université PARIS-SACLAY

Theoretical Bases of Human Tool Use in Digital Environments

Bases théoriques de l'usage d'outils dans les environnements numériques

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580 Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : Informatique
Graduate School : Informatique et sciences du numérique,
Référent : Faculté de Sciences d'Orsay

Thèse préparée dans l'unité de recherche
Université Paris-Saclay, CNRS, Laboratoire interdisciplinaire des sciences du numérique, 91405, Orsay, France
sous la direction de Michel BEAUDOUIN-LAFON, Professeur, Université Paris-Saclay,
et le co-encadrement de Baptiste CARAMIAUX, Chargé de recherche, CNRS.

Thèse soutenue à Paris-Saclay, le 12 Avril 2022, par

Miguel A. RENOM

Composition du jury

Jean-Daniel FEKETE Directeur de recherche, INRIA	Président
Stéphane CONVERSY Professeur, École Nationale d'Aviation Civile	Rapporteur
Yvonne ROGERS Professor, University College London	Rapportrice
Eric LECOLINET Professeur, Télécom Paris	Examinateur
Michel BEAUDOUIN-LAFON Professeur, Université Paris-Saclay	Directeur de thèse
Baptiste CARAMIAUX Chargé de recherche, CNRS	Co-encadrant de thèse

THÈSE DE DOCTORAT

NNT : 2022UPASG034

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

université PARIS-SACLAY

Titre : Bases théoriques de l'usage d'outils dans les environnements numériques
Mots clés : usage d'outils, raisonnement technique, interaction instrumentale, conception d'interactions

Résumé : L'usage d'outils est un phénomène omniprésent dans nos vies. Nous nous servons d'objets comme outils, parfois au-delà des fonctions qui leur sont assignées, à l'instar d'un couteau qui serait utilisé comme tournevis. Selon l'hypothèse du Raisonnement Technique en neuroscience cognitive, les humains font l'usage d'outils en raisonnant sur les interactions mécaniques entre objets. En considérant ce phénomène comme reposant sur les propriétés physiques des objets, cette théorie explique les transferts de connaissances entre interactions objet-objet similaires, donnant une interprétation à l'usage d'un outil à des fins autres que celles qui ont motivées son invention. Dans les environnements numériques, les interfaces utilisateurs contiennent fréquemment des outils avec des fonctions prédéfinies, par exemple, pour formater un texte, défiler un document ou « zoomer » sur une image. Ces outils sont conçus à des fins spécifiques. Cependant, la littérature en IHM met en avant des exemples avec des utilisateurs qui font usage des outils numériques de manières inattendues. Dans cette thèse, j'étudie l'hypothèse du raisonnement technique comme modèle théorique pour comprendre l'usage d'outils numériques, pour leur fonction propre et inattendue. Ceci sur la base des notions que les utilisateurs ont formé du monde numérique. Je commence par décrire une expérience avec des utilisateurs remplissant une même tâche plusieurs fois avec un éditeur de texte, en réduisant progressivement les commandes auxquelles ils ont accès afin d'effectuer cette tâche. Cette méthodologie force les utilisateurs à utiliser une ou plusieurs commandes pour des fonctions inattendues. Alors que la plupart des participants sont arrivés à réaffecter au moins une commande pour une fonction non prévue, certains ont exprimé des difficultés liées aux biais issus de leur connaissance dans des environnements similaires. Ces observations sont mises en lien avec des phénomènes similaires dans l'usage d'outils physiques, en particulier, la notion de « fixité fonctionelle » et le raisonnement technique. Ensuite, je présente une étude sur les interprétations que les utilisateurs font des possibilités d'action sur les objets dans un environnement numérique à partir de son interface, et en particulier ses barres d'outils. Je m'intéresse aux objets graphiques et textuels. J'ai développé un environnement spécifique dont les objets peuvent être modifiés tant avec les commandes graphiques qu'avec les commandes textuelles. L'environnement possède deux barres d'outils correspondant aux deux types de commandes. Les participants doivent effectuer une série de tâches en utilisant les commandes disponibles dans les barres d'outils. Les participants ont montré une tendance à utiliser les commandes de la barre d'outils présentée pendant l'introduction, suggérant un effet d'amorçage qui pourrait empêcher l'exercice du raisonnement technique pour trouver des stratégies alternatives et éventuellement plus efficaces Finalement, je présente une étude sur des utilisateurs professionnels en édition de texte informant la conception des « Textlets », des objets interactifs qui réifient les sélections de texte comme des outils persistants dans l'édition d'un document. Les textlets représentent un concept génératif qui capitalise sur les principes de l'interaction instrumentale. L'étude a montré l'usage d'un textlet de façon écartée de sa conception originale, ce qui suggère qu'une approche instrumentale peut contribuer à l'utilisation d'outils numériques au-delà des fonctions initialement imaginées. Cette thèse apporte des éléments empiriques montrant la pertinence de l'hypothèse du raisonnement technique comme modèle théorique d'interaction et ouvre la voie à la conception d'interfaces utilisateurs centrées sur des outils et des principes généraux liés aux propriétés « mécaniques » des objets numériques.

Title: Theoretical Bases of Human Tool Use in Digital Environments

Keywords: tool use, technical reasoning, instrumental interaction, interaction design

Abstract: Tool use pervades our everyday life. We spontaneously manipulate objects as tools, sometimes for tasks beyond their assigned function, thereby re-purposing them, such as when a knife is used as a screwdriver. The Technical Reasoning hypothesis in cognitive neuroscience posits that humans engage in tool use by reasoning about mechanical interactions among objects. By modeling tool use based on abstract knowledge about object interactions, this theory explains how tools can be re-purposed for tasks beyond their original design as a product of knowledge transfer. In digital environments, user interfaces often provide tools with pre-defined functions, such as formatting, scrolling or zooming, meant to be used for a specific set of tasks. However, the literature offers examples of users re-purposing digital tools in unexpected ways. This motivated me to investigate the Technical Reasoning hypothesis as a theoretical model for digital tool use, based on the users' acquired knowledge of the digital world. First, I studied computer users performing a task in a digital text editor while being constrained to re-purpose some of its commands. While most participants managed to re-purpose at least one command, some experienced difficulty due to biases stemming from their knowledge of procedures and functions learned from similar environments. I relate these observations to phenomena of physical tool use, particularly, technical reasoning and functional fixedness. Next, I studied how users perceive the possibilities for action on digital objects through toolbars in the interface. Using an experimental environment whose objects support both graphics- and text-oriented commands, I controlled the visibility of corresponding toolbars to introduce the environment to participants before performing tasks with both toolbars available. This resulted in strategies where the preferred command types associated with the toolbar presented in the introduction, suggesting a priming effect, which can hinder the exercise of technical reasoning to use alternative and possibly more efficient strategies. Last, I present a collaboration study about extreme users of text editing tools that led to the design of Textlets: interactive objects that reify text selections into persistent tools for text documents. Textlets constitute a generative concept building on principles of Instrumental Interaction. We observed a user re-purposing a Textlet during an evaluation study, supporting the notion that an instrumental approach may contribute to re-purpose digital tools. This thesis provides evidence of the relevance of the Technical Reasoning hypothesis as a theoretical model for interaction and opens the way to the design of tool-centric interfaces.

ACKNOWLEDGMENTS

I want to express my gratitude to the people and institutions who supported me academically, financially, and emotionally throughout this work. First and foremost, to my thesis supervisor, professor **Michel Beaudouin-Lafon**, who encouraged my interest in the fundamentals of HCI and offered me a comprehensive look at the craft of research. I am forever grateful to him for putting his trust in me to carry this project to its conclusion. Likewise, I am grateful to my co-advisor, **Baptiste Caramiaux**, who guided me through finding the many pieces of the puzzle in every analysis. His support was instrumental in conveying my ideas in writing, leaving no finding behind, no visualization unseen, no effect unreported, and no stone unturned. I am also grateful for the support of the European Research Council (ERC)'s grant N°695464 "ONE: Unified Principles of Interaction." I extend my gratitude to professor **Yvonne Rogers**, professor **Stéphane Conversy**, professor **Eric Lecolinet** and **Jean-Daniel Fekete** for their comments about my work and insightful questions during my defense. And I am grateful to those who volunteered to participate in my user studies for their generous time: I wish I could give you all the candy you deserve.

My days at the *Ex)Situ* team were marked by the incredible people I met there. I am grateful to the head of the team, professor **Wendy E. Mackay** for sharing her invaluable knowledge about experimental design and science etiquette. I had the pleasure of working alongside **Viktor Gustafsson**, a game enthusiast who clarified the MMORPG world for me, one IPA at a time; **Elizabeth Walton**, with whom we discussed our careers, administrative ordeals, cat memes, and cultural subjects about America;[1] **Alexandre Battut** and **Arthur Fages**, who taught me a thing or two about the proper demeanor of the French southerner; **Junhang Yu**, who kindly agreed to help me with the technical aspects of my defense; **Téo Sanchez**, who showed me that you can make pizzas at home with a frying pan; **Camille Gobert**, who shall soon be the ministry of software development; and **Han L. Han** and **Martin Tricaud**, partners in science and office matters. I am so glad I met **Benjamin Bressolette**, an inspiring actor for change; **Janin Koch**, my go-to guide for CHI conferences and fine distilled goods; **Theophanis (Fanis) Tsandilas**, who answered my every inquiry about statistical analyses; **Sarah Fdili Alaoui**, a fierce researcher where technology meets the performing arts; **Philip Tchernavskij**, who first mentored me on reading academic papers, formulating research questions, and making science jokes; **Jessalyn (Sally) Alv-**

1 *"Number One."*

ina, an all-around infinite source of humor; **Stacy Hsueh**, a model of strength and determination for pursuing your research goals; **Alexander Eiselmayer**, a guru on calculating the right number of participants; **Abby (Wanyu) Liu**, who once trusted me with her cat "Shamito" for a few days, turning me into a cat spotting enthusiast; **Jean-Philippe Rivière**, with his mastery of the barbecued Camembert cheese; and **Germán Leiva**, **Carla Griggio** and **Marianela Ciolfi Felice**, who advised me on how to go about this journey.

I want to acknowledge those friends with whom I created so many Parisian and Montevidean memories over these years: **Aylen**, who showed me around Paris, reassured me about my path in academia, and was the best person around to reminisce about the homeland; **Yamil**, who so quickly became a *bro*, holding me for better, for worse, for richer, for poorer, almost as much in sickness as in health, to hang around and to snack on a Tuesday night, till the end of his thesis arrived; **Gastón**, who is to blame for changing my life with his *delirious* suggestions that brought me here, with him always there to cheer me on; **Micaela**, the perfect companion to laugh, to whine, to talk it out, to say it as it is, no matter the distance... I could not ask for a friend with whom to feel more in sync about the ups and downs of human existence; **Camila**, who shared her love for Paris; **Vincent**, the most Uruguayan-like French-born person I met; and lovely people from Université d'Évry, among which **Elisabeth**, **Olivier** and **David**, for whose generosity the English language has not provided me with enough words to thank them; as well as **Claire**, **Fabien**, **Christine**, **Fred** and **Jean-Yves**, all of whom received me with open arms into their group: merci à tous pour votre amitié. And, I want to thank my friends back home,[2] who on every visit made me feel as if I had never left.

Special thanks go to **Gustavo Armagno**, who informally lectured me on Human Factors research before I even thought of pursuing this degree; **Javier Baliosian**, who vouched for me every time I needed a voice to resonate with the giants; **Eduardo Grampín**, who opened the doors of the Computer Science department at my old School of Engineering to experiment with a workshop about HCI; and the Agencia Nacional de Investigación e Innovación (ANII) in Uruguay, who agreed to let me stay in France after my master's degree and gave me additional reasons to return home every year. And finally, I am deeply grateful to my parents, **Miguel** and **Laura**, and sister, **Laura**, who—despite some early fears about my obsession with computers[3]—encouraged and supported me all along the way.

2 Nico, Maxi, Juanma, Maru, Guz, Mele, Bea, Willy, Nora, Vane, Chofi, Sofa, Pasto, Lulo, Dani, Fede, Facu, Santi, Nalu ♡, Omi, Bruno, Vicky, and Gabriela, in no particular order.

3 I was 8 years old and had built a lousy cardboard model of a typical desktop computer from the 90s—with its corresponding tower and boxy monitor—before they bought an actual one for home. I guess they got the message.

SYNTHÈSE EN FRANÇAIS

Les outils sont des médiateurs de nos interactions dans le monde physique. Lorsqu'ils sont manipulés, ils deviennent des extensions de nos mains, augmentant nos capacités à percevoir et à transformer l'environnement qui nous entoure (J. J. Gibson, 1986). En particulier, les humains parviennent à utiliser des objets familiers de manière *inhabituelle*, par exemple, en les utilisant pour des tâches allant au-delà de la fonction qui leur est assignée. Ainsi, on pense spontanément à utiliser un couteau comme tournevis ou une chaise comme butoir de porte. Osiurak et al. (2009) postulent que l'usage d'outils par les humains repose sur la capacité à effectuer du *raisonnement technique* sur les interactions entre les objets. En termes simples, l'hypothèse du raisonnement technique de l'usage d'outils modélise notre compréhension des interactions outil-objet comme un processus dans lequel nous faisons correspondre la connaissance des propriétés physiques de l'objet à des principes mécaniques en fonction de notre objectif. Les principes mécaniques sont incorporés à partir de l'expérience antérieure avec des objets physiques—de la même manière que nous construisons un sens commun des lois physiques du quotidien—, et peuvent être transférés à des interactions entre objets analogues (Osiurak et al., 2010). Cette théorie fournit un modèle élégant qui décrit la manière dont les humains font usage d'outils au-delà des fonctions identifiées pour eux, grâce à des abstractions des objets et des interactions impliqués.

Les ordinateurs englobent la notion d'« outil cognitif », i.e., les *outils numériques* par lesquels les utilisateurs opèrent sur des objets (Bødker, 1987). Cependant, les outils numériques sont conçus pour des tâches spécifiques et les utilisateurs s'écartent rarement des usages qui leur sont assignés. De plus, les *design patterns* contemporains « enferment » les outils numériques à l'intérieur des logiciels (Beaudouin-Lafon, 2017), ce qui constitue une différence fondamentale par rapport à l'usage d'un outil physique, que les utilisateurs peuvent utiliser dans toute tâche où ils le jugent approprié. Contrairement au passage d'une tâche à l'autre avec les mêmes outils physiques en main, le fait de passer à une autre application implique un changement de jeu d'outils, ce qui oblige l'utilisateur à appliquer des techniques potentiellement différentes pour accomplir des objectifs similaires. Par exemple, choisir la couleur du texte dans un éditeur se limite généralement à un bouton dans la barre de mise en forme avec un menu déroulant contenant une palette de couleurs, alors que dans les environnements d'édition graphique, il est possible d'utiliser des nuanciers de couleurs facilement disponibles ainsi que l'outil *Pipette*. À l'inverse, s'il est fréquent de pouvoir réutiliser facilement les propriétés d'un texte grâce aux styles de mise en forme, il n'existe pas de concept popularisé de style réutilisable pour les formes graphiques. D'ailleurs, même dans un même environnement numérique, les outils ont tendance à remplir des fonctions limitées à quelques objets. Par exemple, les applications de conception graphique populaires divisent généralement les fonctions entre les actions orientées vers les « pixels » et celles orientées vers les vecteurs. Mais alors que l'on peut utiliser l'outil *Gomme* pour *supprimer* les pixels, il n'a aucun effet sur les vecteurs, bien qu'ils aient des nœuds et des courbes qui supportent eux-mêmes une commande *supprimer*. En revanche, dans le monde physique, une gomme effacera tout ce qui est « effaçable », i.e., qui supporte la commande. On peut penser que cette limitation contraint les utilisateurs à suivre des procédures spécifiques pour accomplir des tâches avec des outils spécifiques relatifs à l'application, ce qui entrave les usages créatifs telles que ceux qui conduisent à l'usage inhabituel d'outils physiques.

La littérature en IHM contient des exemples d'utilisateurs qui parviennent à s'écarter des fonctions assignées aux outils numériques (Ciolfi Felice et al., 2016; Dix, 2007; Dourish, 2003). Par exemple, lorsqu'ils s'envoient des messages ou des courriels à eux-mêmes pour prendre des notes, ou qu'ils téléchargent des fichiers en pièces jointes pour les partager avec des collègues, les utilisateurs se sont appropriés la technologie d'une manière similaire à l'usage d'un couteau comme tournevis (Dix, 2007). Cette correspondance du monde physique avec les interactions numériques est également reprise par certains modèles d'interaction existants visant à tirer parti des capacités cognitives humaines dans les environnements numériques (Jacob et al., 2008; Jetter et al., 2014). En particulier, l'interaction instrumentale (Beaudouin-Lafon, 2000) est basée sur des *instruments numériques* qui font office de médiateurs des interactions entre l'utilisateur et des *objets du domaine*. Cependant, à ma connaissance, les modèles basés sur la médiation par les instruments ne sont pas soutenus par une théorie cognitive de l'interaction. En particulier, nous manquons des connaissances sur la façon dont les utilisateurs peuvent donner un sens aux possibilités d'action entre les outils et les objets, ainsi que sur la façon dont les utilisateurs font des usages inhabituels d'outils numériques.

Dans cette thèse, je m'appuie sur la notion d'usage inhabituel d'outils physiques pour étudier l'hypothèse du raisonnement technique comme théorie explicative de l'usage inhabituel d'outils numériques, généralisable aux interactions médiées par des instruments. J'étudie ces usages inhabituels en concevant des environnements et des protocoles expérimentaux pour observer des participants qui adaptent les commandes pour accomplir des tâches. Cette thèse vise à permettre le développement d'approches pour la conception d'interactions basées sur l'*activité médiée par ordinateur* (Kaptelinin, 1996), et a identifier des principes unifiés d'interaction humain-machine (Beaudouin-Lafon, 2017).

Les projets du présent manuscrit sont inclus dans des chapitres organisés de manière à présenter une histoire cohérente. Les deux premiers chapitres sont consacrés à l'introduction des fondements théoriques de la thèse, en commençant par l'hypothèse du raisonnement technique et en passant à d'autres théories de l'IHM liant l'usage de l'ordinateur à l'usage d'outils dans des activités de travail humaines. Les autres chapitres contiennent des études avec des participants utilisant des prototypes expérimentaux, ainsi qu'une étude observationnelle des pratiques de travail avec l'édition de texte menant au développement d'un nouvel outil numérique suivant les principes de l'interaction instrumentale.

LE CHAPITRE 1 introduit l'hypothèse du raisonnement technique comme modèle de l'usage d'outils par l'humain, en décrivant ses composantes de base auxquelles je fais référence tout au long du manuscrit.

LE CHAPITRE 2 présente les travaux antérieurs sur lesquels je m'appuie pour mettre en relation l'usage d'outils physiques et d'outils numériques, et se termine par la similarité cognitive reflétée dans l'usage inhabituel d'outils numériques.

LE CHAPITRE 3 décrit une expérience visant à explorer l'hypothèse du raisonnement technique dans l'usage d'outils numériques, y compris la conception d'un environnement d'édition de texte et d'un protocole pour forcer les participants à effectuer des usages inhabituels de commandes d'édition de texte.

LE CHAPITRE 4 décrit une expérience visant à explorer les effets d'une interface qui suscite des connaissances spécifiques des utilisateurs sur l'usage de commandes sur des objets numériques.

LE CHAPITRE 5 commence par une étude observationnelle concernant la pratique quotidienne d'utilisateurs d'éditeurs de texte qui doivent respecter des contraintes et des exigences de cohérence dans leurs documents, pour aboutir à la conception d'un nouvel outil numérique qui réifie les sélections de texte en objets réutilisables, et à une courte étude finale d'utilisateurs testant le concept.

LE CHAPITRE 6 conclut par un aperçu des contributions de la thèse et des pistes de travail pour le futur.

CONTENTS

Appendices

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

"...[M]ore often than not I've observed
that convenient approximations
bring you closest to comprehending the true nature of things."

—Haruki Murakami, *Hard-Boiled Wonderland and the End of the World*

Tools mediate our interactions in the physical world. We spontaneously engage with objects to use them as tools (Osiurak, 2020). When manipulated, they become extensions of our hands that augment our capabilities to perceive and transform the environment (J. J. Gibson, 1986). In particular, humans manage to use familiar tools in *unusual* ways, for example, *re-purposing* them for tasks beyond their assigned function. After all, it is not hard to think of using a knife as a screwdriver or a chair as a door stop. Osiurak et al. (2009) posit that human tool use is supported by the ability to perform *technical reasoning* about interactions among objects. Simply put, the Technical Reasoning hypothesis of human tool use models our understanding of tool-object interactions as a process where we match knowledge of physical object properties to mechanical principles according to our goal. Mechanical principles are incorporated from experience with physical objects in nature—similar to how we learn about naïve physics—, managing to transfer it to analogous object interactions (Osiurak et al., 2010). This provides us with an elegant model for how humans re-purpose physical tools, based on identifying analogous abstractions of the objects and interactions involved. Computers embrace the notion of "cognitive tools," namely, the *digital tools* through which users operate on objects (Bødker, 1987). However, digital tools are designed for specific tasks, and users rarely deviate from uses according to the tool's assigned functions. Additionally, contemporary design patterns "trap" digital tools inside applications (Beaudouin-Lafon, 2017), constituting a fundamental break from physical tool use whereby users carry their tools to any task that they see fit.

Unlike switching among physical tool-based tasks, changing the focus to another application implies a change of tool set, which potentially forces the user to apply different interaction techniques to use the new tools. For example, changing the text color in a word processor is usually limited to a toolbar button and a drop-down menu with color swatches, while in graphic editing environments it is possible to use readily available color swatches as well as color pickers. Conversely, while it is frequent to easily reuse text properties through formatting styles, there is no popularized concept of reusable style for graphical shapes. Moreover, despite the malleability offered by computer systems (Coughlan and P. Johnson, 2009), even within the same environment, digital tools tend to fulfill limited purposes associated with few objects. For example, popular graphic design applications usually split functions between pixel-based and vector-based actions. While one may find an *Eraser* tool to *delete* pixels, it has no effect on vectors, despite having nodes and curves that support a *delete* command themselves. By contrast, in the physical world, an eraser will erase anything as long as it is "erasable," i.e., it supports the command. Arguably, this limitation constrains users to follow specific procedures to complete tasks with specific tools pertaining to the application, hindering creative uses such as those leading to re-purposing physical tools.

