurtenbach, G., T. P. Moran, and W. Buxton (n.d.). "Contextual Animation of Gestural Commands." In: *Computer Graphics Forum* 13.5 (), pp. 305–314. DOI: 10.1111/1467-8659.1350305.

Kurtenbach, Gordon (Oct. 1991). "Making Marks Self-revealing." In: *SIGCHI Bull.* 23.4, pp. 60–61. ISSN: 0736-6906. DOI: 10.1145/126729.1056048.

Kurtenbach, Gordon and William Buxton (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. Hilton Head, South Carolina, USA: ACM, pp. 137–144. ISBN: 0-89791-451-1. DOI: 10.1145/120782.120797.

Kurtenbach, Gordon Paul (1993). "The Design and Evaluation of Marking Menus." UMI Order No. GAXNN-82896. PhD thesis. Toronto, Ont., Canada.

Lallemand, C., V. Koenig, G. Gronier, and R. Martin (2015). "Création et validation d'une version française du questionnaire AttrakDiff pour l'évaluation de l'expérience utilisateur des systèmes interactifs." In: *European Review of Applied Psychology* 65.5, pp. 239–252. ISSN: 1162-9088. DOI: https://doi.org/10.1016/j.erap.2015.08.002.

Lane, David M, H Albert Napier, S Camille Peres, and Aniko Sandor (2005). "Hidden costs of graphical user interfaces: Failure to make the transition from menus and icon toolbars to keyboard shortcuts." In: *International Journal of Human-Computer Interaction* 18.2, pp. 133–144.

Lewis, Chris, Noah Wardrip-Fruin, and Jim Whitehead (2012). "Motivational Game Design Patterns of 'ville Games." In: *Proceedings of the International Conference on the Foundations of Digital Games*. FDG '12. Raleigh, North Carolina: Association for Computing Machinery, pp. 172–179. ISBN: 9781450313339. DOI: 10.1145/2282338.2282373.

Lidwell, William, Kritina Holden, and Jill Butler (2010). *Universal principles of design, revised and updated: 125 ways to enhance usability, influence perception, increase appeal, make better design decisions, and teach through design*. Rockport Pub.

Lime Survey (2019). Retrieved July 8, 2019. URL: https://www.limesurvey.org/.

Malacria, Sylvain, Joey Scarr, Andy Cockburn, Carl Gutwin, and Tovi Grossman (2013). "Skillometers: Reflective Widgets That Motivate and Help Users to Improve Performance." In: *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*.

UIST '13. St. Andrews, Scotland, United Kingdom: ACM, pp. 321–330. ISBN: 978-1-4503-2268-3. DOI: 10.1145/2501988.2501996.

Mayer, Sven, Lars Lischke, Adrian Lanksweirt, Huy Viet Le, and Niels Henze (2018). "How to Communicate New Input Techniques." In: *Proceedings of the 10th Nordic Conference on Human-Computer Interaction*. NordiCHI '18. New York, NY, USA: ACM. DOI: 10.1145/3240167.3240176.

McGrenere, Joanna and Wayne Ho (May 2000). "Affordances: Clarifying and Evolving a Concept." In: *Proceedings of Graphics Interface 2000*. GI '00. Montreal, Quebec, Canada, pp. 179–186.

Nielsen, Jakob (Apr. 1993). "Noncommand User Interfaces." In: *Commun. ACM* 36.4, pp. 83–99. ISSN: 0001-0782. DOI: 10.1145/255950.153582.

Noguchi, Kimihiro, Yulia Gel, Edgar Brunner, and Frank Konietschke (2012). "nparLD: An R Software Package for the Nonparametric Analysis of Longitudinal Data in Factorial Experiments." In: *Journal of Statistical Software, Articles* 50.12, pp. 1–23. ISSN: 1548-7660. DOI: 10.18637/jss.v050.i12.

Norman, Donald A. (1988). *The Psychology of Everyday Things*. Basic Books.