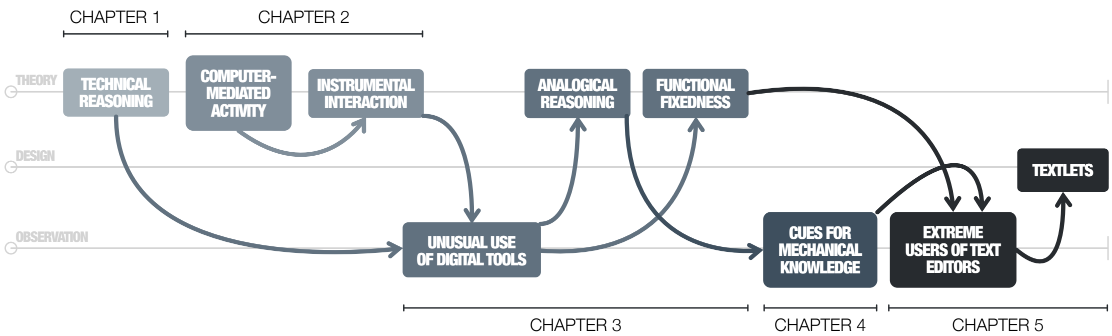
The literature in HCI does contain examples of users who manage to re-purpose digital tools (Ciolfi Felice et al., 2016; Dix, 2007; Dourish, 2003), by using them for tasks that exceed their assigned function, thereby deviating from the designers' expectations. For example, when sending chat messages or email to themselves for note keeping or uploading files as attachments for file-sharing with colleagues, users have *appropriated* technology in ways akin to using knives as screwdrivers (Dix, 2007). This mapping of the physical world onto digital interactions is also picked up by some interaction models aiming at leveraging human cognitive capabilities in digital environments (Jacob et al., 2008; Jetter et al., 2014). In particular, Instrumental Interaction (Beaudouin-Lafon, 2000) is based on *digital instruments* that mediate interactions between the user and the *domain objects* of the system, such as how physical tools mediate interactions with other physical objects. However, to the extent of my knowledge, instrument-mediated models lack support from a cognitive theory of interaction about how users can make sense of the possibilities for action between tools and objects, and in particular, how users re-purpose digital tools.

In this thesis, I draw on the notion of physical tool re-purposing to investigate the technical reasoning hypothesis as explanatory theory for *digital* tool re-purposing, generalizable to instrument-mediated interactions. I study re-purposing as the unusual use of tools by designing experimental environments and protocols to observe users choosing and adapting commands to complete tasks. This thesis aims at supporting a computer-mediated activity approach to interaction design (Kaptelinin, 1996), ultimately contributing to developing unified principles of human-computer interaction (Beaudouin-Lafon, 2017).

STATEMENT

Digital tools proposed by contemporary software serve limited purposes and rarely work beyond the scope of a few system objects. This constitutes a fundamental break from how humans interact with physical tools, which they can re-purpose beyond their technologically and culturally assigned functions. However, there are examples of users who manage to re-purpose digital tools in ways never envisaged by their designers, thus appropriating them for new tasks. The Technical Reasoning hypothesis of human tool use posits the use of physical tools as a product of reasoning about the mechanical principles at play in interactions among objects. In particular, technical reasoning models the re-purposing of physical tools as based on analogies of known object interactions. I am interested in the relevance of technical reasoning as a model for the re-purposing of digital tools. I investigate this question by conducting studies based on experimental digital environments where I observe participants use and re-purpose digital tools to analyze their thought process. My goal is to find evidence of a close cognitive relationship between how users make sense of physical and digital object interactions, so as to support the development of interfaces that are centered on the notion of instrument-mediated activity, and ultimately contribute to the development of unified principles of human-computer interaction.

APPROACH



Triangulation of theoretical, design and observational methodologies in this thesis following the approach to HCI research proposed by Mackay and Fayard (1997).

This thesis takes place in the context of the ERC Advanced Grant "ONE – Unified Principles of Interaction." The project aims at consolidating the sparse universe of conceptual models, interaction styles and devices through a unified theory of interaction that intersects present and future digital environments (Beaudouin-Lafon, 2017). ONE organizes interactive systems in *substrates* to contain information. Additionally, it defines *instruments* as particular substrates that mediate the interaction between users and other substrates in the systems. The goal of ONE is to develop a conceptual model of interaction that empowers users to appropriate digital environments.

Computer-based artifacts mediate human activity with physical and digital objects "*through the interface*" (Bødker, 1997), as psychological tools represented in the digital world. Artifacts are re-purposed and incorporated into new activities, acquiring meaning as instruments (Béguin and Rabardel, 2000). I then capitalize on Instrumental Interaction to offer a computer-mediated perspective of user interfaces, observing that users manage to re-purpose computer-based artifacts in ways unexpected by their designers. I include user studies to investigate how users understand and re-purpose computer-based artifacts to argue about the compatibility of the results with elements of the Technical Reasoning hypothesis. Finally, I include an observational study of extreme users and their practices with commercial software, seeding the development of a concept of digital instrument and the implementation of prototypes around it.

I borrow methods from cognitive psychology, social sciences and engineering. I triangulate between theoretical, observational and design perspectives of these (Mackay and Fayard, 1997) to balance my multidisciplinary approach, starting from the exploration of a cognitive science theory towards understanding its mapping onto digital environments.

From a theoretical angle, I build on theories and methods from cognitive psychology, social sciences and HCI models:

- the technical reasoning hypothesis of human tool use grounds my approach to study digital tool-based interactions (Osiurak, 2020);

- computer-mediated activity (Kaptelinin, 1996) weaves physical tool use with interaction design grounded on activity theory (Bødker, 1997);

- instrumental interaction (Beaudouin-Lafon, 2000) offers a generative theory for the design of interfaces where interaction is mediated by *instruments*; and

- thematic analysis (Braun and Clarke, 2006) is used as a tool for the qualitative analysis of interview notes and verbal protocols.

From an observational angle, I apply quantitative and qualitative methods of research such as observational studies and interviews to understand users' perspectives on digital tools:

- I conduct observational studies of users using experimental digital environments that include observing the participants' approach to re-purpose digital tools and understand their transfer of knowledge from past experience;

- I employ structural and critical-incident interviews to gather information about extreme users of text editors with the purpose of designing digital artifacts; and
- I employ questionnaires to complement the data gathered from observations and interviews, e.g., to measure experience with text editing or creativity traits.

From a design angle, I create environments and digital tools to elicit user behavior and evaluate it:

- I design a prototype text editor that allows an instructor to control the availability of commands to observe users re-purposing the remaining ones;
- I design a prototype editor that allows working with its digital objects as both text and graphics; and
- I collaborate in the conceptual model and design of a prototype that allows for searching and replacing text selections in a document based on a novel digital tool.

OVERVIEW

The projects in the current manuscript are included in chapters ordered so as to present a coherent story. The first two chapters are concerned with introducing the theoretical underpinnings of the thesis, starting with the Technical Reasoning hypothesis and transitioning to other theories of HCI relating computer use with tools and human work activity. The remaining chapters contain studies with participants using experimental prototypes, as well as an observational study of work practices with text editing leading to the development of a novel digital tool following the principles of Instrumental Interaction.

Chapter 1 introduces the technical reasoning hypothesis of human tool use, describing its basic components to which I will make reference throughout the thesis.

Chapter 2 presents the background work on which I draw to weave physical and digital tool use together, closing with the cognitive similarity reflected in the re-purposing of digital tools.

Chapter 3 describes an experiment to explore the technical reasoning hypothesis in digital tool use, including the design of a text editing environment and a protocol to force participants to perform unusual uses of text editing commands.

Chapter 4 describes an experiment to explore interface cues that prime the users' knowledge of digital objects and commands from past experience.

Chapter 5 begins with an observational study concerning the daily practice of extreme users of word processors following constraints and consistency requirements, leading to the design of a novel digital tool that reifies text selections into reusable objects, and a closing short user study testing the concept.

Chapter 6 concludes with an overview of the thesis' contributions and avenues for future work.

PUBLICATIONS

Chapter 3 reproduces the following publication in the ACM 2022 CHI Conference on Human Factors in Computing Systems:

> **Miguel A. Renom**, Baptiste Caramiaux, and Michel Beaudouin-Lafon. 2022. Exploring Technical Reasoning in Digital Tool Use. In *CHI Conference on Human Factors in Computing Systems* (*CHI '22*), (*Honorable Mention*). Association for Computing Machinery, New York, NY, USA, Article 579, 1–17. `https://doi.org/10.1145/3491102.3501877`

Chapter 4 is the basis for an article to be submitted to the 2023 CHI Conference on Human Factors in Computing Systems.

Chapter 5 is based on a publication I wrote in collaboration with Han L. Han and Wendy E. Mackay in Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, and includes modifications of my authorship for the purpose of this manuscript:

> Han L. Han, **Miguel A. Renom**, Wendy E. Mackay, and Michel Beaudouin-Lafon. 2020. Textlets: Supporting Constraints and Consistency in Text Documents. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (*CHI '20*), (*Honorable Mention*). Association for Computing Machinery, New York, NY, USA, 1–13. `https://doi.org/10.1145/3313831.3376804`

1 THE TECHNICAL REASONING HYPOTHESIS

Why are we such good tool users? The Theory of Affordances (J. J. Gibson, 1986) posits that humans perceive objects by what they *afford* to them. For example, an individual may perceive that a step affords climbing to her, whereas a mouse may perceive that the same step affords shelter but not climbing. J. J. Gibson (1986) argues that affordance perception is not based on stored knowledge, but that it is *direct*, based on the animal's needs and body capabilities, i.e., perception is for action (Osiurak and Badets, 2016). In consequence, J. J. Gibson (1986) describes tool use as the product of the direct perception of affordances of detached objects, at a minimum, *carrying* and *grasping*.

Later works on cognitive models of tool use have found that humans elicit stored sensorimotor knowledge about tools, evidenced by the activation of motor stimuli just from observing them (Osiurak and Badets, 2016). In other words, subjects were primed by observation without intention, in contrast with the premise of a direct route between perception and action. Additionally, Osiurak et al. (2010) discuss the body- and hand-centered nature of the Theory of Affordances, arguing that humans do not need to manipulate objects in order to understand how they interact with each other. The authors illustrate this with the fact that "*humans are able to determine that an airplane is heavy enough to crush a car (...) while they can manipulate neither the airplane nor the car*" (Osiurak et al., 2010). Therefore, they argue that humans possess knowledge of technical means, i.e., what the objects "afford" to other objects, such as resistance, sharpness, etc., which they use to reason about the interactions among them. This is reflected in the Technical Reasoning hypothesis (Osiurak et al., 2009), an alternative cognitive model of human tool use that is based on reasoning rather than manipulation.

This chapter is concerned with introducing the reader to the Technical Reasoning hypothesis. In order to do so, I briefly discuss the notion of tool use and some prominent contemporary perspectives of human tool use in cognitive science. I then discuss the Technical Reasoning model in more detail, using examples to introduce its components. Last, I discuss how technical reasoning represents a compelling tool to analyze the re-purposing of physical tools compared to other cognitive models, as this will be an important aspect to analyze in digital tool use.

1.1 HUMAN TOOL USE

Humans spontaneously engage in physical tool use (Osiurak et al., 2010). The traditional view of tool use is that it encompasses the manipulation of *detached objects* to exert actions to transform other objects and organisms in the environment (Osiurak et al., 2010). For example, J. J. Gibson (1986) states that "*when in use, a tool is a sort of extension of the hand, almost an attachment to it or part of the user's own body, and thus is no longer part of the environment of the user.*" However, "*when not in use, the tool is simply a detached object of the environment, graspable and portable, to be sure, but nevertheless external to the observer.*" But, how do tools that are not hand-held fit this concept? What about a tool that requires the user to manipulate the object instead, such as a table saw?

Recently, St Amant and Horton (2008) offer a broader definition of tool use:

> "*Tool use is the exertion of control over a freely manipulable external object (the tool) with the goal of (1) altering the physical properties of another object, substance, surface or medium (the target, which may be the tool user or another organism) via a dynamic mechanical interaction, or (2) mediating the flow of information between the tool user and the environment or other organisms in the environment.*"

In their conceptualization, the authors revisit the notion of "detached" as a requirement, defining the tool as what is "*freely manipulable,*" thus opening the way to consider attached things as tools on the basis of their degrees of freedom. For example, under this notion, one could consider a light switch as a tool to alter the visibility of a room. Additionally, the goals of tool use are expanded to consider capturing information, such as when using a measuring tool or a magnifying glass.

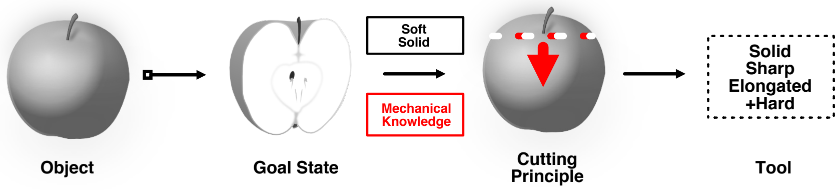
However, St Amant and Horton (2008) echo the notion that a tool is "*what is manipulated*" (Osiurak et al., 2010), as opposed to what is not in contact with the user. In this sense, Osiurak et al. (2010) argue that it is not possible to generalize an objective distinction of tool from object. The authors illustrate this with sanding a piece of wood while keeping the sandpaper attached to the floor (Osiurak et al., 2010). In this example, it would seem obvious for an observer that the sandpaper is the tool, yet, it is not what is being manipulated. The authors argue that while distinguishing tool use from other behaviors may be trivial, we do not have an appropriate psychological distinction of tool and object to employ. For this reason, they advocate for a distinction made on the basis of convenience for the analysis (Osiurak et al., 2010), i.e., letting it be a choice of the observer. In this thesis, I differentiate tools from objects based on socio-cultural and technological conventions, i.e., what we recognize as the tool from experience, e.g., the sandpaper is a tool because it is technologically designed and socially adopted as such.

1.2 COGNITIVE MECHANISMS OF HUMAN TOOL USE

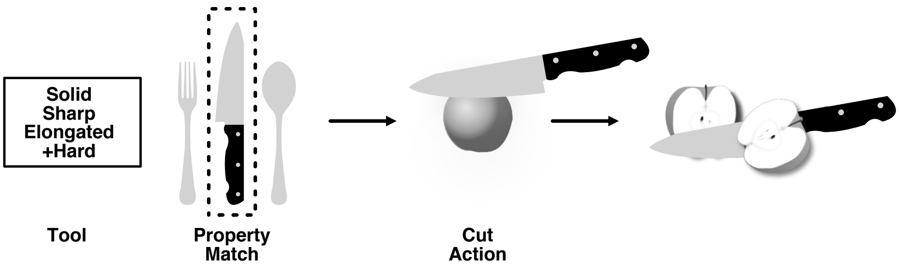
A prominent perspective of human tool use centers around the manipulation of objects and the storage of semantic information for that purpose, namely, the *manipulation-based* approach (Osiurak and Badets, 2016). This is based on studies showing that users' motor knowledge of usual (i.e., conventional) uses of familiar tools is primed just by observing them, and that these individuals are able to perform correct uses of novel (i.e., unfamiliar) tools (Osiurak and Badets, 2016). Proponents of this approach generally agree that two mechanisms compete and complement each other in using tools, namely, *function knowledge* and *manipulation knowledge* (Osiurak and Badets, 2016). Function knowledge is concerned with making semantic associations of tools and objects as to what are their functions and appropriate interactions (Osiurak and Badets, 2016). For example, making the association between hammers and nails to define a pounding action as appropriate. Manipulation knowledge is concerned with identifying the salient structural features of tools so as to infer the way in which they are used from their physical properties (Osiurak and Badets, 2016). For example, realizing how to swing a hammer because of its graspable handle, or figuring out how to grip pliers because of the arms. In this sense, the direct perception of affordances lies behind the manipulation knowledge (Osiurak and Badets, 2016).

Humans also use tools in unusual (i.e., unconventional) ways, such as when a knife is used for turning a screw. This type of tool use is usually regarded as another product of the direct perception of affordances of tools, i.e., manipulation knowledge (Osiurak et al., 2009). Given that these are theoretically two separate systems, i.e., function knowledge for usual use and manipulation knowledge for unusual use, Osiurak et al. (2009) argue that there should not be a significant relationship between the ability to use familiar tools and that to perform unusual uses of tools. In a study involving patients with different brain conditions leading to apraxia, Osiurak et al. (2009) found that their performance of usual uses of familiar tools correlated *only* with their performance of unusual uses of tools, therefore suggesting that these are supported by the same underlying cognitive mechanism. Grounded on these results, the authors posit that humans possess an ability to perform technical reasoning about interactions among objects, based on knowledge of *technical means*, powering the use of tools. This is presented as an alternative to the manipulation-based approach,

Apraxia is a condition identified by severe impairments to use tools. Previous work has found an association between left brain-damaged patients and the inability to demonstrate the usual use of familiar objects (see Osiurak et al. (2009))



(a) Determine the appropriate mechanical action to perform.



(b) Choose the object with the abstract properties required by the mechanical principle.

Figure 1.1: Using mechanical knowledge to slice an apple.

named the *reasoning-based* approach to human tool use (Osiurak and Badets, 2016).

1.3 TECHNICAL REASONING

The Technical Reasoning hypothesis posits that in order to use tools, humans perform technical reasoning, involving matching object properties with technical principles that let them produce simulations of the result of mechanical interactions among objects. Let us illustrate this process with an example of a tool user that desires to split an apple in two halves (Figure 1.1). The user first pictures the *goal state* of the object, i.e., the two halves as separate parts. The next step is to find *how* to split this particular object. Let us assume that she has split other objects before, therefore arriving at the conclusion that a *cut* action could be appropriate for the task. Cutting requires a blade-like object with a sharp edge to be pressed against the target object as it cuts through it transversely. We call this a *technical principle* at play, in this case, *cutting*, resolving the technique or movement of the tool against the object whose outcome would be the goal state (Figure 1.1a).

The user must know that the cutting principle demands that the tool have certain characteristics. To be more specific than "blade-like object," we are roughly talking about something solid (so that it does not bend when pressed), sharp (so that it can cut through the object), elongated (so that it cuts a line as opposed to a hole) and harder than the target object (so that it does not break). This matching of the object properties and the principles is modeled as *mechanical knowledge*, deemed acquired by experience. Let us assume now that a steel fork, a soft plastic spoon and a kitchen knife with a steel blade are the only other objects present in the context of this activity, neither of which have ever been used before by the user.

Having activated the appropriate mechanical knowledge for the cutting principle, the user must look around the environment to find the matching tool for the apple. While the fork may be solid, elongated and harder than the apple, it does not have a sharp edge to cut through it. The spoon, on the contrary may have a sharp-enough edge but because it is made of soft plastic, it is not hard enough to resist the pressure cutting through the apple. The knife, however, matches all the properties necessary to interact with the apple in the way specified by the mechanical knowledge of the cut action. Realizing this, the user proceeds to manipulating the kitchen knife and the apple according to the learned cutting principle until reaching the goal state (Figure 1.1b).

1.3.1 Technical Principles

For the Technical Reasoning hypothesis, tool use involves knowledge of techniques to use the appropriate interaction between the tool and a target object, namely, knowledge of technical principles (Osiurak et al., 2009). For example, the *cutting* principle entails pressing a sharp edge of an object against the target object to produce a gap along the direction of the tool movement; the *lever* principle implies pushing or pulling a side of a long object to move the target object on the other side in the opposite direction. the *smashing* principle involves pounding the target object with an object having a larger surface to break it down into smaller pieces; etc. Technical reasoning would model a simulation of the *mechanical* interaction resulting from the movement of the tool against or upon the target object (Osiurak and Badets, 2016), e.g., how the knife breaks apart the apple as the blade cuts through it. This model, however, is not concerned with how the movement of the tool is carried out because that pertains to motor functions (Osiurak and Badets, 2016). Technical principles are only concerned with object-object interactions.

A fundamental aspect of tool use lies in the ability to transfer technical principles to new tool-based interactions (Osiurak et al., 2010). In the example of splitting an apple, I illustrate this as the user recalling situations where she has split objects, arriving to the conclusion that a cut action will suffice. After all, we do not need to learn how to cut from scratch when facing a new fruit, as if it were any different from other fruits, food or materials that can be cut with a kitchen knife (Osiurak and Badets, 2016). This applies not only to the target object but to the interaction between the tool and the object as a whole. Let us consider the example of using a serrated knife instead

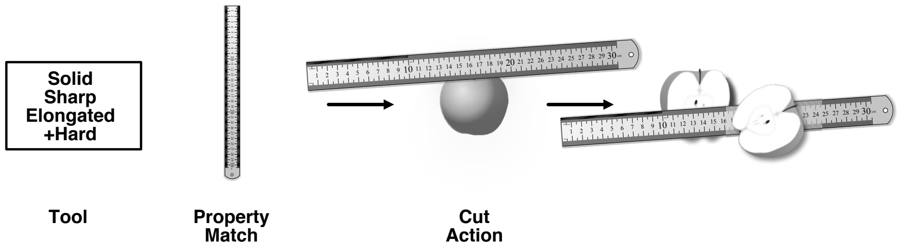


Figure 1.2: Cutting an apple using the ruler instead of a knife.

of a kitchen knife to cut the apple. With a serrated knife, instead of pressing the blade downwards against the apple, the user presses it slightly while performing a back and forth movement along the length of the blade. However, this is in essence identical to the *sawing* principle used to cut wood, where neither saw nor wood resemble the knife or the apple. Therefore, an abstraction of the interaction must be in place in order to reify the mechanical action, i.e., give it a new use, so that it fits the two situations, rather than just understanding each in terms of the specific objects involved.

1.3.2 Mechanical Knowledge

Let us take the example of cutting an apple once again. If we were asked to use a metallic ruler to cut the apple—for lack of a kitchen knife—we would realize that we can do it using the same mechanical action as with the knife (Figure 1.2). Essentially, the cutting principle is transferred between both scenarios. However, how can one tell that both situations are analogous within the cutting principle? After all, the ruler is not designed with a sharp edge for cutting nor is it a popular convention to use it for such tasks. Necessarily, there should be a mechanism in place to recognize what is "similar enough" between a knife and the ruler to realize this.

Mechanical knowledge models knowledge of technical principles and properties of objects (Osiurak et al., 2010). Humans then use mechanical knowledge to match technical principles with objects by their abstract properties, so as to, for example, determine what are the necessary conditions for an object to be able to cut another. By *abstract*, the authors mean that it is not objective of each and every object, but rather a subjective description of the technical means that they offer, e.g., resistance, softness, etc. (Osiurak et al., 2010). After all, one does not need to differentiate between steel or iron in order to tell that either can cut through an apple.

The process to determine which mechanical principle applies is modeled as establishing the properties of the tool relative to the target object— such as being harder, heavier, longer, etc.—so as to produce the desired effect after performing the mechanical action directed by the principle (Osiurak et al., 2010). For example, this is illustrated in the *characteristics* of the knife and the apple, using words such as *sharp* and *solid*. For the mechanical action associated to the cutting principle it is necessary to have the tool be solid, sharp, elongated and harder in comparison to the target object. Back to the example, the knife fulfills these requirements to cut an apple, and so could a metallic ruler being sharp enough to cut it. A skeptic reader could raise her eyebrows upon my lack of precision regarding the sharpness factor. However, I am not referring to an objective measure of sharpness for a cutting action—even if it existed—but rather its characterization relative to the target object, e.g., a knife can be sharp enough to cut an apple, but may not be sharp enough to cut through glass.

The essential concept underlying the Technical Reasoning hypothesis is that no object serves a unique technical mean, e.g., knives can cut, pierce, press, etc., and no technical mean is unique to a particular object, e.g., cutting can be achieved with knives, metallic rulers, plastic cards, etc. (Osiurak et al., 2009). Mechanical knowledge explains the transfer of technical principles through the identification of the similar technical means between different object interactions. As such, a steel knife is harder than an apple, as could be the metallic ruler; it is long enough to split an apple, as could be the metallic ruler; and is sharp enough to cut an apple, as could be the metallic ruler. The abstract and relative nature of mechanical knowledge is better explained as how we understand naïve physics. Indeed, humans do not need to understand or be able to explain the laws of gravity to know that a table can support a glass (Osiurak et al., 2010). Through mechanical knowledge, the reasoning-based approach to human tool use models how humans make sense of interactions between objects, drawing on experience acquired in the physical world.

1.4 FUNCTION KNOWLEDGE & TOOL RE-PURPOSING

Both the manipulation-based and the reasoning-based approaches to human tool use agree that humans accumulate *function knowledge* associated to objects (Osiurak and Badets, 2016), such as that knives *cut* apples and screwdrivers *turn* screws. We call the latter examples of *usual* use (Osiurak et al., 2009) because they fulfill technological purposes or follow social conventions, i.e., what they were made for or how we learned that they are used. Nevertheless, besides usual uses, humans are able to use objects in *unusual* ways (Osiurak et al., 2009), re-purposing them for tasks beyond their learned function. Such is the case when managing to use a ruler to cut an apple instead of using a knife.

The manipulation-based approach suggests that both the use of unfamiliar tools and the unusual use of familiar tools are based on the perception of affordances (Osiurak and Badets, 2016). However, as

Osiurak et al. (2010) argue, the perception of affordances is centered on the user and not on the target object. For example, one can perceive that a wrench affords pounding by gripping its handle as far from its head as possible, but how does one arrive at the conclusion that pounding is the right action for the target object? What if the target object is one that we cannot manipulate? On the other hand, the reasoning-based approach posits that understanding tool-object interactions is based on technical reasoning, stating that the perception of affordances happens after mechanical knowledge, in order to manipulate the objects to fulfill the interaction. In this case, function knowledge is acknowledged as semantic information, useful to identify where to find tools appropriate for familiar tasks, e.g., knowing that knives are for cutting food and usually found in the kitchen (Osiurak and Badets, 2016). Deciding on using them is considered always a product of mechanical knowledge.

Re-purposing describes a unusual use of an object where the user momentarily adopts a meaning deviating from their technological purposes and social conventions. A ruler used as a knife is just a matter of the context, product of a competition between semantic knowledge of proper uses of tools and the circumstances that push the user to resolve the situation with what is available at the moment. This does not mean that re-purposing permanently changes the use of a tool. Arguably, people do not start reaching for knives for turning screws once they have done it for the first time. Rather, technical reasoning appears as a mechanism to solve a situation that requires reasoning in a context where the "right tool" is not available. In this sense, Osiurak et al. (2010) situate technical reasoning in a dialectical process with body action, where humans perceive movement as a problem that they solve with technical reasoning, but whose outcome they need to translate into body action. Based on the Technical Reasoning hypothesis, mechanical knowledge lies at the root of using and re-purposing tools, identifying what is "familiar" in objects, i.e., the abstract technical means, to give them meaning within and beyond learned conventions.

1.5 SUMMARY

In this chapter, I have presented a cognitive model of tool use that relies solely on understanding the relationships between tools and objects, setting aside their manipulation. The Technical Reasoning hypothesis posits that humans possess the ability to reason about interactions among objects using mechanical knowledge (Osiurak et al., 2010) (Figure 1.3). Mechanical knowledge describes technical principles and the technical means offered by objects, through which humans make sense of mechanical interactions such as tool use. Based

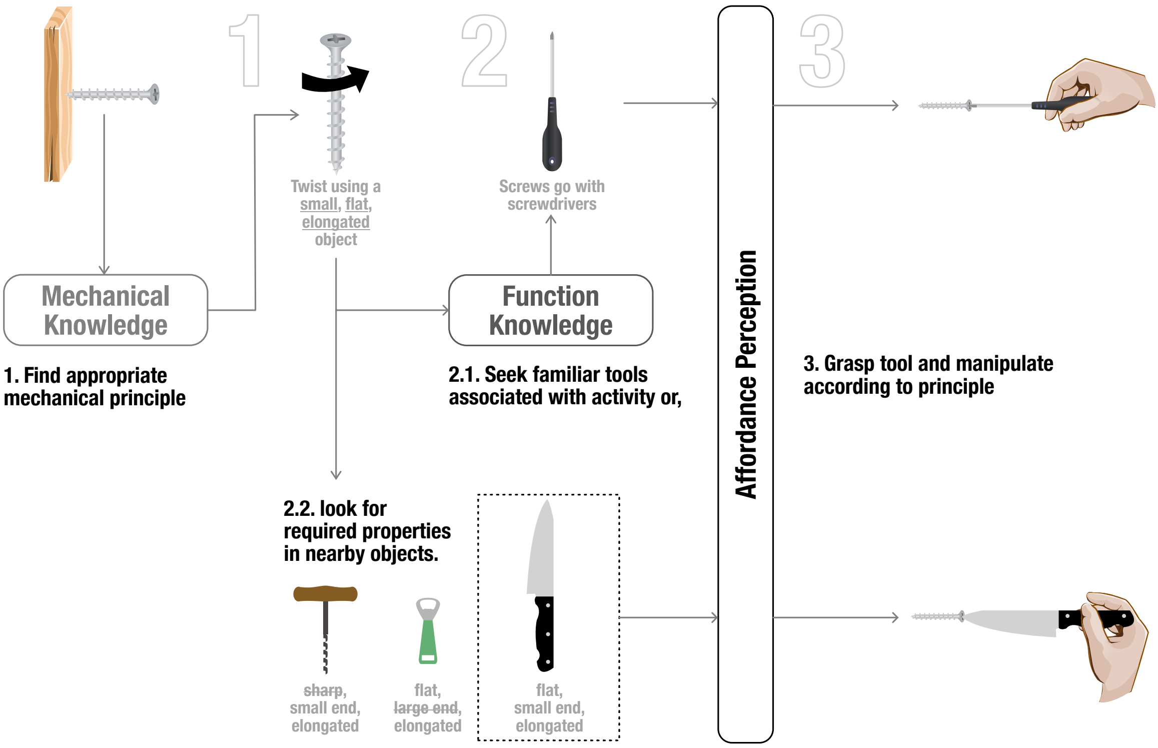


Figure 1.3: The process of choosing a tool to drive a screw according to the technical reasoning model.

on this model, humans acquire abstract knowledge of object properties, later transferring them to other objects that share them for similar mechanical interactions, in particular, resulting in the unusual use of objects. Mechanical knowledge then explains the re-purposing of physical tools, delivering a model for how users identify the properties of a knife that let it drive a screw when there is not a screwdriver at hand (Osiurak et al., 2009). In the following chapter, I switch the focus to computer use, weaving the physical and digital worlds together to explore the notion of digital tools.

2 BACKGROUND

In the previous chapter, I have introduced the Technical Reasoning hypothesis, describing a cognitive mechanism supporting physical tool use. In this chapter, I attempt to thread a theoretical relationship between physical tool use and digital tool use. I start from Activity theory (Leontiev, 1978) and the seminal work by Bødker (1987) bringing in the notion of *computer-based artifacts* as the mediating means of HCI. Next, I look at the work by Béguin and Rabardel (2000) on the incorporation of artifacts into practice, discussing the role of re-purposing in their evolution. From there, I look at examples of interaction styles that build on our understanding of the physical world, notably, Instrumental Interaction (Beaudouin-Lafon, 2000). Finally, I focus on the role of re-purposing in the appropriation of technology (Dix, 2007; Dourish, 2003), looking back at how it relates to the computer-mediated activity perspective (Kaptelinin, 1996) and the evolution of artifacts.

2.1 HUMAN WORK AS A MEDIATED ACTIVITY

The Human Activity theory (Leontiev, 1978) describes human work as an interaction between a subject and an object, *mediated* by means, such as tools, artifacts, instruments, etc. This theory situates human activity in a socio-cultural environment, where knowledge of work practice and tool use circulates between individuals and is influenced by culture enforced by the surrounding community. Notably, this influence is reflected in the goals and techniques associated with tools for specific tasks. For example, we learn that knives are for cutting food and forks are for picking up bites, even though knives can also be used to pick up bites. Furthermore, some cultures enforce knives to be handled with the right hand and forks with the left hand as a form of "etiquette."

Activity theory models work activity as composed of *actions* directed towards *objects* through one or more *operations* directed towards mediating means (Leontiev, 1978). Actions are conscious and associated with the goals of the activity, e.g., slicing. Operations are unconscious, internalized movements upon the mediator, associated with the action, e.g., cutting. Under normal circumstances, mediators are "invisible" to the user because actions (conscious) are focused on the object and not the mediator (Leontiev, 1978). This is illustrated by a skilled user of a tool who operates it comfortably without the need to pay attention to its movement, focusing on whether the transformations applied to the object take place according to the goal. Objects constrain what operations are applicable for an action according to its properties (Leontiev, 1978, p.102), e.g., cutting, poking, scooping, etc. For example, if the actions is slicing fruit, the operations involved in carrying it out with an apple will differ greatly from those carried out on a pineapple, e.g., cutting the former and sawing the latter. However, operations can fail to produce the goal state, bringing the attention to the mediator (Leontiev, 1978), for example to perform recovery actions. In this case, the mediator becomes an object of the activity with actions directed towards it, for example, to sharpen the knife before trying to cut again or to change the way in which it is handled.

Mediators acquire meaning from being incorporated into activities (Kaptelinin, 1996). In other words, we cannot think of a mediator without thinking of an activity with which we associate it (Bødker, 1987). For example, let us use the example of a rock by J. J. Gibson (1986). Let alone, no particular designation comes to mind. However, if used for holding paper, it is a paper weight; if used for throwing at things, it is a projectile; if used for breaking a coconut open, it is a utensil; etc. A similar train of thought arises from the perception of affordances, giving meaning to objects in the environment relative to the intention and body capabilities of the user. However, unlike the perception of affordances, Activity theory posits the meaning of tools as a social construct, as opposed to *only* the product of direct perception. In this sense, Leontiev (1978) describes a dialectical process occurring between tools and practice, where practice is shaped by the tools available, and in turn, tools are transformed and adapted according to practice, for example, when adding a magnet to the tip of a screwdriver to keep screws attached.

In sum, Activity theory posits that human activity is a mediated phenomenon situated in a socio-cultural environment. It establishes that mediating means acquire meaning as part of a dialectical process occurring between individual and social levels (Kaptelinin, 1996). Practice is constrained by the objects, which determine what operations and mediators can produce a desired goal state, much like mechanical knowledge describes the result of mechanical actions between two objects. In light of Activity theory, computers are tools mediating an ever-growing set of human activities, e.g., education, research, banking, finance, etc. In particular, the *user interface* puts forward more than salient physical features around which social conventions arise. It frames virtual relationships between non-physical "objects" that make the workings of computer-based operations and actions. For example, we can analyze text editing commands as mediating user action upon digital text. Therefore, we can understand user interface elements as mediators of human work through an inductive activity-theoretical analysis of computer-based activity.

2.2 COMPUTER-BASED ARTIFACTS

Computers evolved from basic text-based terminals to embed complex GUIs with multiple functions and objects. Additionally, Bødker (1997) observes that computers went from being the trade of skilled workers to now serving a wide spectrum of skill levels among users, powered by the computer's growing ubiquity. Early approaches to HCI regarded users and computers as equivalent "information processing units" (Kaptelinin, 1996) that could be subject to the same form of analysis based on goals and actions directed to the interface, e.g., GOMS (Card et al., 1980). Kaptelinin (1996) posits that computers should be seen as "*a special kind of tool mediating human interaction with the world*", and points at the growing interest for the HCI community to study human-computer interaction as a mediated phenomenon. In this regard, Bødker (1987) presents an activity theoretical analysis based on the notion that users act "*through the interface*" towards their goals.

Bødker (1987) introduces the term *computer-based artifacts* to refer to the mediating means present in the user interface. We can interpret computer-based artifacts as the *functions* that an interface provides. Therefore, interfaces offer multiple artifacts that participate in a variety of activities directed towards different types of objects (Bødker, 1997). For example, when using a modern word processor, we find functions for text editing, images, tables, etc. Interfaces then establish the conditions for how actions are performed on these objects, i.e., they answer "*how* actions can be done" (Bødker, 1987, p.38). As mediators, artifacts are "*something which only reveals itself in breakdowns and situations of reflection*" (Bødker, 1987, p. 38), such as how tools are invisible mediators of physical activity (Leontiev, 1978).

The *object* of a computer-based activity can exist in reality as a physical object or be present only in the user interface. For that matter, Bødker (1987) distinguishes between the presence of objects "in the artifact," "in the use" or "outside the artifact." For example, a digital spreadsheet is *present in the artifact* given that it is both an artifact through which we carry out an activity, and an object on which we focus the activity, because it only exists in the interface as such (Bødker, 1987). A printed letter in a word processor is *present in the use* because, although there is a representation on the interface to which actions are directed, the object of the activity is the printed sheet that will be produced at a later stage (Bødker, 1987). Finally, a digital controller for a MIDI instrument is *present outside the artifact* given that the controls on the interface are attached to the buttons and knobs present on the physical object, letting the user verify the results of her actions on the actual object of the activity.

As mediators, interface functions are the target of unconscious operations to complete the actions until facing breakdowns (Bødker, 1987).

For example, once learned, a computer-savvy user may duplicate a file by executing the cut/copy/paste commands over the file without giving much thought to what these commands entail internally. However, executing these commands encompasses several elements in terms of the stack of operations involved. Bødker (1987) distinguishes between three types of aspects of an operation:

- "Physical:" present in the operation of buttons and other physical objects in the device;
- "Handling:" present in the operation of the application and its functions;
- "Subject/Object Directed:" present in the way the application lets the user switch focus between subjects and objects of the activity.

The physical aspect is concerned with the input devices, such as the keys in the keyboard to copy or the movement of the mouse to select the file icon. The handling aspect entails how the user interface lets the user engage with the artifacts, such as whether the commands are available through shortcuts or menus, a selection of the icon is necessary and how it is done, e.g. clicking or dragging a rectangle around it, etc. Last, the subject/object directed aspect talks to the artifacts and how they support the different objects of the activity. For example, working on a text document, the user may need to switch to work on tables or images within the document, requiring access to different artifacts to carry out the operations that pertain to them. Therefore, letting users focus on the objects and forget about the artifacts relies on the effective integration of these aspects into hardware, software and interface.

The invisibility of an artifact stands from the user "forgetting" that it is in use, for example, to the point where drawing with a stylus or a mouse becomes a matter of changing operations, keeping the actions and goals focused on the interface (Bødker, 1987). In this sense, activity theory closes the gap between physical and digital tools, given that neither is the focus of human activity. Notably, computer-based artifacts are subject to individual and cultural influences through which they acquire their meaning (Bødker, 1987). This includes how the user interface is taught, e.g., how to indent a paragraph in a word processor, and users' developing and sharing their own practices with others (Mackay, 1990). In this regard, Bødker and Klokmose (2016) put forward that there is no useful distinction between physical and digital, citing examples where both levels are rather *entrenched* in an action from the user perspective, e.g., the conceptual distinction between digital typing and physically tapping on the keyboard keys.

2.3 FROM ARTIFACTS TO INSTRUMENTS

Artifacts are introduced in tasks to solve technical problems (Béguin and Rabardel, 2000). Physical artifacts can be used in a multiplicity of tasks, sometimes in unusual ways, such as using a knife to turn screws. Despite being able to use knives as screwdrivers, people do not usually look for a knife when they want to drive screws, nor associate knives with that activity. In this regard, Béguin and Rabardel (2000) posit that artifacts carry a psychological and social *function* that ties them to certain activities, giving more substance to the *meaning* acquired by tools in use (Kaptelinin, 1996). The authors then employ the concept of *instrument* to distinguish an artifact by its given function in an activity. As such, a knife used for cutting vegetables is a *slicing instrument* while the same knife used for peeling is a *peeling instrument*.

The use of artifacts creates new perspectives about existing problems that allow for the evolution of practice and, in turn, that of the artifacts (Béguin and Rabardel, 2000). Let us illustrate this with the use of a familiar artifact/tool. A typical fixed-size wrench is made of a resistant and heavy material such as iron or steel, so that we can exert a strong torque force to loosen bolts without breaking the wrench apart. An adjustable wrench can be seen as a step in the evolution from a fixed-size wrench, stemming from the need to work on multiple bolt sizes without switching artifacts. The heavy construction of wrenches led to using them to pound on rusty bolts to loosen them before applying the torque force, which probably motivated the development of wrenches with a flat side for hammering. In summary, artifacts are incorporated into practice, leading to its evolution and that of the artifacts themselves.

Instruments are created and evolve with the use of artifacts. Furthermore, Béguin and Rabardel (2000) associate the evolution of instruments with the use of artifacts for purposes other than those assigned to them, e.g., the knife as a screwdriver. They illustrate how modern (digital) technology is subject to this phenomenon as well, exemplified with how pilots purposefully give "wrong" inputs to an airplane's computer in order to induce the autopilot system to act according to their intentions (Béguin and Rabardel, 2000). Rather than focus on the deviant aspect of this use, the authors regard it as individuals making their own instruments, characterizing an *instrumental genesis* process (Béguin and Rabardel, 2000). Instrumental genesis occurs through two complementary processes: the incorporation of artifacts into new practices—the *instrumentation* process—, and the evolution of artifacts themselves—the *instrumentalization* process.

Users follow "utilization schemes" (Rabardel, 1995) which define ways in which to engage in using artifacts, e.g., swinging a hammer, turning a screwdriver, etc. Utilization schemes are acquired from individual experience with artifacts as well as through passing knowledge among peers in society. Instrumentation then is concerned with the evolution of utilization schemes and their application to new artifacts. For example, hammering is a utilization scheme mainly associated with hammers. A user can adapt her technique with a hammer as she develops the skill, thus evolving the hammering scheme. However, she can then apply the hammering scheme to use a wrench as a hammer, thus developing a new practice around wrenches. In both cases, instrumentation takes place through the evolution of utilization schemes within and between artifacts. In particular, instrumentation situates the re-purposing of artifacts in the evolution of practice, resonating with how mechanical knowledge is transferred between physical object interactions.

Instrumentalization centers on the assignment of function based on the artifact's properties. Béguin and Rabardel (2000) posit that objects' physical properties, e.g., shape, weight, etc., become associated with utilization schemes, becoming themselves recognizable features in other artifacts that deem them suitable for new utilization schemes. For example, the weight and elongated shape of a wrench may call for the use of a hammering scheme (Béguin and Rabardel, 2000); the thin arrow-shape tip of a knife may call for the use of a screwdriving scheme; etc. Instrumentalization does not take place in re-purposing artifacts, but rather in the evolution that takes place as a consequence of the application of utilization schemes beyond the artifact's original design. Instrumentalization then becomes evident in changes and developments of new artifacts, such as the wrench with an added rectangular tip for hammering bolts.

Summing up, *instrument* implies a socio-cultural meaning of an *artifact* in a way that they are not interchangeable (Béguin and Rabardel, 2000). Particularly, instrumentalization processes shed light on how the perceived physical properties of objects make artifacts recognizable as suitable for a given use. Initially, this can be thought of as the perception of affordances for tool use. However, this also resonates with how mechanical knowledge describes the perception of object properties and their relationships in order to make sense of mechanical interactions (Osiurak et al., 2010). Instrumental genesis is rooted in the notion that "*the design process [of instruments] continues throughout use*" (Béguin and Rabardel, 2000), referencing the dialectic process occurring between tools and practice (Leontiev, 1978). In this work, the authors aim at providing guidelines for the design of digital interactions based on instrument-mediated activity, built on notions of human activity in the physical world.

2.4 FROM PHYSICAL TO DIGITAL ENVIRONMENTS

Interaction styles establish the way in which users interact with digital objects. For example, the command-line style of interaction specified a text-based language to express actions and target objects. With further advancements in computing power, GUIs became the norm, and the WIMP interaction style took over, bringing the windows, icons and menus present in a multiplicity of contemporary digital environments such as desktop or mobile user interfaces. However, the ubiquity of computers and their evolution in computing power and uses, as well as in the diversity of devices, keep pushing the need for interactions that go beyond the traditional WIMP style into a *post-WIMP* era (Van Dam, 1997). Raskin (2000) calls attention to the need for more "humane" interfaces, engaging with interaction styles and models that account for human capabilities. This suggests that observing our physical reality and the way in which we interact in it could offer insights to leverage human ability in the interaction within digital environments.

Direct Manipulation styles (Shneiderman, 1983) bring aspects of the interaction with physical objects to digital environments. In a Direct Manipulation interface, objects support actions on themselves, offering immediate and continuous feedback of the results to the user as she performs an action, e.g., resizing by dragging its corners. The key benefit of this interaction style is the sense of doing things as how they are already done in the physical world (Hutchins et al., 1985). Shneiderman (1983) posits that the physicality of Direct Manipulation engages users in problem solving techniques—such as that with physical objects (Duncker and Lees, 1945)—, observing that "*suitable representations of problems are crucial to solution finding and to learning.*" However, as Hutchins et al. (1985) point out, its advantage is also one of its limitations, because although it helps ease the learning curve, it also does not leverage the power of computers to, for example, automate repetitive actions.

Reality-based Interaction (Jacob et al., 2008) proposes a framework for interaction design that is based on users' existing skills from physical reality. The authors draw on the observation that contemporary interaction styles push the boundaries of traditional Direct Manipulation, "*using actions that correspond to daily practices within the non-digital world*" (Jacob et al., 2008). Reality-based Interaction posits *themes* based on human skills used in the physical world, such as knowledge of *naïve physics* or *environment awareness and skills*, and illustrate them with examples of their relevance for interaction design. For example, users' knowledge of *naïve physics* can be leveraged for interactions where objects offer resistance to be activated, such as the *pull-to-refresh* technique used in mobile applications. *Environment awareness and skills* can let users understand cues of directions or limitations in an interface, such as using grids to delimit the walk-able space in a virtual reality environment.

In the same regard, Blended Interaction (Jetter et al., 2014) is based on *conceptual integrations* or *blends* of knowledge of the physical world and digital environments. The authors define conceptual integrations as a cognitive mechanism in charge of understanding new concepts based on finding similarities with past experience. Unlike Reality-based Interaction (Jacob et al., 2008), Jetter et al. (2014) argue that the users' experience interacting within digital environments should be considered as a source of experience in human reality, feeding the conceptual integrations when making sense of other aspects of reality, such as new digital environments. The notion of finding matching similarities resonates with how technical reasoning describes the process of analogical reasoning to transfer mechanical knowledge (Osiurak et al., 2010). However, the authors do not focus specifically on a mediated perspective of user interfaces, or on any particular interaction among digital objects.

These represent examples of interaction models, styles or frameworks that build on how humans interact in the physical world. However, these models do not take advantage of the fantasy-like reality represented by digital environments (Hutchins et al., 1985). In this regard, the work by Bødker (1987) looks at the user interface in terms of its artifacts and object representations, and fits it in the organization of the physical reality described by Activity theory. However, physical artifacts can become the object of an activity, for example, facing "breakdowns" or being part of an instrumental genesis process (Béguin and Rabardel, 2000). A more interesting interaction model could arise from making artifacts more prominent objects of the interface, giving users the potential to explore and make them their own instruments.