— (May 1999). "Affordance, Conventions, and Design." In: *Interactions* 6.3, pp. 38–43. ISSN: 1072-5520. DOI: 10.1145/301153.301168.

— (2004). *Emotional design : why we love (or hate) everyday things*. New York: Basic Books. ISBN: 0465051367.

— (Nov. 2006). "Logic versus Usage: The Case for Activity-Centered Design." In: *Interactions* 13.6, pp. 45–46. ISSN: 1072-5520. DOI: 10.1145/1167948.1167978.

— (2008). "The way I see IT: signifiers, not affordances." In: *Interactions* 15.6, pp. 18–19. URL: https://interactions.acm.org/archive/view/november-december-2008/the-way-i-see-itsignifiers-not-affordances1.

— (2009). *The design of future things*. Basic books.

— (May 2010). "Natural User Interfaces Are Not Natural." In: *Interactions* 17.3, pp. 6–10. ISSN: 1072-5520. DOI: 10.1145/1744161.1744163.

— (2013). *The Design of Everyday Things*. Basic Books.

O'Neill, Shaleph (2008). *Interactive Media: The Semiotics of Embodied Interaction*. Springer Publishing Company, Incorporated. ISBN: 9781848000353.

O'Neill, Shaleph, David R Benyon, and SR Turner (2002). "Semiotics and interaction analysis." In: *Proceeding of the 11th European Conference on Cognitive Ergonomics*. Citeseer, pp. 44–50.

Orne, Martin T (2009). "Demand characteristics and the concept of quasi-controls." In: *Artifacts in Behavioral Research: Robert Rosenthal and Ralph L. Rosnow's Classic Books* 110, pp. 110–137.

Peres, S. Camille, Franklin P. Tamborello, Michael D. Fleetwood, Phillip Chung, and Danielle L. Paige-Smith (2004). "Keyboard shortcut usage: The roles of social factors and computer experience." In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 48. 5. SAGE Publications Sage CA: Los Angeles, CA, pp. 803–807. DOI: DOI:10.1177/154193120404800513.

Pong, Nicole Ke Chen and Sylvain Malacria (2019). "Awareness, Usage and Discovery of Swipe-Revealed Hidden Widgets in IOS." In: *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*. ISS '19. Daejeon, Republic of Korea: Association for Computing Machinery, pp. 193–204. ISBN: 9781450368919. DOI: 10.1145/3343055.3359713.

Ramakumar, Adhithya (2017). *An analysis of the interactions on iOS 11*. Retrieved September 1, 2019. URL: https://uxdesign.cc/an-analysis-of-the-interactions-on-ios-11-2abc072d42ed.

Raskin, Jef (2000). *The Humane Interface: New Directions for Designing Interactive Systems*. USA: ACM Press/Addison-Wesley Publishing Co. ISBN: 0201379376.

Rasmussen, J. (May 1983). "Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models." In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13.3, pp. 257–266. ISSN: 0018-9472. DOI: 10.1109/TSMC.1983.6313160.

Roth, Volker and Thea Turner (2009). "Bezel Swipe: Conflict-free Scrolling and Multiple Selection on Mobile Touch Screen Devices." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA: ACM, pp. 1523–1526. ISBN: 978-1-60558-246-7. DOI: 10.1145/1518701.1518933.

Ryan, Richard M and Edward L Deci (2000a). "Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being." In: *American psychologist* 55.1, p. 68.

Ryan, Richard M, C Scott Rigby, and Andrew Przybylski (2006). "The motivational pull of video games: A self-determination theory approach." In: *Motivation and emotion* 30.4, pp. 344–360.

Ryan, Richard M. and Edward L. Deci (2000b). "Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions." In: *Contemporary Educational Psychology* 25.1, pp. 54–67. ISSN: 0361-476X. DOI: https://doi.org/10.1006/ceps.1999.1020.