2.5 INSTRUMENTAL INTERACTION

Beaudouin-Lafon (2000) introduces Instrumental Interaction, a post-WIMP interaction model combining the power of Direct Manipulation and the mediated nature of human activity in the physical world. An instrumental interaction is based on *Domain Objects*, i.e., the objects of interest in the environment, and the use of *Interaction Instruments* (Beaudouin-Lafon, 2000). Domain Objects are the focus of the interactions—the equivalent of target objects in Activity Theory—, carrying attributes that users can edit. Attributes can be simple values attached to the object, such as its position on a 2D canvas, or else can be domain objects on their own, grouping simple values that describe them, such as texture properties or animation descriptors. For example, working in an image editing environment, the user may turn her attention to the ordering of the layers and furthermore, the blend mode of one layer in particular, thus switching focus between multiple domain objects. Interaction Instruments mediate the interaction between user and domain objects, translating user actions into commands. For example, a scroll bar takes the role of a command to change the position of the content of a target window, moving it in one direction; a highlighter tool represents a command to change the highlight property of target text; etc. Instruments can also create and destroy new objects, for example, inserting rectangles by dragging across a canvas, duplicating existing shapes or erasing pixels in a raster image with an *Eraser Tool*.

Users *activate* instruments by either interacting with their representation on screen (a spatial activation)—such is the case of a scroll bar—or by selecting the instrument with a previous action (temporal activation), e.g., selecting a tool from the toolbar. In both cases, the instrument is *attached* to a physical input device, e.g., a mouse, setting the instrument "under the user's control" (Beaudouin-Lafon, 2000, p.449). Additionally, instruments can become domain objects when not in use, thus becoming the target of user interactions (Beaudouin-Lafon and Mackay, 2000). For example, a formatting instrument for quick text styles could be the target of other formatting actions to create or modify existing styles.

Instrumental interactions are split into user-instrument interactions and instrument-domain object interactions (Beaudouin-Lafon, 2000). The user-instrument interaction starts at the physical *action* that the user performs on the instrument, e.g., drag the scroll bar's thumb, followed by the *reaction* of the instrument, e.g., the thumb moves. The instrument-domain object interaction is concerned with executing the *command* on the domain object generating a *response* on the latter, e.g., the window's content moves. This may involve an additional step to the user-instrument layer with the instrument providing *feedback* to the user. The idea behind this is to keep the user informed by following the schema of Direct Manipulation through the continuous representation of the effects of user actions.

Instrumental Interaction draws on the users' interaction with objects in the physical world, where they "*rarely fingerprint but often use pens and pencils to write*" (Beaudouin-Lafon, 2000). Additionally, it provides a model for interfaces that follows the notion of human activity through mediating means (Bødker, 1987; Leontiev, 1978). An interesting perspective is the development of use and the cultural influence in it. Mediators need to be understood in the context of an activity (Kaptelinin, 1996), where both the mediator and the activity interact in a dialectical on each other. For example, turning the color picker tool used for text colors into a tool for changing the color of any element that carries such property. A complementary analysis of the development in the use of artifacts could help us understand how to design instrumental interactions that model and leverage the evolution of physical artifacts.

2.6 DIGITAL TOOL RE-PURPOSING

Dourish (2003) defines *appropriation* as "*the way in which technologies are adopted, adapted and incorporated into working practice.*" In this regard, Mackay (1990) argues that users first adapt to the constraints imposed by technology and later adapt it to their needs, in a *co-adaptive* phenomenon. A recurrent practice for supporting the adaptation of software has been to offer *customization* options based on pre-determined aspects of the system that can be tailored to the individual and organizational needs (Dourish, 2003). However, Dourish (2003) separates the "explicit reconfiguration" involved in customization from the *unanticipated* use that characterizes its appropriation. This implies that the design for appropriation should leverage the spontaneous adaptations of system functions rather than pre-establishing which tailoring options will be available for it.

Re-purposing is a spontaneous event of tool use that deviates from the tool's conventional uses, defined by technological or social constraints (Béguin and Rabardel, 2000). Dix (2007) draws a parallel between the deviant use of a screwdriver as a paint can-opener and using an email server for file-sharing within an organization, pointing at both as examples of appropriation of technology. The sense of "ownership" arising from appropriating artifacts (Dix, 2007) suggests an inherent long-term relationship originating in the continued re-purposing practice. After all, this is what the instrumental genesis process describes (Béguin and Rabardel, 2000), where re-purposing a tool one time is the first step towards instrumentation, but does not end in instrumentalizing the tool and giving it a new meaning. Therefore, a nuanced interpretation of the example given by Dix (2007) should account for the long-term aspect of appropriation, rather than the single spontaneous occurrence of re-purposing.

Appropriation and instrumental genesis then would originate in re-purposing and consolidate through the long-term practice that morphs into a new meaning for the tool. Finding a common thread between the cognition behind the re-purposing of physical and computer-based artifacts could shed light on how to leverage this ability in interface design. The Technical Reasoning hypothesis offers a compelling model for how physical tools are re-purposed, based on mechanical knowledge to understand the relationships between object properties and mechanical principles, and analogical reasoning to transfer this knowledge to new situations. Arguably, users learn causal relationships between digital objects, and previous work has focused on the analogy as a tool for learning user interfaces (Carroll et al., 1988) and on the consistency of interactions in order to leverage the ability to reason analogically (Rieman et al., 1994). Therefore, the re-purposing of digital tools seems a good place to start in studying the relevance of the Technical Reasoning hypothesis for HCI.

2.7 SUMMARY

Activity theory posits that human interaction in the world is a mediated phenomenon between the subject and the object of an activity, subject to individual and communal experience (Leontiev, 1978). Mediators can only be understood in terms of their function in human activity (Kaptelinin, 1996). For example, a stone becomes a paper weight when it is adopted as such within an activity. Bødker (1987) posits an activity theoretical analysis of HCI, observing that user interfaces mediate human work through *computer-based artifacts*. As such, application functions and interface objects are seen as the mediators of interactions with both digital (internal) and physical (external) objects.

Béguin and Rabardel (2000) posit that artifacts are subject to instrumental genesis processes based on re-purposing them, through which they acquire their meaning as they are progressively incorporated into activities. Some existing interaction styles aim at leveraging human capabilities for interaction in the physical reality (Jacob et al., 2008; Jetter et al., 2014; Shneiderman, 1983), but they do not leverage the mediated perspective of offered by Activity theory. On the other hand, Instrumental Interaction (Beaudouin-Lafon, 2000) puts interface functions as both artifacts and objects, modeling interactions with objects as mediated by digital instruments.

Mackay (1990) presents evidence that users adapt technology, sharing custom developed practices and incorporating those developed by others. Dourish (2003) and Dix (2007) put the emphasis on the re-purposing of digital artifacts as evidence that users appropriate technology for their own needs. Previous work has focused on analogies and consistency as ways towards learning how to interact with interfaces (Carroll et al., 1988; Rieman et al., 1994). In this regard, the Technical Reasoning hypothesis offers a model based on analogical reasoning about interactions among physical objects in order to transfer the adequate knowledge to reach a goal state.

In the following chapter, I investigate users re-purposing digital tools to argue about their thought process and discuss about its compatibility with mechanical knowledge and technical reasoning. Such evidence could open the way to designs that leverage our understanding of physical reality, offering support to interaction models that follow an instrument-mediated perspective.

3 EXPLORING TECHNICAL REASONING IN DIGITAL ENVIRONMENTS

As I have cited it before, tool use is ingrained in our interaction with the physical world (Osiurak et al., 2010): Physical tools mediate our interactions with the environment, becoming extensions of our hands (J. J. Gibson, 1986), and we routinely use objects as tools beyond their assigned function (Dix, 2007), e.g., using a knife as a screwdriver, a phenomenon we associate with *tool re-purposing*. I have introduced the Technical Reasoning hypothesis (Osiurak et al., 2009) in Chapter 1, which posits that human tool use is based on the ability to reason about the mechanical interactions among physical objects. It models human tool use as a product of matching *abstract* knowledge about mechanical principles and properties of objects, e.g., a *sharp* blade can cut through an orange's *soft* skin. Technical reasoning therefore provides an elegant model to understand the re-purposing of physical tools as the transfer of abstract mechanical principles to other interactions among analogous abstract *properties*.

Is this model also at play when interacting in digital environments? Computer applications often use a tool metaphor, whereby the user can select the best tool for the task from a tool palette and apply it to objects of interest, and computers have been referred to as "tools for the mind."[1] Beaudouin-Lafon (2000) introduced Instrumental Interaction, an interaction model based on digital tools, called instruments, that mediate the interaction between the user and domain objects in the system. The HCI literature also features examples of users re-purposing tools in ways unexpected by their designers (Dix, 2007). However, while these observations seem to support a parallel between physical and digital tool use, I also observe that digital tools often work with a limited set of target objects, hindering users from taking advantage of the flexible nature of digital interactions. For example, the *Eraser* tool in Adobe Photoshop is designed to delete pixels —by making them transparent— but produces no effect on vector-based objects, which must be deleted with a different tool.

In this chapter, I present a revised version of an article being published at ACM CHI 2022 addressing the question of whether technical reasoning can be used to study digital tool use and re-purposing. While tool re-purposing has been discussed and observed

1 Steve Jobs referred to the computer as "the equivalent of a bicycle for our minds" in the documentary film *Memory & Imagination* by Michael R. Lawrence

in HCI (Ciolfi Felice et al., 2016; Dix, 2007; Han et al., 2020), its underlying cognitive mechanisms have not been studied. Likewise, the Technical Reasoning hypothesis has not been brought forward as explanatory theory in HCI before. A better understanding of digital tool re-purposing could offer support to design interaction techniques based on digital tools that users can own, towards less application-centric environments. This chapter's contributions are twofold: it introduces an experimental design to evaluate users re-purposing digital tools, and brings forward the Technical Reasoning hypothesis as a model to explain this phenomenon. For this work, I implemented an experimental text editor controlling conditions to induce unusual uses of its commands, based on an experimental design around the unusual use of physical tools by Osiurak et al. (2009). We studied a group of participants completing a task with this digital environment to analyze their thought process framed by the Technical Reasoning hypothesis. Additionally, we compared their performance in usual and unusual uses of commands to self-reported *experience with text editing* and *creative personality* scores so as to understand their role in re-purposing digital tools. I conclude this chapter with a discussion of the results in light of the Technical Reasoning hypothesis.

3.1 RELATED WORK

This section begins by characterizing digital tool use and how it is addressed in the HCI literature. Next, it includes a review of studies that establish cognitive similarities between physical and digital tool use and discusses them from a re-purposing perspective. It then presents evidence of digital tool re-purposing from the HCI literature. Finally, since tool re-purposing can be seen as a case of creative problem solving, it includes a review of related work from the literature on creativity.

3.1.1 Digital Tool Use

As discussed in Chapter 2, HCI has a long history of discussing digital tools as mediating means for the interaction between the user and target objects, in particular with Bødker's early work on the socio-cultural approach to HCI, incorporating activity theory to user interface design (Bødker, 1987). Taking this wider perspective, Bødker (1987) characterizes digital tools as artifacts that mediate our activity within computers. On a similar note, Béguin and Rabardel (2000) discuss the process of incorporating physical artifacts in terms of an *instrumental genesis* occurring through complementary instrumentation and instrumentalization processes. In particular, the instrumentalization process, "*extends the artifact's intended use,*" regardless of its predefined purpose, e.g., when using a knife as a screwdriver. Additionally, Instrumental Interaction (Beaudouin-Lafon, 2000) is both an analytic and generative model for post-WIMP interfaces where user interactions with domain objects are mediated by digital instruments. This model draws on the conceptual similarity between digital and physical tool use. While a mediated perspective of instrumental interaction can provide insights about its relevance as a model to support instrumentation and instrumentalization processes, a cognitive framework would provide additional terms and concepts to model the mechanisms behind the spontaneous deviation from usual uses of artifacts, i.e., their re-purposing.

Other work has focused on *perceptual* similarities between physical and digital tool use. For example, Bérard and Rochet-Capellan (2015) study the sensorimotor similarity between physical and digital interactions with a target-acquiring task designed to measure the transfer of a motor skill required for a physical task after training with one of three digital setups (touchscreen, mouse, trackpad). The authors find that the group that trained with the touch screen experienced a significant transfer from the digital setup, while the groups that trained with the mouse or trackpad did not. In another study, Bergström et al. (2019) assess tool extension (Cardinali et al., 2009) in a digital environment through the difference in response time between congruent and incongruent simultaneous visual and tactile stimuli in a pointing task using a trackpad or a mouse. The authors conclude that participants experienced tool extension using both the trackpad and the mouse. Singley and Anderson (1987) study negative transfer of knowledge between two versions of the same text editor, differing in the shortcuts associated with the same commands. Their results show significant positive transfer between the editors despite differences in the command layout. However, the authors argue that "*declarative knowledge, [i.e., conscious knowledge] of a special sort must have contributed to transfer.*" The first two studies provide evidence that users experience effects similar to physical tool use in a digital environment, in terms of sensorimotor knowledge. The last study provides evidence of transfer of knowledge to use digital commands across digital environments. However, they do not address the cognitive processes leading to selecting and using digital tools to interact with target objects.

3.1.2 Re-purposing of Digital Tools

Dix (2007) frames re-purposing as a form of improvisation to "work with what we have to hand." As an example, the author compares using an email server for sharing files within an organization to using a screwdriver for opening paint cans. We also find references to re-purposing in digital environments under concepts such as *customization* (McGrenere, 2002), *appropriation* (Dix, 2007; Dourish, 2003), *co-adaptation* (Mackay, 1990) or *ambiguity* (Gaver et al., 2003), to name a few. Work on these concepts focuses on understanding users' practices around technology (McGrenere, 2002) and the need to adapt digital tools to their activity (Mackay, 1990). However, while these works address the design of systems that offer flexibility for its users, they do not focus on the cognitive abilities that make users leverage such flexibility.

Re-purposing is also observed anecdotally in some user studies. For example, during a user study of *StickyLines* (Ciolfi Felice et al., 2016), a graphical editor that manages shape alignment through persistent, "tweakable," magnetic guidelines, the authors observed that some participants spontaneously used guidelines as a tool for grouping shapes rather than as an alignment tool. Similarly, in a study of *Textlets* (Han et al., 2020), a system that supports the reification (Beaudouin-Lafon and Mackay, 2000) of text into persistent objects with various behaviors, the authors report a participant discussing a search Textlet to highlight occurrences of words that he should *not* use. We noted that in both these cases, observations of tool re-purposing occurred spontaneously and in open-ended tasks that gave participants the liberty to use the tools in that way.

In summary, while we find evidence of re-purposing strategies where digital tools are used in unexpected ways, these have not been studied systematically and we still do not understand why and how users come up with these unusual uses of digital tools.

3.1.3 Factors in Tool Re-purposing

Fitts and Posner (1967) posit that motor skill acquisition goes from a stage of high consciousness—for example, about the manipulation of a tool—to one where no conscious cognitive effort is necessary to perform an action. This is in line with the distinction made by Anderson (1983) between *declarative knowledge* and *procedural knowledge*. According to Anderson (1976), all knowledge is acquired in declarative form and can gradually become "proceduralized" so as to perform actions in a direct way, without interpretation or conscious effort, i.e., procedurally. If computer users operated based *only* on procedural knowledge, they would have difficulty finding alternative ways to complete tasks or devising new uses for software. Similarly, Activity theory (Leontiev, 1978) distinguishes between *actions*, which are conscious, and *operations*, which are subconscious: actions become operations through practice, but operations can become actions when a problem occurs. This is consistent with the work by Ericsson et al. (1993), who present evidence that "deliberate practice" over an extended period of time amounts to expert performance. Since experience with a tool amounts to skillful use, it is possible that it would play a role in tool re-purposing.

However, Duncker and Lees (1945) show evidence that humans experience *functional fixedness*, i.e., bias from knowledge of the assigned functions of tools, when facing a novel task requiring the creative use of a familiar object. That is, humans unconsciously associate functions to objects and therefore require a less conscious effort to use them, which becomes the source of an unconscious bias in creative problem-solving situations. Therefore, a tool user is likely to experience functional fixedness when required to use a familiar tool in an unusual way, i.e., to re-purpose it. This phenomenon has been acknowledged in HCI, for example, in Oh and Findlater's work on gesture customization (Oh and Findlater, 2013). Therefore we can expect functional fixedness to hinder tool re-purposing for experienced users.

Furthermore, re-purposing can be studied as a creative solution to a tool-based problem (Vass et al., 2002). Coughlan and P. Johnson (2009) argue that novel outcomes in creative settings are "produced from novel processes and tools" for which "the malleability of tools and their ability to be appropriated is key." For example, in the *StickyLines* and *Textlets* studies mentioned earlier, the participants did not get to train or familiarize themselves with the environment for more than a few minutes, yet some spontaneously found creative solutions by re-purposing the tools. Therefore we can expect creative individuals to exhibit tool re-purposing behavior. This also suggests that prolonged experience and practice with tools may not be a requirement for spontaneous re-purposing.

To the best of our knowledge, cognitive models of re-purposing, appropriation or creative use of tools had received little attention in HCI. We are interested in the Technical Reasoning hypothesis because it explains tool re-purposing without the need for any manipulation knowledge. Therefore, in this work, we assumed the validity of this hypothesis for human use of physical tools and sought to assess its applicability to digital tool use.

3.2 STUDY: COMMAND RE-PURPOSING IN TEXT EDITING

We wanted to observe whether and how participants elicit mechanical knowledge about text editing. By focusing on declarative knowledge about a task, i.e., what participants express about their actions, we can analyze the motivations and reasoning towards their use of commands, in particular, unusual uses. For this purpose, we designed a simplified text editing environment where an experimenter controls the availability of its commands, e.g., insert characters, paste text, or change color. Participants perform an identical task repeatedly but with an increasingly limited set of commands. By progressively reducing the commands available to solve the same task, we wanted to induce participants to find alternative techniques that rely on the remaining commands, which we expected would lead them to re-purpose one or more of them.

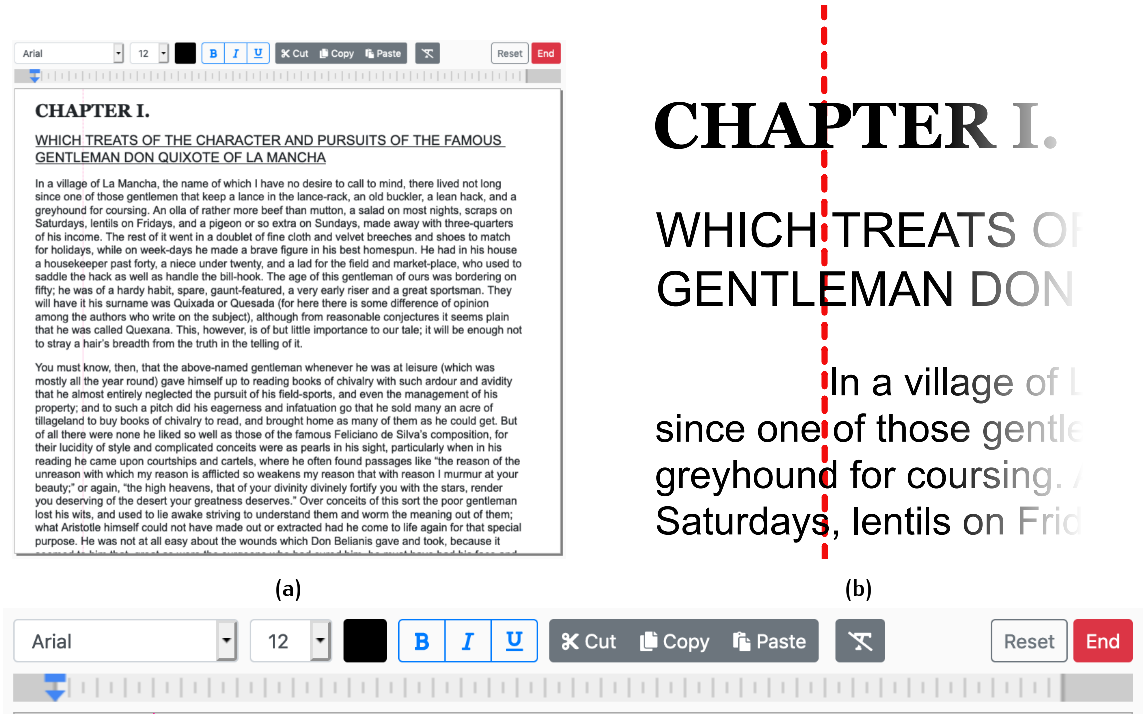
Our design borrowed from the "*Unusual Use of Objects Test*" by Osiurak et al. (2009), which asks participants to carry out tasks under conditions that force them to re-purpose tools, e.g., asking them to eat yogurt with a fork, which can only be performed by using its handle as a spoon. We expected participants to use familiar techniques before exploring and effectively re-purposing commands. We also expected experience and creativity traits to be associated with finding unusual techniques to complete the task. Finally, we expected participants to elicit knowledge from past experience with other digital environments to find ways to re-purpose text-editing commands.

3.2.1 Design of the Task

We sought a task for which participants were unlikely to find a direct equivalent in the physical world, i.e., a "purely digital" task with low risk of transfer from experience with physical tools. Arguably, text editing falls under this condition nowadays, given that most computer users have experience typing on keyboards, but few (if any) have experience with physical typesetting. This led us to design a task that requires participants to set the indentation of a paragraph, following a visible guideline shown at a specific distance from the left margin (Figure 3.1b). Participants have access to basic formatting, layout and editing commands (Figure 3.1c) to complete the task. As the session progresses, the experimenter disables these commands to force participants to devise new techniques to complete the task, based on the remaining commands. Our protocol follows a similar principle to that used by Maier (1931), in which participants had to repeatedly demonstrate alternative techniques to solve the same problem, performing unusual uses of objects available in the environment. More recently, K. P. O'Hara and Payne (1999) offer an example of a design controlling the availability of a command, disabling it for a fixed amount of time for certain participants.

To design the set of commands, I ran 6 pilot testing sessions with participants recruited from our lab. I asked each participant to show me as many alternative techniques as they could to complete the task. We identified 5 recurring approaches, listed in Table 3.1. Every technique is coded with the primary command that it uses. For example, `Color` consists of inserting arbitrary characters and making them invisible by coloring them the same as the page's background color. Its primary command is therefore the *Color* command.

I observed uses of both the Tab key and the Spacebar during pilot testing. However, because of the similarity between these approaches,



(c)

Figure 3.1: (a) The editor in its initial state. (b) The goal state showing the indentation of the first sentence of the paragraph with the length indicated by the guideline. (c) The complete toolbar and ruler when all commands are enabled.

i.e., both insert a blank character to indent, we designed the tabulator character to be wider than the required indentation. The result is an indentation larger than necessary, making the Tab key a poor solution and ultimately forcing participants to resort to the Spacebar for a more precise technique.

3.2.2 Participants

I recruited 18 adult computer users but had to discard data from two participants: **P6** faced a code defect that led her to finding techniques that were not consistent with what is possible with a standard word processor; I wrongfully gave **P8** instructions that led him to understand what type of technique would produce interesting results for the study, thus making the entire session invalid. Of the 16 remaining participants, 9 self-identified as male, 7 as female. 14 were between 30 and 39 years old, one between 18 and 29 years old, and one between 50 and 59 years old. Participants' backgrounds included 5 in computer science (**P2, 5, 10, 12 and 14**), 3 in social sciences (**P3, 9 and 11**), 2 in sales (**P15 and 17**), 2 in architecture (**P1 and 18**), 2 in graphic design (**P4 and 16**), 1 in economics (**P7**) and 1 in mathematics (**P13**).

Table 3.1: Techniques we expect participants to use, identified during pilot testing. The names correspond to the primary commands for the technique to succeed. Techniques marked *Familiar* involve familiar use of the primary command, while those marked *Re-purpose* involve re-purposing it. The one-letter code is used as a shorthand to identify each technique.

Code	Name	Use	Procedure
`R`	`Ruler`	Familiar	Use the paragraph indentation control to set the indentation of the first line as necessary.
`S`	`Spacebar`	Familiar	Position the text caret at the beginning of the paragraph and use the Spacebar on the keyboard to insert spaces that push the sentence to the right.
`T`	`Tabulator`	Familiar (poor)	Position the text caret at the beginning of the paragraph and use the Tab key to insert a tabulation character to push the sentence to the right. The tabulator size is designed to *not* match the indentation of the goal state and therefore it is considered to produce a bad result.
`C`	`Clipboard`	Re-purpose	Select and copy an existing space or series of spaces, position the cursor at the beginning of the paragraph and paste the copied spaces, reproducing the effect of the Spacebar.
`X`	`Color`	Re-purpose	Position the text caret at the beginning of the paragraph and insert arbitrary characters until the beginning of the sentence is at the desired position. Then, color the arbitrary characters with the same color as the document background to make them invisible to the eye.

3.2.3 Setup

The setup is designed to carry out the study remotely due to the Coronavirus pandemic of 2020. I developed an experimental text editor (Figure 3.1a) that runs in web browsers with a JavaScript interpreter, with functionality based on that found in commercial text editors. The application was hosted on a virtual server running on our lab's infrastructure. A back-end system let me control the editor functions available to the participant.

The editor interface includes a toolbar with widgets to change the text's font face, size, color and variations (bold, italic and underlined), as well as buttons to cut, copy or paste a selection, and one to clear the selection's format, i.e., set it back to the default style (Figure 3.1c). Underneath the toolbar, a ruler lets users change the page margins, current paragraph's margins and current paragraph's indentation (Figure 3.1c). Operation requires a keyboard and a pointing device such as a mouse or track pad; the environment is not designed for touch interfaces. Additionally, the editor supports undo/redo using keyboard shortcuts and the browser's or operating system's contextual menu. Cut, copy and paste also work using the traditional keyboard shortcuts (*Ctrl+X*, *Ctrl+C* and *Ctrl+V*).

The initial state contains a pre-loaded text over which the participant performs the task (Figure 3.1a). Every new trial presents the editor in this state. Two separate buttons to the right of the toolbar allow the participant to *Reset* the document to the initial state and *Finish* the ongoing trial to jump to the next one or finish the session when they have reached the last trial (Figure 3.1c).

3.2.4 Procedure[2]

I generated a unique URL for each participant and shared it via email so they could run a local copy of the editor on their browser to collect the associated data in the server for later analysis. Using a video conferencing app with support for screen sharing, the participant shared a video stream of the browser window where they loaded the environment. Participants started by filling out a pre-session questionnaire about their experience with text editing. Next, I introduced the task and the editor interface, containing a document with a title, subtitle and several paragraphs. The participant was asked to indent the first paragraph of the document as indicated by the guideline (Figure 3.1b), using any command supported by the editor. After the task was considered complete, the participant finished the trial and answered the post-trial questionnaire while I added a new trial to the session. The participant received identical instructions to complete the same task in every trial.

2 This protocol was approved by Inria's Institutional Review Board (COERLE).

Before starting the next trial, I identified the technique used to complete the task in the previous trial (see Table 3.1) and used the back-end control of the session to disable the key command of the technique for all subsequent trials. For example, if trial 1 was solved with the technique *R*, i.e., the participant's technique was based on using the ruler, the ruler was disabled for every trial $n > 1$. The participant saw a list of the commands that would be disabled in the editor and pressed a *Continue* button to start the next trial. This list was not shown in the first trial because all the commands were enabled (Figure 3.1c). History commands, i.e., undo and redo, were not controlled and thus were always available.

Participants were asked to think aloud (Hoffman, 1989) throughout the entire session whenever they were working in the editor. A participant ended a trial by pressing a *Finish* button on the screen when they considered the task complete. The number of trials at the end of a session is the same as the number of different techniques that the participant used to solve the task before a 30-minute countdown runs out. The countdown only ran when the participant worked on the editor, and was stopped between trials. Participants could end a trial before completing the task, thus giving up and finishing the session.

At the end of the session, participants answered a demographics questionnaire including their age range and gender. A post-session-questionnaire assessing creative personality was sent within 24 hours after the session. This delay was meant to minimize the effect of the participant's performance during the session on the self-perception of their creative personality. We decided not to use the questionnaire before the session to avoid priming participants about our interest in creative outcomes.

3.2.5 Data Collection

I collected answers to pre-session questionnaires, post-trial questionnaires, demographics questionnaires and post-session questionnaires (see Section a.1). The pre-session questionnaire was designed to assess the participants' self-reported *experience* with text editors, problem solving attitudes towards software and signs of appropriation of text editing functions using 5-point Likert-type questions. It was used to calculate a text editing experience score for each participant. The post-trial questionnaire was used in connection with the notes from the verbal protocol to assess the participant's thinking process. It measured the self-perception of the quality of the result with a 5-point Likert-type item and contains a series of Yes/No items to characterize the thought process towards the technique. The demographic questionnaire collected the age and gender of the participants. The post-session questionnaire collected self-reported measures of *creativity* as a personality trait using the complete list of standardized 5-point Likert-type items from the *Originality/Creativity* scale of the International Personality Item Pool (IPIP) (Goldberg et al., 2006). We used this questionnaire to calculate a creativity score for each participant.

I took notes of the participants' verbal protocol and my observations of the participants' screen via screen sharing. I recorded audio and video of the participant's screen. I kept logs of keystrokes and changes to the sample document to build an event log of each trial (see Table a.4 in Section a.2 for a full list of command types). All the data was referenced by participant number.

3.2.6 Data Analysis

Using the questionnaire responses, we calculated the participants' text editing *EXPERIENCE* and *CREATIVITY* scores. Creativity items were valued based on the IPIP's scoring instructions, where individual Likert-type scale values are added to calculate the general score. We used the same approach to calculate the *EXPERIENCE* score.

For each participant's trial, we designated its *TECHNIQUE* from the considered levels `Spacebar`, `Ruler`, `Tabulator`, `Clipboard` and `Color`, corresponding to the technique used to complete the task during the trial. We used this value to count the number of participants that used each technique as well as gather information regarding the participants that used re-purposing techniques (`Clipboard` and `Color`).

Additionally, we measured *#SOLUTIONS* as the number of successful techniques used throughout the session. For example, if a participant succeeded performing `Spacebar`, `Ruler`, `Clipboard` and `Color`, $\#SOLUTIONS = 4$ (see Section a.3 for the complete results). I produced code for the statistical analysis and generation of plots as we went over the process. We investigated the associations of *#SOLUTIONS* with *EXPERIENCE* and *CREATIVITY*. We then performed logistic regressions to study whether *EXPERIENCE* and *CREATIVITY* associated with any of the re-purposing levels (`Clipboard` and `Color`) of *TECHNIQUE*.

Last, we used a deductive (top-down) thematic analysis (Braun and Clarke, 2012) of the notes and recordings of participants thinking aloud based on the audio and video recordings I took of the sessions, including answers to questionnaires and notes. For this part, I took charge of coding and extracting the data from my notes, which later ran through several iterations discussing with my supervisors.

3.3 RESULTS: QUANTITATIVE ANALYSIS

This section reports the quantitative analysis we performed on the event logs and questionnaires (see Section a.3 for a summary of the collected data). First we looked at the techniques found by participants. Then, to further our analysis of participants' performance, we looked at the differences in the number of types of commands involved in each technique. Next, we analyzed the relationship between the number of techniques and the participants' self-evaluated experience and creativity scores. Last, we analyzed whether experience and creativity scores predicted the use of re-purposing techniques.

3.3.1 Most Participants Re-purposed Commands

We analyzed the event logs to record the first command of the session for every participant: 11 participants began with the Tab key and 5 began with a command associated with the ruler. Note that this is not necessarily the command that they used to solve the task in their first trial.

Participants performed a mean $\#SOLUTIONS = 3.31, SD = 1.14$. All 16 participants used `Spacebar`, 14 used `Ruler`, 11 used `Clipboard`, 7 used `Color` and 5 used `Tabulator` as solutions. All the participants who performed `Color` also performed `Clipboard`, resulting in 11 participants who re-purposed at least one command. Finally, 5 participants performed only familiar techniques: 4 performed `Spacebar` and `Ruler`, and 1 performed only `Spacebar`.

We calculated the median trial number for each technique and observed that participants predominantly started with `Ruler`, with `Spacebar` as second technique. `Clipboard` or `Color` occurred always at least on the third trial. This was also the case for `Tabulator`, although it was used more as a last resort before giving up (see Section a.3 for details). Our results were in line with our expectation that participants would perform re-purposing techniques only after familiar ones, i.e. `Clipboard` and `Color` would always take place after both `Spacebar` and `Ruler`. When `Tabulator` was deemed acceptable, it was always tried after `Ruler` and `Spacebar` and before `Color`.

3.3.2 The Set of Used Commands Expanded with the Technique's Difficulty

We analyzed the extent to which participants explored the set of available commands before performing a technique that they consider successful. We logged the commands that participants used during each trial and classified them by type, e.g., insert space, delete characters. We measured *#TYPES* as the number of different command types used before completing a trial, where high values for a given trial indicate using or exploring a large number of different command types. We then analyzed *#TYPES* by *TECHNIQUE* (Figure 3.2).

We used linear mixed-effects models to analyze differences in *#TYPES* between techniques, accounting for the repeated measures of the same participant as a random effect. We used two models with `Spacebar` and `Ruler` as baselines, i.e., as intercepts, respectively, be-

Table 3.2: Results of linear mixed-effects models of #*TYPES* by *TECHNIQUE*. The top half uses the #*TYPES* of `Spacebar` trials while the bottom half uses the #*TYPES* of `Ruler` trials as intercept for each respective model. Both show significant effects of `Clipboard` and `Color` with higher #*TYPES* involved in devising these techniques.

Technique	**Diff.**	**Std. E.**	**p**	**95% CI**
Spacebar Technique as Intercept				
`Spacebar`	**4.313**	.917	.000	[2.515, 6.110]
`Ruler`	.312	1.202	.795	[-8.29, -3.58]
`Tabulator`	3.101	1.746	.076	[-.321, 6.523]
`Clipboard`	3.670	1.299	.005	[1.125, 6.216]
`Color`	4.317	1.521	.005	[1.336, 7.297]
Ruler Technique as Intercept				
`Spacebar`	-.312	1.202	.795	[-2.669, 2.044]
`Ruler`	**4.625**	.976	.000	[2.711, 6.539]
`Tabulator`	2.788	1.763	.114	[-.667, 6.244]
`Clipboard`	3.358	1.335	.012	[.741, 5.975]
`Color`	4.004	1.561	.010	[.946, 7.063]

cause `Spacebar` and `Ruler` were the most used techniques to which we wanted to compare the others. Our results show that #*TYPES* for `Clipboard` and `Color` are significantly above `Spacebar`'s ($p = .005$ in both cases) with no significant difference with `Ruler` and `Tabulator` ($p > .05$), meaning that the set of different commands that were tried is larger for `Clipboard` and `Color` compared to `Spacebar`. #*TYPES* for `Clipboard` and `Color` are also significantly above #*TYPES* for `Ruler` as baseline values ($p = .012$ and .010 respectively) with no significant difference with `Spacebar` and `Tabulator` ($p > .05$). Table 3.2 reports the model results, where *Diff* stands for the difference between #*TYPES* means. These suggest that `Clipboard` and `Color` are less familiar than `Spacebar` and `Ruler`, because they required a broader exploration of the available commands to be performed.

3.3.3 Experience and Creativity Correlate with the Number of Techniques

We then analyzed the impact of experience and creativity scores on participants' performance. We studied whether there exists a relationship between the number of techniques used by the participants (#*SOLUTIONS*) and their self-reported experience (*EXPERIENCE*) and

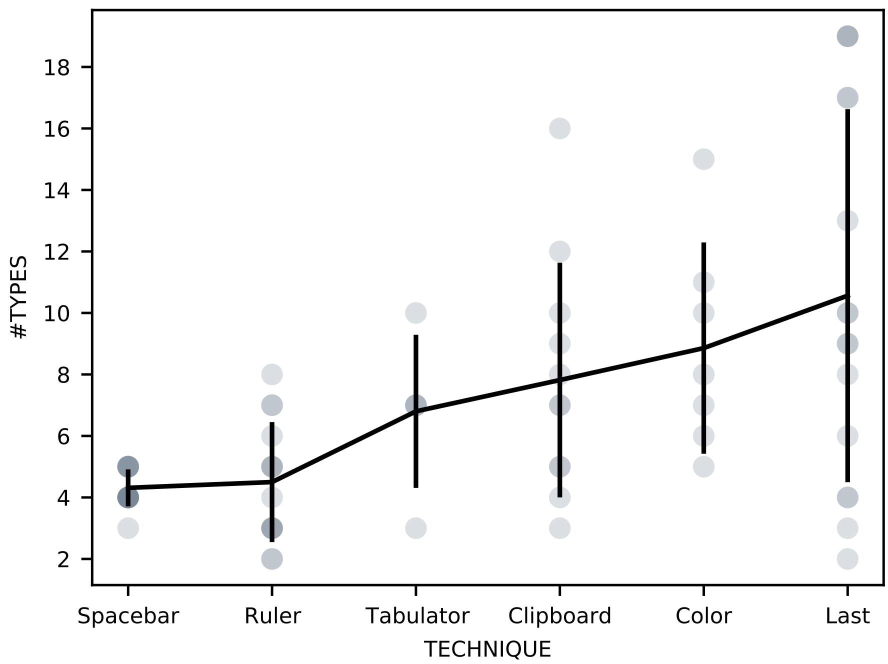


Figure 3.2: Means of *#TYPES* for each *TECHNIQUE* across participants, including the last trial with no technique. Higher values suggest that the participants used a greater set of commands in the trial where they used the technique.

creativity (*CREATIVITY*) scores. Pearson's correlation tests show that *#SOLUTIONS* is correlated with both *EXPERIENCE* ($r = .55$, $p = .03$) and *CREATIVITY* ($r = .73$, $p = .001$) (Figure 3.3).

Only participants above the mean *EXPERIENCE* ($M = 17.31$, $SD = 4.54$) performed `Tabulator`, and only those above the mean *CREATIVITY* ($M = 39.38, SD = 3.56$) performed `Color`. We did not interpret that greater experience makes participants mistake a "poor" technique (`Tabulator`), i.e., one that does not achieve the requested goal, for a "good" one (`Color`), i.e., one that achieves the goal albeit in a non-standard way. In fact, none of the participants who finished the task by performing `Tabulator` did it before their third trial, suggesting that it was a last resort before giving up or trying unconventional methods.

Both *EXPERIENCE* and *CREATIVITY* correlate positively with the number of alternative techniques, although *CREATIVITY* shows a stronger relationship (Pearson's $r = .73$ vs. $.55$). Therefore, our results show that both creativity and experience associated with finding alternative techniques to complete the task.

3.3.4 Creativity is the only Significant Predictor of Re-purposing

Next, we focused the analysis on the association between both creativity and experience and the two re-purposing techniques, `Clipboard` and `Color`. We used binary logistic regressions to model the likelihood

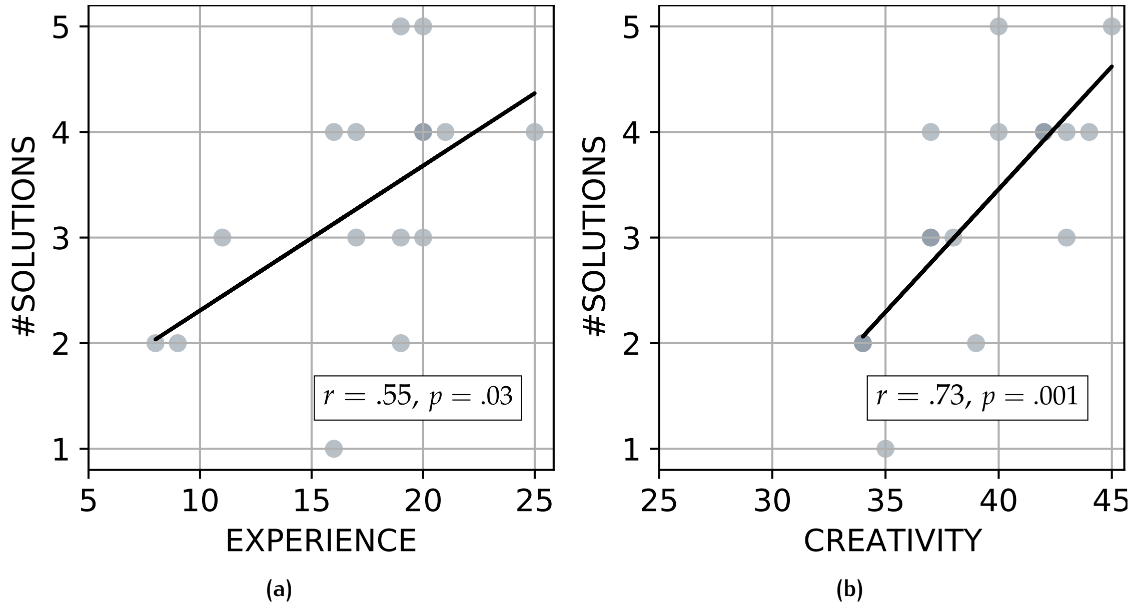
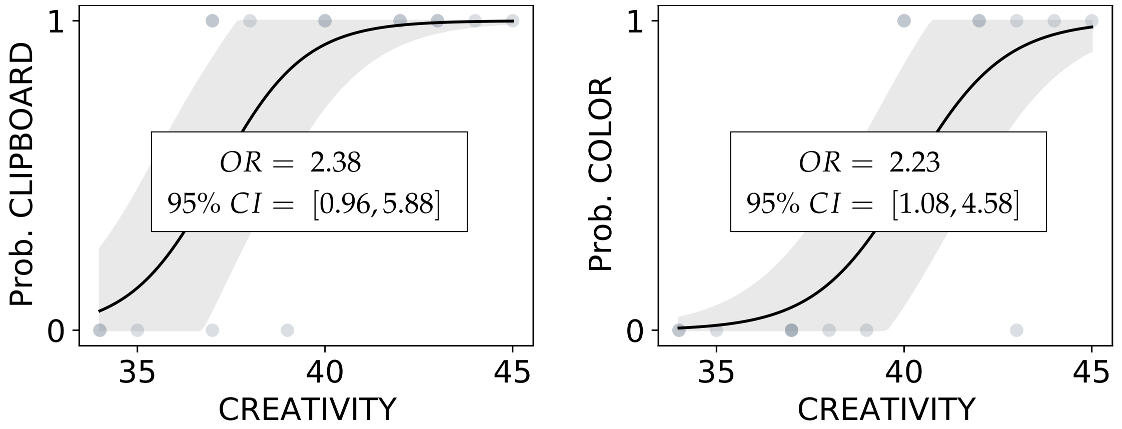


Figure 3.3: Scatter plots of *#SOLUTIONS* vs.*EXPERIENCE* and *CREATIVITY* scores showing positive correlations.

of performing `Clipboard` and `Color` as functions of *EXPERIENCE* and *CREATIVITY* separately, i.e., the predictor variables. We used the McFadden's pseudo R^2 as a measure of the fit of the model, with values above 0.20 representing a good fit (Domencich and McFadden, 1975). We reported the odds-ratios (*OR*) and its 95% confidence interval to indicate the rate of change in the odds with every change by a unit in the predictor. As an example, a model predicting re-purposing with $OR > 1$ for *CREATIVITY* would mean that the likelihood of re-purposing increases by *OR* with every 1 unit increase of *CREATIVITY*.

We constructed four models detailed in Table 3.3. We found that *CREATIVITY* is a significant predictor of the likelihood of performing `Color` to solve the task ($p = .03$, $OR = 2.23$, 95% CI = [1.08, 4.58]). Pseudo $R^2 = .52$ indicates a good fit of the model. We also found a "borderline" effect for *CREATIVITY* as a predictor of `Clipboard` ($p = .06$, $OR = 2.38$, 95% CI = [.96, 5.88], Pseudo $R^2 = .52$). On the other hand, models using *EXPERIENCE* as a predictor do not produce any significant or borderline effect for either technique. We plotted the predictions of `Clipboard` and `Color` as functions of *CREATIVITY* from our models, shown in Figure 3.4a and Figure 3.4b respectively. Our results show that *CREATIVITY* was a significant predictor of performing `Color`.



(a) (b)

Figure 3.4: Probability of `Clipboard` (a) and `Color` (b) as functions of *CREATIVITY*. Both plots include a visualization of the 95% confidence interval and an overlay of the data points used to train the models. (a) shows a curve with a borderline effect with $p = .06$ while (b) presents a logit curve with good fit and a significant result.

3.4 RESULTS: QUALITATIVE ANALYSIS

This section presents the results of our thematic analysis of the notes and recordings of the study. I analyzed the verbal protocols focusing on two categories: traces of procedural knowledge and traces of technical reasoning. I used codes for actions, functional fixedness, reflections about approaches, knowledge about text editing environments, transfer from past experience, and how participants found the commands that they needed. We cross-analyzed the codification to review the quotes associated with each code.

First, we presented the findings related to potential biases due to experience and procedural knowledge. Next, we presented findings about cues of text-editing knowledge used to re-purpose commands that we attribute to mechanical knowledge in technical reasoning.

Table 3.3: Results of logistic regressions modeling the likelihood of performing `Clipboard` and `Color` as functions of *EXPERIENCE* and *CREATIVITY* separately. Rows indicate the predictor variable, p-value, odds-ratio, its 95% confidence interval, and the model's pseudo R^2.

	TECHNIQUE							
	`Clipboard`				**`Color`**			
Score	**p**	**OR**	**95% CI**	R^2	**p**	**OR**	**95% CI**	R^2
EXPERIENCE	.09	1.29	[.96, 1.74]	.19	.33	1.14	[.88, 1.47]	.05
CREATIVITY	.06	2.38	[.96, 5.88]	.52	**.03**	**2.23**	**[1.08, 4.58]**	**.52**

Overall, we found that all participants elicited knowledge of text editing and text-based commands and most participants elicited reasoning towards re-purposing at least one command to complete the task.