Schramm, Katherine, Carl Gutwin, and Andy Cockburn (2016). "Supporting Transitions to Expertise in Hidden Toolbars." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: ACM, pp. 4687–4698. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858412.

Scimeca, Dennis (2011). *Radoff: Games Are Not Skinner Boxes*. Ed. by Gamasutra. Retrieved Jully 25, 2020. URL: https://www.gamasutra.

com/view/news/126470/Radoff_Games_Are_Not_Skinner_Boxes.php.

Serrano, Marcos, Eric Lecolinet, and Yves Guiard (2013). "Bezel-Tap Gestures: Quick Activation of Commands from Sleep Mode on Tablets." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Paris, France: ACM, pp. 3027–3036. ISBN: 978-1-4503-1899-0. DOI: 10.1145/2470654.2481421.

Shneiderman, Ben (Aug. 1983). "Direct Manipulation: A Step Beyond Programming Languages." In: *Computer* 16.8, pp. 57–69. ISSN: 0018-9162. DOI: 10.1109/MC.1983.1654471.

— (1997). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 3rd. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. ISBN: 0201694972.

Spolsky, Avram Joel (2008). *User Interface Design for Programmers*. Apress.

Svanæs, Dag (Apr. 2013). "Interaction Design for and with the Lived Body: Some Implications of Merleau-Ponty's Phenomenology." In: *ACM Trans. Comput.-Hum. Interact.* 20.1. ISSN: 1073-0516. DOI: 10.1145/2442106.2442114.

Tak, Susanne, Piet Westendorp, and Iris Van Rooij (2013). "Satisficing and the use of keyboard shortcuts: being good enough Is enough?" In: *Interacting with computers* 25.5, pp. 404–416.

Torenvliet, Gerard (July 2003). "We Can't Afford It! The Devaluation of a Usability Term." In: *Interactions* 10.4, pp. 12–17. ISSN: 1072-5520. DOI: 10.1145/838830.838857.

Tyack, April and Elisa D. Mekler (2020). "Self-Determination Theory in HCI Games Research: Current Uses and Open Questions." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, pp. 1–22. ISBN: 9781450367080. DOI: 10.1145/3313831.3376723.

Vermeulen, Jo, Kris Luyten, Elise van den Hoven, and Karin Coninx (2013). "Crossing the Bridge over Norman's Gulf of Execution: Revealing Feedforward's True Identity." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Paris, France: ACM, pp. 1931–1940. ISBN: 978-1-4503-1899-0. DOI: 10.1145/2470654.2466255.

Vicente, Kim J. and Jens Rasmussen (1990). "The Ecology of Human-Machine Systems II: Mediating 'Direct Perception' in Complex Work Domains." In: *Ecological Psychology* 2.3, pp. 207–249. DOI: 10.1207/s15326969eco0203_2.

Ware, Colin (2012). *Information Visualization: Perception for Design*. 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 9780123814647.

Watson, Diane, Mark Hancock, Regan L. Mandryk, and Max Birk (2013). "Deconstructing the Touch Experience." In: *Proceedings of* *the 2013 ACM International Conference on Interactive Tabletops and Surfaces*. ITS '13. St. Andrews, Scotland, United Kingdom: Association for Computing Machinery, pp. 199–208. ISBN: 9781450322713. DOI: 10.1145/2512349.2512819.

Weinschenk, Susan (2011). *100 things every designer needs to know about people*. Pearson Education.

Wensveen, S. A. G., J. P. Djajadiningrat, and C. J. Overbeeke (2004). "Interaction Frogger: A Design Framework to Couple Action and Function through Feedback and Feedforward." In: *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*. DIS '04. Cambridge, MA, USA: Association for Computing Machinery, pp. 177–184. ISBN: 1581137877. DOI: 10.1145/1013115.1013140.

Wigdor, Daniel and Dennis Wixon (2011). *Brave NUI world: designing natural user interfaces for touch and gesture*. Elsevier.

colophon

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/

Final Version as of January 18, 2021 ().