3.4.1 Participants Elicited Procedural Knowledge of Familiar Techniques

All participants demonstrated pre-existing knowledge associated with the task in their first trial, manifested in the first-ever command that they used in the environment. Even if many actions would be backed by declarative knowledge due to the think-aloud protocol, we expected that trivial actions such as selecting text with the mouse or typing would be carried out backed by procedural knowledge. We found that this knowledge sometimes blocked the participants or caused them to approach the problem in ineffective ways. Furthermore, some participants who managed to re-purpose one or more commands were critical of the unorthodox nature of these approaches. We analyzed how the participants' past experience influenced exploration and decision-making.

3.4.1.1 First Actions were Based on Common Practice

All participants elicited knowledge based on usual practice in their first approach. For example, **P5** had the ruler disabled in her second trial and decided to use the Tab key: "*the ruler wasn't there so I went straight for the tabulator; I didn't even look [at the interface].*" **P10** illustrates a similar case coming from software IDEs. He selected a sentence and pressed the Tab key, only to see the text being replaced by a tabulator character, about which he said: "*I realize I don't know how to use the ruler on top; I try to [manipulate text] as if it was code.*"[3]

From our quantitative analysis, all participants elicited a first approach using either `Tabulator` (11/16) or `Ruler` (5/16). Contrary to our expectation, none of the participants attempted to perform `Spacebar` as their first approach. We argue that these attempts to perform `Ruler` and `Tabulator` first have their roots in priming by recognition of the environment, leading to an action associated with the task. Furthermore, our analysis suggests that participants followed a procedure, e.g., "to indent, press the Tab key," overlooking other alternatives. This is consistent with the notion of procedural knowledge (Anderson, 1976) associated with the indentation task.

3 Although this is not standard behavior in text fields, professional word processors interpret the Tab key pressed after a text selection as an indentation command, identical to how IDEs do it.

3.4.1.2 ***Personal Experience Induced Biases and Blocks***

All participants justified ineffective approaches based on usual practice. For example, **P3, 5, 7, 15 and 16** finished a trial with `Tabulator` on the basis of it being their usual approach in text editors. Furthermore, **P13 and 16** attempted to use the smallest font size available to make characters invisible enough while still occupying space, so as to push the sentence to the right; this did not work, but they considered it valid on a "best effort" basis: "*The result was not perfect, but I used the smallest character I could find*" (**P13**).

While not necessarily ending the session, 10 participants expressed feeling blocked after standard commands became unavailable, e.g., the Spacebar: "*I don't see other tool that would generate a blank space*" (**P1**). Similarly, **P4** felt lost at the lack of a ruler: "*[the ruler] is how I understand it is always done.*" **P4, 7, 9, 11, 12, 16 and 17** expressed frustration as they explored the interface for a replacement for either the Spacebar or the ruler: "*I'm feeling frustrated and dumb*" (**P12**); "*at this point I'd be searching on the Internet*" (**P4**).

This apparently led **P3, 5, 7, 15 and 16** —all above the median *EXPERIENCE*— to accept `Tabulator` despite its poor result. This supports the characterization of `Tabulator` as a last resort in our quantitative analysis, reflected in **P16**'s comment: "*[I am] leaning on leaving the tabulator and accept the [extra] space that is left. (...) My solution is resignation.*" However, attempting to use the tabulation character was a general occurrence, even among those who did not accept it as a solution. For example, **P13** rejected `Tabulator` yet called it "*the way to [indent].*" This suggests that the experts' tendency to accept `Tabulator` was based on usual practice, i.e., the sense that it was right because it worked in other text editors.

P4 recognized a bias from her daily practice when stating: "*I have my mind set on a design application so I try to do things that I could do with the design application.*" Similarly, **P10 and 12** realized that they had grown habituated to code editing, as they attempted to select text and press the tabulator key to indent text as in their code editing environments.

In summary, all participants demonstrated a sense of what *should* and *should not* be done in our text editor, and attempted to use familiar techniques (`Spacebar`, `Ruler` and `Tabulator`) before finding themselves thwarted and having to spend time exploring the environment.

3.4.1.3 ***All Participants Elicited Knowledge of Other Text-editing Environments***

During their exploratory phase, all participants demonstrated experience with text editing commands beyond what the experimental editor offered. For example, **P7** was able to describe the way in which she uses her own text editing application to achieve the goal: *"If I were in Word (...) I'd go to [the] Paragraph [menu] and then [to the] Indentation [section] and set the indentation I want."* **P3 and 12** extended the possibilities we had considered for the editor by attempting to insert spaces using ASCII codes.[4] Similarly, **P10** showed her understanding of the system clipboard by concluding, after she performed `Clipboard`, that the copied text would still be stored in the operating system's memory: *"I can paste... I have the space that I had copied before."* However, since she had used `Clipboard` in the previous trial, all clipboard commands were now disabled.

This is not to say that all participants mastered the commands they knew about. For example, 7 participants expressed that they were not sure about which was the appropriate slider in the ruler to control paragraph indentation. **P7** stated: *"one control moves the line [indentation] and the other moves the paragraph [margin] but I didn't know which one was which so I tested."* However, even if the participants had superficial knowledge of some commands, they had a sense of whether they were pertinent to the task or not. For example, as **P1** hovered with the cursor over the toolbar, she voiced a mental checklist: *"...font face won't do anything for me, the size neither..."* This shows that participants made associations between the task and commands, regardless of whether they were available in our experimental editor, as if trying to find the procedures that would work for the task.

3.4.1.4 *Some Participants Judged Techniques Based on Knowledge of Good Practice*

Some participants who managed to overcome their blocks criticized the re-purposing techniques and, to our surprise, considered `Spacebar` to be a bad approach. For example, **P1**, on her way to performing `Spacebar` to complete the task, changed her mind and switched to find if `Tabulator` was possible instead, stating that the latter *"is more correct."* **P14** said of `Spacebar`: *"it is introducing new characters and it's not formatting,"* and when he realized that there was no other option, added: *"I'm not satisfied by that but it does the trick."* **P18** compared `Spacebar` to *"stacking books to prop a monitor up."*

About `Clipboard`, **P12** felt it was against *"the rules"* of text editing, stating that in prior trials he *"was not trying to cheat but find a* reasonable *way"* to complete the task. **P18** characterized it as *"patching things"* and established a parallel between `Color` and *"using Corel DRAW or Illustrator the wrong way."*

Nevertheless, other participants expressed satisfaction after performing `Clipboard` and `Color`. This is best exemplified with **P3** who, after performing `Clipboard`, recognized her action as an unorthodox use of the color command when she said: *"I feel like I am MacGyver"*.

4 In Windows, using a keyboard's Number Pad with Number Lock on, it is possible to insert a character by typing its ASCII code with the ALT key pressed.

In sum, participants were reluctant to re-purpose commands even when they saw their effectiveness, justified by what they deemed "*good practice*". This supports the use of procedural knowledge, because even though participants managed to break free from their block, they showed a strong reliance on a procedure-based approach.

3.4.2 Participants Elicited Technical Reasoning

All participants demonstrated a basic knowledge about the commands and interactions that apply to a text-editing environment, i.e., text properties and the mechanics of text editing. Additionally, some participants made associations between re-purposing approaches and past experience using techniques from digital environments besides text editing, showing their ability to transfer knowledge. Some participants demonstrated knowledge of text-based commands while evaluating those that could lead to a solution vs. those that would not, consistent with the use of technical reasoning. Finally, participants tried commands without expecting successful results, an approach that we characterized as "poking" at the interface for inspiration.

3.4.2.1 All Participants Elicited Knowledge of Text Mechanics

All participants described how the commands' effects would or would not produce a desired result. For example, **P1** was concise in her approach: "*All I know is that I have to put a character in front [of the sentence],*" explaining how to push existing characters to the right side of the page. **P12** explained why an initial idea he had would not work: "*I tried cutting the text and putting the cursor at the guideline, but without anything written, that's impossible.*"

This was also reflected in steps that left participants close to a re-purposing approach without necessarily realizing it on the first try. For example, **P2** reflected: "*I can insert characters [in front] but if I delete them [the remainder] will move,*" thus understanding that characters in front of the sentence push it towards the right but missing the fact that `Color` could make them invisible. **P10** thought of another approach: "*I'm going to check if there's a font size 0 but no. It would be complicated to align anyway*" showing an understanding that a text size 0 would imply zero-length characters.

Our observations suggest that all participants expressed some form of knowledge about the mechanics of digital text, i.e., the principles governing digital text input, which we interpreted as a form of mechanical knowledge about digital text.

3.4.2.2 Some Participants Elicited Knowledge of Text Properties

7 participants (**P1, 3, 9, 10, 12, 17 and 18**) described their need for an object to act as a blank space, referring to it in various ways. **P3**

characterized it as "*a* letter *that is a space.*" This led her to search for "*an emoji that doesn't work and therefore looks like a blank space,*" which she found. This was an original solution that we had not accounted for. Similarly, **P10** said: "*there are* symbols *that are not drawn with some fonts, [such as] accents,*" although he could not reproduce it in the experiment. Lastly, some participants were less technical in their descriptions of the object properties they sought. For example, **P18** said: "*I don't know what to call it [...] it's an* empty *character.*"

We analyzed these descriptions as expressing object knowledge, i.e., object properties that are needed to complete the task. By the end of their session, 6 of these 7 participants performed `Clipboard` and 3 performed `Color`. This suggests that they had declarative knowledge about text-based properties, i.e., knowledge of properties of text objects.

3.4.2.3 Some Participants Transferred from Past Experience

7 participants expressed associations with their practice using other digital environments after performing `Clipboard` and `Color`, suggesting that they transferred knowledge from other applications. **P1, 10 and 18** thought of `Color` in relation to a graphical editing trick they perform where they overlap shapes with the same fill color as the background to mask parts of the content underneath them. **P1** said: "*[In Photoshop,] I put white squares on top of everything.*"

Closer to text editing, **P13** saw in `Color` his own use of LaTeX's `\phantom` macro, which draws a blank space the length of the characters passed as argument. **P12** took some time to realize that he could perform `Clipboard`, after which he reacted saying: "*I can't believe I didn't think of this before! I normally do that for the 'ñ'; I [search for it on] Google and copy it from there.*" This was identical for **P10**, who explained how he performs `Clipboard` frequently to insert characters that cannot be typed with the keyboard. Additionally, **P10** crystallized his experience as a web developer when he attempted to write HTML character entities[5] instead of searching for conventional keyboard-based techniques.

Arguably, these participants re-purposed text-editing techniques through analogies, i.e., recognizing surface aspects of the task that matched past experience. We analyzed this approach as the *transfer* of knowledge to a new task, similar to how technical reasoning relies on the transfer of mechanical knowledge.

5 HTML Entities are markup to print reserved HTML characters in a document, e.g., the `&space;` entity renders a blank space character in the browser.

3.4.2.4 *All Participants Consciously & Aimlessly Tried Commands in the Environment*

All participants performed one or more actions about which they were quite certain that it would not produce a result towards the solution. **P9** stated it explicitly: *"I'm gonna randomly press the* Paste *button (...) I'm out of ideas."*

Despite generally good knowledge of the formatting commands, **P2, 11, 13, 14 and 18** did not know the purpose of the *Clear Format* button and decided to test it while exploring for ideas. **P3, 4, 7, 10 and 13** tested key combinations of the Alt/Option or Control key with multiple characters, hoping that they could hit a shortcut that they did not know to indent a line or insert a space. **P12** went even further: *"I know this would not work in any editor, but maybe in this one [...] if you put underscores and then you underline them, maybe you [will] cancel them."* This demonstrates technical reasoning, combining the mechanical knowledge about characters having an "underlined" attribute that the "Underline" command can unset, and inaccurate knowledge about the underscore character having the underlined attribute set by default.

The fact that some participants tried random actions when they were out of ideas is probably due to the design of the experiment, which could have given them the impression that there was yet another solution. However, it also revealed their knowledge of text environments as some of these actions had a certain logic to them, including the fact that text editors have a lot of hidden commands and features, and the knowledge that it was possible to recover from errors with the "undo" command.

3.5 DISCUSSION

Our findings suggest that most participants engaged in technical reasoning to re-purpose a command in our text-editing task. The notion of "good practice" expressed by some of the participants suggests that functional fixedness was a factor in blocking or limiting uses to the culturally-assigned functions of commands. This section is closed with some implications of this work for HCI.

3.5.1 Evidence of Technical Reasoning & Functional Fixedness in Digital Tool Use

Overall, participants demonstrated text editing knowledge characterized by *principled* expectations consistent with an understanding of text mechanics. The inability of some participants to describe these principles in words despite being able to apply them, does not contradict the notion of mechanical knowledge for a digital interaction because it is defined as based on abstractions of causalities, in the same way that we understand gravity without necessarily being able to explain it.

Among the majority of participants who re-purposed at least one command, their expression of associations with past practice suggests that they found analogies with other digital environments on which to ground their approach (Rieman et al., 1994). Such transfer is also a sign of the participants exerting technical reasoning based on their mechanical knowledge of another environment. The association of command re-purposing cases with high creativity scores is compatible with the creative aspect behind using familiar tools in novel ways observed in creative problem-solving with physical tools (Coughlan and P. Johnson, 2009). Additionally, our observation of participants using commands without clear purposes resonates with fidgeting and fiddling behaviors involving physical objects around the work space, associated with creative processes (Karlesky and Isbister, 2016).

The lack of significant association between re-purposing and experience seems consistent with the perspective by Carroll and Rosson (1987) stating that users often focus on completing tasks rather than on exploring the interface for alternative strategies, namely, a "production" bias. Additionally, users frequently approach new tasks based on interpretations of old ones, known as an "assimilation" bias.

For the minority of participants who did not manage to re-purpose commands, our observations of bias and blocks show an effect akin to functional fixedness (Duncker and Lees, 1945). This is further supported by their justification of poor results based on usual practice, which resonates with the notion of mental set (Wiley, 1998) in problem-solving, i.e., participants stuck using a learned pattern to complete a task when it is not possible to use it. It also echoes the discussion by Cockburn et al. (2014) about "satisficing," a phenomenon evidenced notably in users learning a minimal subset of functions adapted to their needs and rarely exploring the interface for more efficient alternatives.

In sum, all participants elicited knowledge of text editing tools compatible with mechanical knowledge of physical tools. Finally, while some participants experienced functional fixedness about the use of text editing commands, most of them elicited a reasoning process towards re-purposing these commands that is compatible with the Technical Reasoning hypothesis.

3.5.2 Implications for HCI

Our findings can extend existing interaction models such as Instrumental Interaction (Beaudouin-Lafon, 2000), to account for Technical Reasoning. Instead of focusing on the multiplicity of domain objects with which an instrument (or tool) interacts, we could design them to operate on the *properties* of these objects instead, by taking advantage of users' ability to grasp technical principles from observing the effects of tools on these properties, and their ability to perform technical reasoning. For example, instead of defining the objects with which a color picker can interact, we would rather model it as a tool that interacts with the `color` property of objects. Thus, any domain object with such a property, e.g., a shape, cell or text selection, would react to the color picker being used on it, and more generally to any tool that uses this property.

As a short case study inspired by our experiment, standard word processors deal with many different sizes: text size, line spacing, image size, margin size, etc. However, text size and line height are usually controlled by number input fields, e.g., in the toolbar, while images support resizing by direct manipulation, and margins require the use of a dedicated ruler. Based on the reification and polymorphism principles of Instrumental Interaction (Beaudouin-Lafon and Mackay, 2000), there is an opportunity for redesign by creating a new tool whose mechanical principle is to alter the size of any object with a size-like property. This resize tool could be used to resize text in a selection, by dragging its corners as is done for images; line spacing, by placing the cursor between lines and dragging up and down; images, by keeping the current direct manipulation of handles; and margins, by dragging the sides of paragraphs or the page.

Technical reasoning offers a model based on reasoning to ground the design of interfaces for appropriation (Dix, 2007) and creative use (Coughlan and P. Johnson, 2009). It capitalizes on 'real-world' cognitive abilities (Jacob et al., 2008) that underlie our understanding of interactions among objects based on knowledge of their properties and the principles that govern them. Additionally, technical reasoning complements existing theoretical work in HCI grounded in ecological psychology and Activity theory, such as technology affordances (Gaver, 1991) and mediated action (Kaptelinin and B. Nardi, 2012). As such, the Technical Reasoning hypothesis brings new general concepts to HCI theory that can inform old and new practices in interaction design (Rogers, 2004). Ultimately, we believe that a reasoning-based approach to designing interactions offers a promising path to leverage instrumental genesis processes (Béguin and Rabardel, 2000) in digital environments, enabling the adoption and appropriation of digital tools and overcoming the limitations imposed by current software.

3.6 SUMMARY

The Technical Reasoning hypothesis is a theoretical model of human tool use based on reasoning about mechanical principles and physical object properties that explains how tools can be used in unusual ways to achieve specific goals, i.e., tool re-purposing. In this work, we designed an experimental environment and conducted a study forcing participants to re-purpose digital tools and analyze their thought process, assessing its compatibility with the technical reasoning model. We showed that most participants managed to re-purpose at least one digital tool in order to complete a text layout task. We also found that participants with higher self-reported creativity scores were more likely to use one of the re-purposing techniques, compatible with creative problem-solving situations with physical objects. We interpreted these results as a sign of exerting technical reasoning over digital tools and objects, rather than applying procedural knowledge about text editing situations. Our analysis of the verbal protocols showed that participants elicited a transfer of knowledge about text objects and text editing mechanics compatible with mechanical knowledge about physical objects.

Arguably, our results support designing interactive systems that leverage the users' ability to perform technical reasoning to re-purpose and use digital tools in their own ways. Such an environment should convey the "nature" of its objects to users, so as to prime the appropriate knowledge to interact with them. In order to study possible cues to inform users, the next chapter focuses on the effect of the environment presentation to prime "mechanical" knowledge about digital interactions.

4 EXPLORING CUES FOR MECHANICAL KNOWLEDGE

In Chapter 3, I presented evidence that computer users perform technical reasoning in order to carry out unusual uses of digital tools. The Technical Reasoning hypothesis posits that humans possess *mechanical knowledge* that encompasses abstract knowledge of physical properties of objects and mechanical principles about their interactions (Osiurak et al., 2010). Technical reasoning and the notion of mechanical knowledge challenge the notion of *direct perception* of function posited by the Theory of Affordances (J. J. Gibson, 1986), instead offering a cognitive model that interprets object interactions based on knowledge accumulated from past experience. So, what would it mean to possess "mechanical" knowledge about digital tool use?

One way to understand users' knowledge of digital environments is through the rules that they incorporate as they gain experience, namely, the *principles* of the digital world. Text editing makes for a compelling example because the principles that govern it (e.g., what happens when you insert a character, delete it, select it, etc.) are almost perfectly consistent across applications, operating systems and platforms, i.e., users know what to expect out of their actions. Thus *digital principles* can be seen as the "technical laws" of digital environments. Our study in Chapter 3 suggests that users transfer digital principles between tasks, re-intepreting the use of digital tools for purposes beyond their design, in the same way that physical tools are used beyond their culturally and/or technologically assigned function (Kaptelinin and B. Nardi, 2012).

Previous research in HCI has addressed the transfer of knowledge to convey functions and help users learn how to interact in new digital environments. Initially, it mainly focused on transitioning between text-based word processing environments (Douglas and Moran, 1983) and later expanded to GUI development, reflected in visual design concepts, notably, the notion of *visible* affordances brought to HCI by Norman (1988)—later reformulated as *signifiers* (Norman, 2008). These are concerned with letting the user know what is possible and how to make the function work, e.g., which item under a menu executes the command. However, when it comes to physical tools, technical reasoning is based on the abstract properties of objects to, for example, know when a learned mechanical principle applies to the relationship between a tool and an object, e.g., cutting bread the way you saw a tree.

In this chapter, I present my work on learned principles about digital interactions. In particular, I was interested in how environmental cues of an interface convey *principles* rather than *functions*, so as to induce a transfer of knowledge that we can relate to the "mechanical" knowledge of the digital world. I developed an experimental editor whose contents support both text- and graphic-based interactions, i.e., they support either text- or graphic-based commands. We designed a protocol around this environment where participants are presented with either text- or graphic-oriented toolbars, and analyzed their strategies towards the objects for completing a series of tasks. We investigated the results framed by mechanical knowledge and the Technical Reasoning hypothesis.

4.1 RELATED WORK

This literature review looks at fundamental theory about how humans interpret and learn about the possibilities for action with physical and digital objects, i.e., the user's mental model (Johnson-Laird, 1989). It begins with the Theory of Affordances introduced by J. J. Gibson (1986) to refer to the perception of physical objects. Next, it focuses on how users transfer past experience to make sense of new interfaces, and the effects of such reliance on knowledge of other digital environments. The goal is to provide HCI-relevant background about how users perceive what is possible with an interface.

4.1.1 Affordance Perception

The Theory of Affordances (J. J. Gibson, 1986) models the use of tools as the product of *direct perception* of the objects' salient features, allowing an animal to infer what a tool affords relative to its body capabilities, e.g., "*an elongated object of moderate size and weight affords wielding*" (J. J. Gibson, 1986). J. J. Gibson goes on to conceptualize tools as being *detached* objects that afford, in particular, grasping and carrying (J. J. Gibson, 1986). Additionally, McGrenere and Ho (2000) note that the direct perception of features rejects any interpretation based on experience or cultural background, almost as an objective capturing of information, except that it is subjected to the animal's body. Kaptelinin and B. Nardi (2012) argue that while direct perception may very well explain the mechanism behind the manipulation of some primitive physical objects, it falls short of explaining how humans operate contemporary technology. The authors illustrate this with how an individual can operate a power drill which displays no visible association between the trigger and the drill part, thus having no salient feature that could be directly perceived as connecting the two (Kaptelinin and B. Nardi, 2012). Gaver (1991) provides a hierarchical approach that would seem to address this dissonance through *sequential affordances*. In Kaptelinin and B. Nardi's example, this would amount to noticing the "push" affordance of the trigger, leading to see the drill turn thus inferring the affordance of the power drill as a whole. But what about other buttons that could adjust settings of the power drill or even more so, buttons in a control panel?

Arguably, we share cultural conventions, such as those that tell us that a green button makes a machine go or that a red button in a yellow box next to it produces an emergency stop. Kaptelinin and B. Nardi (2012) and McGrenere and Ho (2000) share the idea to extend affordances to consider the cultural influence affecting the use of technology. After all, users could arrive at the conclusion that the trigger activates the drill part in a power drill just from having observed other power tools, rather than requiring the perception of their association every time. In digital environments, GUIs make extensive use of cultural conventions such as adding shades and depth effects to buttons and scroll bars to indicate what is click-able, scroll-able, etc. (McGrenere and Ho, 2000) without them being specific of a particular environment or task. More recently, Norman (2013) used the term *signifier* to refer to the perceivable visible properties of objects that tell us how to operate them. However, the prevailing paradigm in desktop and mobile interfaces still recurs to WIMP-based elements (Van Dam, 1997), in particular, keeping functions and options organized inside menus which make them invisible and do not reveal what the interface affords until they are used on its objects. Therefore, users must recur to methods other than the perception of affordances—or signifiers—in order to discover unfamiliar functions and solve new problems in digital environments.

4.1.2 Analogical Reasoning

Humans often address problem-solving through analogies of past experiences to interpret current situations (Gick and Holyoak, 1980), making use of knowledge of old solutions for new problems. Consequently, analogy is pervasive as an approach to designing interfaces that can be easily adopted by novice users (Carroll et al., 1988). For example, the desktop metaphor in the Xerox Star (J. Johnson et al., 1989) relied on an analogy of the physical office to convey the possibilities for interaction offered by its files, folders and bin icons to infer that files can be *put* into folders.

However, analogies can also mistake users into thinking that certain actions are possible when they are not, configuring a *negative transfer* of knowledge. In this regard, Carroll et al. (1988) talk about the "mismatches" when it comes to using metaphors as a way to teach use. Similarly, Douglas and Moran (1983) found negative transfer taking place in participants of a study when using an analogy of typewriting to teach them how to use a digital word processor. Other studies have looked at the use of analogies based on other word processors as a way to transition *between* digital tools (Karat et al., 1986; Polson et al., 1986; Ross and Moran, 1983), showing both positive and negative transfer taking place. More generally, Rieman et al. (1994) offer an analysis based on comparing two cognitive processing frameworks, advocating the need for consistency between interfaces in order for analogies to be effective.

Cockburn et al. (2014) observe that users are frequently subject to "satisficing" (Simon, 1956), i.e., a tendency to stick to strategies learned as a novice, despite their suboptimal performance, hindering the learner's path towards expert performance. Arguably, this would serve as advice against teaching how to use a new, more powerful digital environment based on concepts of a different one. For example, Tetzlaff (1986) shows the results of an experiment where participants are taught a text editor which has built-in commands for two styles of text editing, namely, *line*[1] and *screen*. One group of participants is taught the commands of each style in sequential order, while the other is taught in a disjoint way, thus learning to accomplish tasks combining commands from both styles. The author observes that participants taught in a disjoint way have more difficulty distinguishing between the two styles built into the application, which translates in difficulty to transfer the appropriate knowledge to complete the task. Therefore, while analogies of other systems may help initiate users to a new (more powerful) system, it may also hinder them from reaching an expert user's level.

4.1.3 Mechanical Knowledge

The Technical Reasoning hypothesis describes an analogical reasoning process occurring from matching the similarity between two interactions among objects, e.g., slicing bread with a serrated knife with sawing a tree, which results in the transfer of mechanical knowledge to use the appropriate action. The results presented in Chapter 3 constitute a lead towards the Technical Reasoning hypothesis as a framework for digital interactions, observing that users elicit knowledge of digital objects and principles of digital interactions based on past experience in digital environments (Renom et al., 2022). Therefore, a "mechanical" knowledge of the digital world could model the way in which analogies of digital tool uses are made.

In the Technical Reasoning hypothesis (Osiurak et al., 2009), mechanical knowledge is acquired through experience with physical objects, both implicitly and explicitly learning conventions around the use of particular objects. For example, we probably learn how to cut with knives before we can discover it by ourselves, therefore learning

1 See `https://en.wikipedia.org/wiki/Line_editor` for a description of Line editors.

the cutting principle explicitly as part of a cultural practice. As such, this knowledge is extended to other knives, whereby we acquire the abstract knowledge of their properties to recognize "knives" in other objects regardless of their cultural function as such.

As a reminder, mechanical knowledge is not concerned with the direct perception of affordances of objects, i.e., what they afford the user, but rather with resolving their interaction with other objects. For instance, a serrated knife slices bread because of its teethed blade ripping across the crust and crumb with each movement. Analogously, a digital tool interacts with an object as long as that interaction is supported, which can be analyzed as an *actuator* that works on objects with certain properties, similar to the Instrumental Interaction model (Beaudouin-Lafon, 2000) (see Chapter 2). A "mechanical" knowledge of the digital world could offer an additional tool to analyze and design instrument-mediated interactions, accounting for the users' past experience with a given instrument and its effects on digital objects, e.g., how users assume that the mouse cursor can select any text. However, although we have examples of instrumental interfaces (Beaudouin-Lafon and Lassen, 2000; Ciolfi Felice et al., 2016), to the extent of our knowledge, no studies have focused on how users make sense of the effects of a digital instrument on a given object. In the following section, we present the design of an experiment using a digital environment that mixes both text- and graphic-based commands, in order to study users translating environmental cues into possibilities for action on digital objects.

4.2 STUDY: TOOL BEHAVIOR UPON AMBIGUOUS OBJECTS

We wanted to observe users' strategies editing content in a digital environment where the objects support both text- and graphic-based commands. I developed an experimental editor that supports a subset of common text- and graphic-oriented commands. In its initial state, the editor displays a canvas containing words and emojis arranged in floating positions (see Figure 4.1). We asked participants to recognize the environment, describe how they would interact with it and use it to recreate compositions based on the initial content. We expected participants to interact with the environment according to familiar visual and interactive cues. We first focused on the visual cues, controlling the toolbar displayed to observe whether their description of a selection technique associates with it. Second, we focused on the environmental cues, meaning, the combination of the visual cues and an interactive cue to "confirm" the environment, such as performing a selection and visualizing the system's feedback according to the participant's expectations. In order to verify the effects of priming

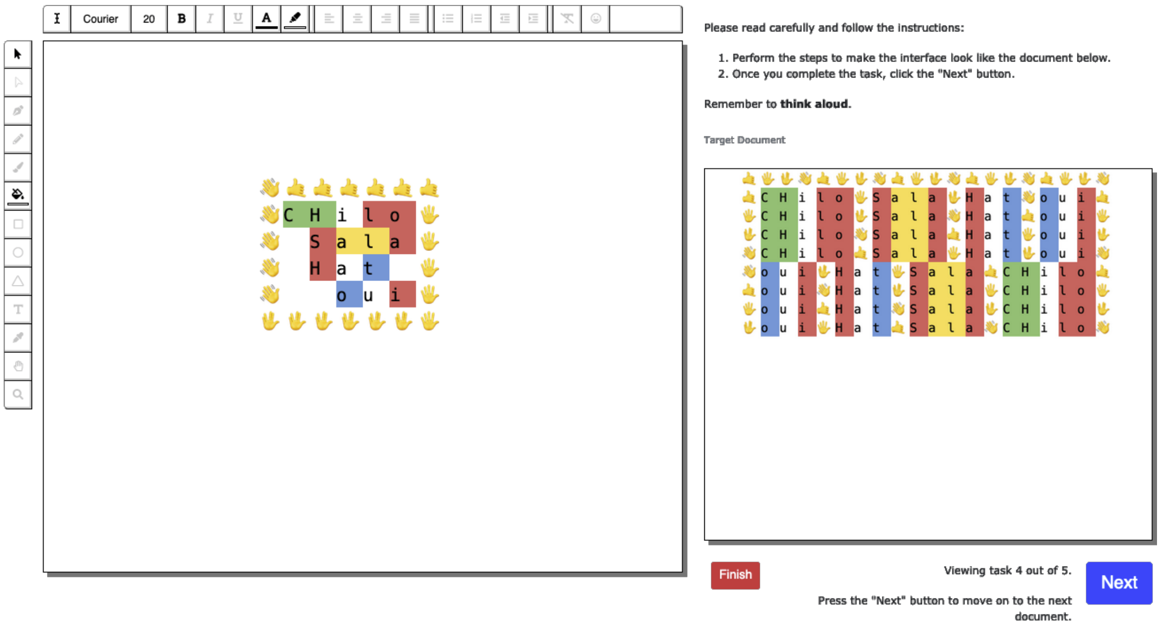


Figure 4.1: The full view of the experimental environment during Task 4. To reproduce the image on the right in optimal time, the participant required combining text and graphic commands.

knowledge with environmental cues, we asked participants to perform a series of tasks using the editor, so as to measure their choice of commands and their overall approach. We expressed our expectations in the following hypotheses:

H1: *Visual cues* prime the knowledge to select objects in the environment, i.e., a toolbar with text-based tools primes selection of objects as text, while a toolbar with graphic-based tools primes selection of objects as vector shapes; and,

H2: *Interaction cues* prime the knowledge to edit objects in the environment, i.e., a text-based interaction primes the use of text-based commands, while a graphic-based interaction primes the use of graphic-based commands for editing.

4.2.1 Task

Participants interacted with a content editor that displays monospaced text characters organized in words and 4 emoji elements in a canvas, spread out so as to avoid inducing the idea of a text document (Figure 4.1). This content editor supports interacting with elements both as if they were text and graphics. For example, a user can select the Text Highlighter tool on the top toolbar, which will switch the mouse cursor to the i-beam, letting her select the objects as a sequence of characters on a text document. Similarly, selecting the Pointer Tool will switch the mouse cursor to an arrow, letting her drag and drop objects as floating shapes on a canvas.

We used a between-participant design with one factor controlling the type of toolbar visualized according to the participant's group (Toolbar). Participants in *Text* saw a *Text Toolbar*, those in *Graphic* saw a *Graphic Toolbar* and those in *Control* saw only the canvas with no toolbars. Groups *Text* and *Graphic* were asked to perform a selection of the objects in the canvas. For example, they could highlight text if they were in the *Text* group or perform a rectangular selection if they were in the *Graphic* group. Finally, all participants interacted with a version of the editor that has both *Text Toolbar* and *Graphic Toolbar* with some of its buttons enabled, in order to perform 5 tasks corresponding to incremental steps towards a goal state. These tasks could be carried out using either only graphic commands, only text commands or a mix of both types, so as to evaluate any priming effect of the environment.

To design the 5 tasks, I ran 11 pilot testing sessions with participants both from inside and outside our lab. I tested different object representations and layouts for the canvas' content, aiming at inducing ambiguous interpretations about the appropriate interaction, i.e., text-based, graphic-based or other. We decided to use a mix of text characters arranged as words and emojis, scattered in the canvas so as to look like a vector graphics document yet made only of text elements. This is based on the premise that participants are familiar with emojis inserted as part of a text. The choice of commands for the toolbars was made weighing between those that are recognizable from popular software, e.g., the pointer for moving graphic objects and the i-beam for typing, and those meant for text and graphics that achieve similar visual results, e.g., the paint bucket and the text highlighter both change the background color property.

The first three tasks (Figure 4.2a, Figure 4.2b and Figure 4.2c) constitute small steps to familiarize the participant with the environment, such as finding out which tools and shortcuts can be used, and were designed to be straightforward. **Task 1** requires turning a spread-out composition into a new layout, re-positioning elements while keeping characters together (selecting multiple elements at once) and putting emojis right next to them with a particular alignment. **Task 2** involves re-positioning existing character elements to form new words with a particular alignment and duplicating the existing 4 emojis to design a frame around the new word alignment. **Task 3** adds background/highlight color to some of the characters in the same layout as Task 2.

The last two tasks (Figure 4.2d and Figure 4.2e), on the contrary, involve more effort and induce the need to devise strategies and find commands to make them less cumbersome. **Task 4** takes the colored words resulting from Task 3 to create a much more complex composition made of different, repetitive patterns, and requires an efficient use of the canvas' space. The goal should induce the need to perform efficient vertical selections that are only possible with the pointer tool in a graphic-based approach, while the text-based approach should allow for easy reproduction of horizontal patterns. **Task 5** requires vertically centering the composition from Task 4 and changing the background/highlight color of each line to create a striped composition of the same colors as those used in Task 3. I designed this task so that participants who use the text highlighter tool perceive it as straightforward, while participants using the fill tool are limited to a repetitive pattern of selecting color and clicking to change each individual element's background color.

4.2.2 Participants

I recruited 37 adult computer users via calls for participation over email and social networks, and word of mouth from participants on a "first-come, first-serve" basis until completing 12 participants for each of the 3 groups. Candidates were selected on the basis of self-reporting themselves as knowledgeable about computers. I discarded data from 1 participant because of inconsistencies due to a software error. Of the remaining 36, 17 self-reported as female and 19 as male. Self-reported years of experience among participants were between 11 and 20 for

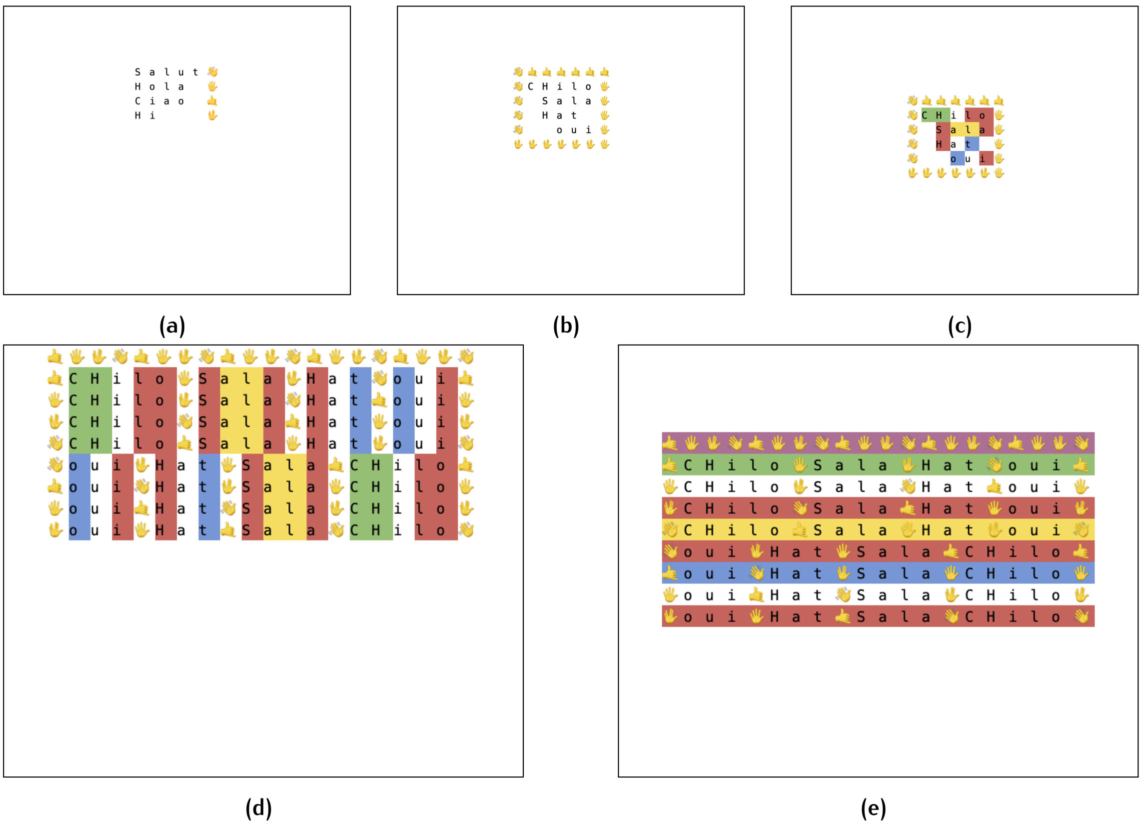


Figure 4.2: Screenshots of the tasks during a session.

text editing and between 5 and 10 for graphic editing. The median self-reported frequency of use was "Almost daily use" for text editing and "A few times a month" for graphic editing.

4.2.3 Setup

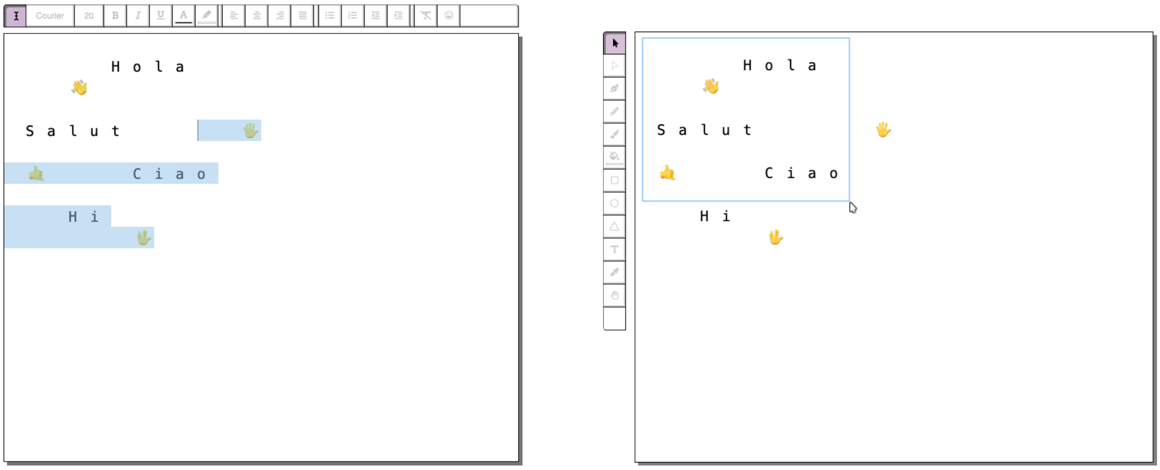
I implemented the setup to carry out the study remotely. Participants ran a local copy of the experimental environment on a web browser with support for JavaScript. The editor supports a subset of text- and graphic-editing commands. The interface layout borrows from popular text- and graphic-based editing environments, with most of its toolbar buttons in a disabled, grayed-out state to induce recognition without suggesting availability. The stack uses a Vue.js application running on the client-side, in charge of rendering the editor, collecting input events and registering answers to questionnaires in the browser's local storage. The environment generates a JSON file containing the event logs and answers to questionnaires which are stored in the participant's browser. The application scripts were hosted in a virtual server running on our lab's infrastructure.

When carrying out tasks, the interface (Figure 4.1) is comprised of a text-based toolbar at the top, a graphic-based toolbar on the left side and a rectangular canvas at the center. Only the commands that are relevant are enabled throughout the tasks (Table 4.1). The text toolbar includes a button named "Text Mode" to display the text cursor on the canvas, i.e., edit content as text, and additional buttons for modifying the font face, size and style, setting the text color, highlighting a text selection and inserting emojis. Additional buttons that are recognizable from popular text editors are displayed but disabled, e.g., the buttons for text alignment. The graphic toolbar includes a button for a pointer tool for direct manipulation of objects in the canvas and a button for a fill tool to point and click at objects to change their background color property. As for the text toolbar, additional buttons resembling those of popular vector graphic editors are provided but disabled, e.g., a node tool for modifying shape nodes.

As the mouse cursor is moved over the canvas, it changes into an i-beam when using "Text Mode," an arrow when using the pointer tool and a paint bucket when using the fill tool. Clipboard commands are made available through browser menus and standard keyboard shortcuts (cut with *Ctrl+X*, copy with *Ctrl+C* and paste with *Ctrl+V*). When either the text cursor or the pointer are active, each keep their own clipboard storage, meaning that, for example, an object copied using the pointer can only be pasted while using the pointer. The editor does not support history commands for undoing or redoing changes. This allows for a simpler implementation and logging capabilities of the environment, as well as capturing more actions from the participants' approach when they recover from mistakes.

Table 4.1: Buttons in the text and graphic toolbars required to complete the tasks.

Text Cursor		Activates a blinking text cursor at the end of the last character (top-down, left-to-right direction). If graphic selections are present at the moment of pressing the button, they are cleared. If necessary, spaces and line breaks are inserted before the elements to preserve the layout from the graphic mode.
Highlighter		Sets the background color of a text selection. If the text cursor is not present at the moment of pressing the button, it is activated. If graphic selections are present at the moment of pressing the button, they are cleared. If necessary, spaces and line breaks are inserted before the elements to preserve the layout from graphic mode.
Pointer		Activates the pointer tool to manipulate characters as shapes in a 2D space. If the text cursor is present at the moment of pressing the button, text selections are cleared and the text cursor is deactivated.
Fill		Activates the fill tool to change the background color of individual characters by point-and-click interaction. It is not possible to color multiple characters by dragging the mouse cursor across them. If the text cursor is present at the moment of pressing the button, text selections are cleared and the text cursor is deactivated.



(a) (b)

Figure 4.3: The editor at the point of selecting elements for the first time, for *Text* (a) and *Graphic* (b) groups. The button corresponding to the selection tool is the only one activated in the toolbar and the participant can only perform a selection according to the tool. All other commands remain disabled until having to complete tasks.

The canvas contains individual characters using a monospaced font for letters and representations according to the Unicode standard for emojis. All elements are padded to occupy a square slot. When dropping elements, an automatic grid adjustment ensures that all elements are automatically positioned in the nearest slot. All elements are independent from each other, meaning that characters can be selected individually from the rest of the elements with either the pointer tool or the text cursor. In order to select more than one element, the user can highlight elements as text or use the pointer tool to create rectangular selections and/or Shift+Click on each element.

The canvas is pre-loaded with initial content, in particular, with each goal state during the tasks, depicted in the image to the right of the interface (Figure 4.2), regardless of the participants' result in the previous task. After completing task N, the participants click a *Next* button outside the editor's interface that saves the action log of task N, reloads the content in the canvas according to the goal state of task N (to correct any differences in the result among participants) and displays the goal state for task N+1 in the image to the right. In this way, we make sure that every task starts with identical conditions for all participants. A *Finish* button outside the editor allows to finish the session without completing the remaining tasks.

4.2.4 Procedure

A session begins by sending the participant a unique URL corresponding to their assigned number. Using a video conferencing application with support for screen sharing, participants share a video stream of the browser window where they open the URL.

All participants encounter a canvas with content made of words and emojis, and toolbars around it according to their group. At this point, participants observe the interface and describe the purpose of the editor, i.e., what type of editing do they think the environment can do. After giving the response, participants continue describing the steps to leave all the elements in the canvas in a selected state.

Next, participants in the *Text* and *Graphic* groups see one toolbar button activated according to the toolbar type: those in *Text* see the "Text Mode" button and those in *Graphic* see the "Pointer" button activated (Figure 4.3). These participants perform the steps to leave all the elements in the canvas in a selected state according to the tools available in the toolbar. Participants in *Control* skip this stage and proceeded directly to the next one.

Last, all the participants encounter an interactive version of the editor with both its graphic and text toolbar on the top and left sides of the canvas respectively, with some of its buttons enabled but deactivated (Figure 4.1). Participants observe the interface and describe the purpose of the editor one more time. They then use the keyboard and mouse to engage with the editor functions to complete 5 tasks requiring them to replicate a series of images displayed in a column to the right. When they finish reproducing the image, they press a "Next" button to load the next image corresponding to the next task. Every task begins with all the toolbar commands deactivated, forcing the participant to decide which command to use first, instead of continuing with the last command used in the previous task. Participants are asked to think aloud (Hoffman, 1989) as they perform actions on the editor and are encouraged to use any command that they deem useful.

Because of the differences across digital environments, participants who fail at executing supported actions can be assisted. For example, if a participant attempted to select multiple elements by keeping the Control key pressed, I indicated that this is possible with the Shift key. I also gave confirmation when a participant expressed that a function was not present. For example, if a participant attempted to execute an "Undo" command, I indicated that history commands are not supported and that fixing mistakes would require reversing the steps manually or refreshing the browser to start the task over.

At the end of each task, the experimenter verifies that the result resembles the goal state before the participant proceeds to the next task. If noticeable differences are present, the participant is asked whether they are sure that the task is complete, pointing at the difference in question if they take more than 15 seconds to spot it. The participant can end the session at any point by pressing a "Finish" button on the side of the editor or by closing the browser window. At the end of the last task, the session is complete and participants answer a questionnaire about their performance and past experience with text and graphic editing environments, leaving the demographic items for the end. After submitting the answers, the participant is asked to upload a file containing the action logs from all tasks and the answers to the questionnaire.

4.2.5 Data Collection

I recorded audio from the call and video from the stream of the participant's shared screen. I took notes of the participants' responses about the purpose of the editor and the type of steps they describe to select content. Then, I collected action logs from tasks, including keystrokes of character and meta keys, toolbar interactions, tool commands on the canvas objects and clipboard commands.[2] These include timestamps for every action. I took notes of the participants' approached as revealed by their verbal protocol to assess their thought process. Finally, I collected answers to the end questionnaire. This was divided into three parts: demographics, daily experience with text and graphic editing software, and experience with the experimental editor, by assessing the use of 3 tools from the toolbar that were relevant to the task, their use of text or graphic editing approaches, a self-reported measure of the prevalence of one approach over the other, and knowledge of functions from other software that was applied to the tasks. This assessment was used in connection with the notes from the verbal protocol and action logs of their performance during the tasks. All data was referenced by participant number. The experimental design and data collection were approved by Inria's Institutional Review Board (COERLE).

4.2.6 Data Analysis

I carried out structured observations during the sessions, focusing on factors in the participants' preferences for an approach and mentions of past experience with other digital environments. I used my notes from each session to identify whether participants perceived the environment as text- or graphic-based. I later analyzed the responses about the selection technique that they performed (*SELTECH*), and classified them between *Text Selection* for text-based techniques, *Graphic Selection* for graphic-based techniques and *Other Selection* for alternative responses.

First, we performed independence tests to determine whether different types of TOOLBAR associated with a particular *SELTECH* (H1). Additionally, we used the action logs to analyze the participants' use

2 A complete list of the actions captured during sessions is included in Table b.2

of commands to complete the tasks. Next, we analyzed the number of command executions by their type—between "graphic" and "text"—and designated each task's *APPROACH* as based on *Graphic-only*, *Text-only* or *Mixed* types of commands. We consider an approach to be *Mixed* when it has more than 5% of the total number of its commands be of a secondary type, so as to discard unintentional or playful uses (see Appendix b for a more detailed explanation). We did not analyze data from commands that do not produce modifications to the canvas, e.g., selections, changing tools, etc.

Next, each participant was identified according to a type of PRIMING identifying the environmental cues (toolbar and selection feedback) according to their group (*Text Priming* for the *Text* group, *Graphic Priming* for the *Graphic* group and *No Priming* for the *Control* group). We performed independence tests of PRIMING and *APPROACH* to determine whether the environmental cues associated with particular approaches to complete the tasks (H2). This test was carried out both by aggregating the approaches from all tasks as well as by testing individually for each task. I developed the code for the statistical tests and produced the visualizations, including those to assess the proportion of command types and approaches involved in each task for each group (some are included in Appendix b).

4.3 RESULTS

In this section I present the results of our data analysis. We were interested in studying the priming effect of the toolbar layout and the interaction with the digital environment, i.e., the effect of *environmental cues*, in our case, stemming from the visualization of toolbars, i.e., the visual cues, and the interaction with objects in the environment, i.e., the interaction cues. For this purpose, we analyzed the participants' interpretations of the environment as well as their performance completing tasks. We made use of the responses about the selection technique after visualizing the editor interface and the action logs using the editor during the tasks. We complemented our results with observations gathered from my notes of the participants' verbal protocol.

4.3.1 Toolbars had an Effect on the Selection Technique

We counted the number of responses of each type from the description of the steps to select all the elements in the canvas. According to each TOOLBAR displayed, 10 (83%) with *Text Toolbar* described a *Text Selection* technique and 11 (92%) with *Graphic Toolbar* described a *Graphic Selection* technique, while we counted 9 (75%) among those with *No Toolbar* describing a *Text Selection* technique and the remaining 3 (25%)

Table 4.2: Count of selection technique class by interface layout of the editor.

Toolbar	**Selection Technique**		
	Text Selection	*Graphic Selection*	*Other Selection*
Text Toolbar	10	2	0
Graphic Toolbar	1	11	0
No Toolbar	9	3	0

describing a *Graphic Selection* technique. These observations include participants who did not necessarily recognize a text or graphic editing environment but that nevertheless described an expected selection technique. For example, **P5** (in *No Toolbar*) described the environment as: "*maybe a chat room*" before proceeding to describe a text selection technique. No other selection techniques were observed. Table 4.2 shows the counts for each TOOLBAR. All expected frequencies were above 5. A Chi-square test of independence showed a statistically significant relation between TOOLBAR and *SELTECH* ($\chi^2(2) = 16.4250$, $p = .0003$). We ran post-hoc pairwise comparisons using Fisher's exact test due to the small values in the sub-tables. Results showed significant differences in *SELTECH* between *Text Toolbar* and *Graphic Toolbar* ($p = .0019$) and *Graphic Toolbar* and *No Toolbar* ($p = .0055$) but not between *Text Toolbar* and *No Toolbar* ($p = 1.000$) –all p-values corrected with Bonferroni's technique for 3 comparisons.

We followed the recommendation to use the Chi-square test of independence only if 80% or more of the frequencies in the table are above 5 and none of them are below 1 (Agresti, 2006).

Our data suggests that the presence of both text or graphic toolbars separately had an effect on the participants' decision to perform a text- or graphic-based selection of the objects, respectively, thus verifying our first hypothesis (H1). However, when observing the relationships between toolbar conditions, we found that performing a graphic selection associated with displaying the graphic toolbar, while there were no significant differences in selection technique between displaying the text toolbar and not displaying any toolbar. In other words, both in the absence of a toolbar and the presence of the text toolbar, participants described a text selection technique. Arguably, this could be due to the contents of the document being perceived as text, rather than graphics.

4.3.2 Priming had a Effect on the Overall Approach Used

We analyzed the action logs of the tasks to extract the types of commands per task per participant. We designated the approach (*APPROACH*) based on the used command types, classified as *Text-only* when commands were only text-oriented, *Graphic-only* when commands were only graphic-oriented and *Mixed* when they used both types of commands. Table 4.3 and Figure 4.4 illustrate the counts of

Table 4.3: Count of *Text-only* (T), *Graphic-only* (G) and *Mixed* (M) approaches by task (see Figure 4.4 for a visualization).

	Task Number														
	1			**2**			**3**			**4**			**5**		
	APPROACH														
Priming	**G**	**M**	**T**	**G**	**M**	**T**	**G**	**M**	**T**	**G**	**M**	**T**	**G**	**M**	**T**
Text Priming	3	1	8	2	2	8	4	1	7	2	3	7	2	3	7
Graphic Priming	8	1	3	9	1	2	5	4	3	9	1	2	3	8	1
No Priming	8	0	4	7	2	3	4	1	7	5	3	4	3	7	2

these classes for every task and type of priming (PRIMING). Figure 4.4 suggests that priming with text toolbar and text interaction (*Text Priming*) associated with more participants using text approaches on all tasks. However, we needed a deeper analysis to compare the *Graphic Priming* vs.*No Priming* conditions, i.e., when users had graphic cues vs. no interaction cues before working on the tasks.

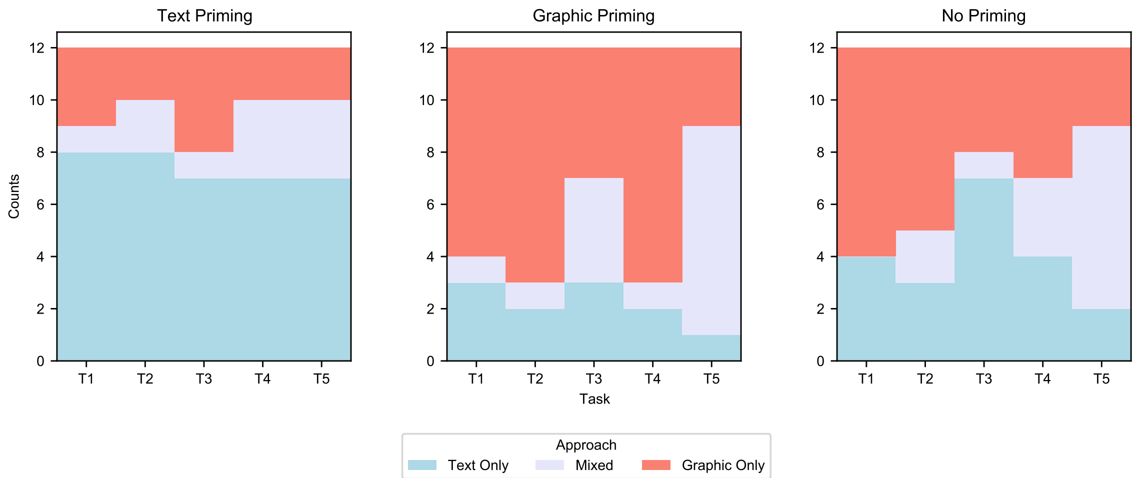


Figure 4.4: Frequencies of each approach among participants across tasks for each priming group (see Table 4.3 for the numbers).

We wanted to verify the effect of PRIMING with either text (*Text Priming*) or graphic (*Graphic Priming*) cues on the *APPROACH* used to complete the tasks. We first looked at the aggregated data of all the approaches across all tasks (120 observations from 5 tasks for 2 groups of 12 participants each). We tabulated the frequencies and observed that all the expected values were above 5. The chi-square test of independence shows that there was a statistically significant association between the type of priming and the approach used to complete the task ($\chi^2(2) = 24.466$, $p < .005$). An analysis of the standardized residuals shows that for the overall tasks, *Text Priming* caused a large deviation (values > 2 for small tables (Agresti, 2006)) in *APPROACH* with more *Text-only* approaches (4.84) and less *Graphic-only* approaches, while the opposite occurred with *Graphic Priming*, with deviation towards more *Graphic-only* approaches (3.92) and less *Text-only* approaches.

We repeated the analysis tabulating *APPROACH* for each task separately. Expected values were under 5 for all tables. The Freeman-Halton's extension of Fisher's exact test (Freeman and Halton, 1951) shows that there were statistically significant associations between the type of priming and the approach used only for tasks 2 ($p = .013$) and 4 ($p = 0.019$). An analysis of the standard residuals for these two tasks shows that the associations were the same as for the overall case but with a less pronounced effect (see Table b.1), given that residuals for individual tasks are smaller than in the overall case.

We repeated these analyses for 0%, 10%, 15%, 20%, 25% and 49% as additional thresholds defining *Mixed* approaches. The chi-square tests of independence showed statistically significant associations between Priming and *APPROACH* ($p < .05$) for all thresholds. For individual tasks, we observed similar significant results of the Fisher's exact test for all thresholds except for 49% where an additional significant association between Priming and *APPROACH* was found in task 1. Our results were consistent with our initial assessment of Figure 4.4 suggesting that, overall, priming with text interaction cues associated with increased use of text-only commands, whereby priming with graphic interaction cues associated with increased use of graphic-only commands to complete the tasks, thus supporting our second hypothesis (H2). However, a more granular analysis shows that this was the case only for tasks 2 and 4, while others did not associate with any particular approach. This difference with the overall case can be attributed to the difficulty of each individual task, forcing participants to try out alternative commands to reduce the effort involved in repetitive tasks. This analysis excludes control participants, therefore, before diving into our notes from individual tasks, we compared the approaches used by the primed groups with those used by the control group.

4.3.3 Control Participants and Primed Groups Followed Similar Approaches

We wanted to know if our interaction cues (Priming) had an effect on the choice of an approach (*APPROACH*), compared to not seeing a toolbar or performing a selection before the tasks as was done for the *Control* group. We aggregated the approaches from all tasks, collapsing the counts from *Text Priming* and *Graphic Priming* into one category for priming to compare with *No Priming* (180 observations). All expected

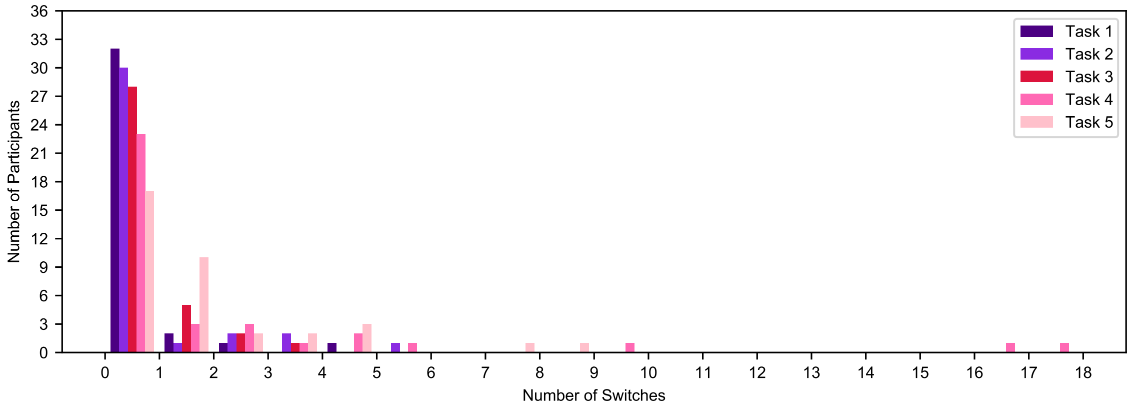


Figure 4.5: A histogram to display the number of participants who switched command types a given number of times for a given task.

values in the table were above 5. A chi-square test shows no statistically significant association between the type of priming and the approach used by participants ($p = .582$). The Freeman-Halton's extension of Fisher's exact test shows no statistically significant association between the type of priming and the approach in individual tasks ($p > .05$ in all tests). This suggests that participants in the control group (12) did not use significantly more mixed approaches than the primed group (24), despite the absence of priming. We therefore looked at our notes of participants performing the tasks to understand their process behind choosing an approach.

4.3.4 Most Participants Worked Based on One Representation at a Time

Most participants switched command types once or less, as seen in Figure 4.5 for any individual task. The two most extreme cases involved changing command types 16 and 17 times, both during task 4. For tasks 1, 2 and 3, most participants stayed on the same command type from beginning to end, suggesting that they assumed the first representation that they found (see Figure b.11 for a detailed visualization). On the other hand, in cases with only one switch, participants seem to have found a more convenient representation of the objects after a while, although also deciding to stay on it. For instance, Figure 4.5 shows task 5 as having the most participants switching command types only once, compatible with observations of participants using the fill tool in the beginning and later finding the text highlighter tool to use it until the end. In sum, for most tasks, participants chose to stick to one representation based on the command types that applied to it, rather than switch back and forth between command types according to what is most convenient.

4.3.5 Some Approaches Originated in Personal Preference

Some participants' descriptions of their experience confirm the effect of our priming technique. For example, **P19** (*Text Priming*) used text commands exclusively in all tasks but mentioned having recognized some of the tools from Adobe Illustrator. When asked for a reason why he would not use the pointer tool for selecting and moving objects, he said: "*I didn't use it because I got caught on the idea that it was more a text editing environment which respects lines and spaces.*" That is, **P19** framed it as a text editing and decided on the approach accordingly. **P15** (*Text Priming*) also used text commands exclusively and acknowledged the effect of priming more directly: "*Because of the way the text was highlighted when I selected it—that is, as in a text editor and not in a box—I assumed that the program worked primarily as a text editor and in addition as a graphical editor.*" This also applied to participants such as **P1**, **P8**, and **P15** with text approaches, or **P9**, **P13**, **P16** and **P29** with graphic approaches, who all stated that the interaction cues made them think of the problem according to the environment.

However, priming seems not to have been at the root of some participants' approaches based on a single command type. A good example of this is **P31** (*Graphic Priming*), who was the only participant primed with graphic cues to perform all tasks exclusively with text-based commands. When asked about this, he said: "*I mostly use text editors [in daily life], rather than graphic editors. That's why I'm more comfortable with text [editing] and use [it] whenever I can.*" In particular, during task 3, **P31** sought a command to change the background color of text characters, stating: "*I will use [the highlighter tool] because I think [the fill tool] fills the [graphic] shapes,*" thus identifying the task as text editing and then choosing tools according to what seemed appropriate. Like **P31**, **P17** (*Text Priming*) was the only example in the other direction, using exclusively graphic-based commands after being primed with text interaction cues. Regarding this fact, she said: "*Even if I initially saw it as a text editor, after using it—and even more so with the left toolbar—I found it more comfortable to edit as if I was 'dragging' images instead of chunks of text.*" **P34** (*Graphic Priming*) was an example in between these two. He performed tasks 1, 2, 4 and 5 using exclusively graphic-based commands and decided to explore an alternative way in task 3, using the highlighter after having tested the fill tool. However, although he had discovered and used the text highlighter to complete task 3, he carried out task 5 using the fill tool exclusively, thus completing the task rather inefficiently. Asked about this, he said: "*[I] found a way that worked and stuck with it, plus, I have a big bias towards graphical design.*" Therefore, although there was a confirmed priming based on the interaction cues, some participants chose command types as a matter of personal preference.

4.3.6 Some Mixed Approaches Originated in Efficiency

Some participants expressed using commands for their perceived efficiency. For example, **P27** (*Graphic Priming*) recognized the possibility to use text commands in addition to graphic commands in the digital environment, but preferred using the latter because: "*It was just easier to stay within the graphic mode as it allowed for easier copy and paste rather than moving the hand between my trackpad and keyboard all the time.*" **P23** (*Text Priming*) saw a "game" in the problem, trying to find the "correct" answer. When reaching task 5, even before checking what changes were needed, she stated: "*I'm sure there is an easy way.*" She tested different ways to color multiple backgrounds at the same time using the fill tool, which was not supported. After coloring 2 lines using the fill tool on each character individually, she thought of text selection as a way to select multiple objects and tried to combine it with the fill tool, causing the text selection to be cleared. She performed a second text selection revealing her thought process about the environment: "*this is just text,*" after which she used the text highlighter tool for the remaining lines in task 5, thus characterizing it as a mixed approach. In this sense **P23** represents participants such as **P10**, **P20**, **P21** or **P26** who were concerned with finding the "right" approach and explored the interface focused on the tools that would offer it, with seeming disregard for which actions were supported by the objects.

4.3.7 Some Mixed Approaches were Based on Inaccurate Mental Models

Finding out about commands of the secondary type did not necessarily translate into an accurate mental model (Johnson-Laird, 1989) of the editor. As a matter of fact, many participants discovered that a secondary type applied "by accident," guided by the symbols on the buttons and an interest in efficiency, rather than a conviction about their pertinence to the object type. For example, **P33** (*Text Priming*) completed the first two tasks with a *Graphic-only* approach, switching to a *Text-only* approach for task 3 and later using *Mixed* approaches for both tasks 4 and 5. As a reason, he said: "*The order of the elements [in the canvas] made me think of a graphic editing environment, but after verifying that I could select text using the '[text] cursor' button (...) [I realized] that it was more convenient to select everything as if it were a text editor.*" In other words, **P33** inferred the possibility to interact with the objects as text, as a consequence of a text interaction. While some participants were motivated to discover more useful commands by a problem-solving attitude, **P33** represents a subgroup who switched their perception of the objects after an interaction and never reflected back on the previous perception. In sum, while some participants managed to incorporate the dual "nature" of objects in their support of both types of commands, thus consciously following mixed approaches, **P33** stands as an example of participants who *replaced* a representation with another, providing evidence of an incomplete mental model of the digital environment.

4.3.8 All Participants Identified Other Digital Environments as Sources of Knowledge

All participants gave one or more examples of applications that inspired their decisions on how to complete the tasks. 24 participants mentioned Microsoft Word as their inspiration for text editing strategies. When it comes to graphic editing strategies, 13 named Microsoft Paint, 7 Adobe Illustrator, 7 Adobe Photoshop and 5 Microsoft PowerPoint among the most mentioned applications on which to base their solutions. In particular, **P7** recalled PageMaker at the end of the session because of its mixed text- and graphic-oriented tasks—referring to the pointer tool as the cue—, although this participant used only text-based commands.

P18 was quick to associate Microsoft Word with the use of the text highlighter and clipboard commands, but had difficulty describing how a text application influenced the way in which she operated the text cursor, stating: "*These are things that you don't know that you know... they are just there.*" When asked to clarify, she added: "*Let's assume I used knowledge about the [text] cursor [from Word] but... it is simply something that I know that is there, **like knowing how to walk**.*" In general, participants omitted the source of their knowledge about how to operate with text, limiting themselves to mentioning the highlighting and clipboard commands. In the next section, I discuss these results in relation to mechanical knowledge.

4.4 DISCUSSION

Our results suggest that the environment primed the participants' knowledge about other graphic and text editing environments. This is expressed in the participants' description of the selection action after observing the interface with either one or no toolbar. Participants in the control group—who did not have any toolbars in the first stage of the experiment—described predominantly more text selections at this stage, suggesting that object representations on the canvas were biased towards text-based interactions. However, this was later contrasted by the general strategies used by control participants, who did not elicit a preference for text-oriented commands. In this section, I discuss the knowledge involved in using commands after participants have been primed with toolbars and its relationship with mechanical knowledge.

4.4.1 Toolbars and Effects Primed the Object's Possibilities for Action

Most participants elicited a preference for the type of command used in their first task throughout the remaining tasks, compatible with "functional fixedness" (Duncker and Lees, 1945) on the type of the objects. This is supported by the very few command type switches observed across all participants. In particular, participants in the control group did not fixate on their assumptions from their first look—where they predominantly described a text-based selection—but rather split almost evenly between using text- and graphic-based commands after the two toolbars were revealed. This behavior suggests that the effects of commands on the objects offered confirmation of their type, thus fixating on one for the remaining of the session, e.g., "*I tried and could select objects as text in the beginning, therefore, the objects are text.*" This could also be observed as "satisficing," (Cockburn et al., 2014; Simon, 1956), supported by the fact that knowledge of interactions with text objects does not usually apply to graphic objects—and vice-versa—in conventional digital environments. Arguably, control participants used the first command or commands to probe the objects for their "nature," similar to using physical tools to test the reality of their affordances, such as when testing a pen on a slippery surface or probing the grip of pliers on an odd-shaped object.

Contrary to participants in the control group, those who were subject to either a text- or graphic-oriented interaction elicited a preference for the command types associated with it, supporting our second hypothesis. This suggests that the environment's environmental cues—rather than the commands' effects—associated with fixating the objects' *possibilities for action*. This was verified by the low number of command type switches across all participants and tasks. This observation is repeated in some participants in the priming groups who were not sure about the object type until probing with a command, but whose first choice of a command to test belonged in the toolbar corresponding to their priming. In sum, most participants who did not interact with the system before performing tasks seem to have been primed by the effect of the first command that they used. On the other hand, most participants who were assigned a particular interaction before performing tasks, used predominantly more commands associated with the initial interaction.

4.4.2 Knowledge Centered on the Effects of Commands

All participants knew of a "proper" command to accomplish a task based on past experience with other environments, e.g., set the text background color with the paint bucket or use the text highlighter. However, some seemed unaware of—or simply disregarded—the ambiguous nature of the objects on the canvas, for example by editing them as text and then choosing the paint bucket to change the text highlight color without looking for a tool found in text editing environments. Moreover, Microsoft Paint was frequently cited as a source of graphic editing knowledge, despite the fact that it is a raster-based environment—where objects are "stuck" on the canvas and need to be cut-out to be moved or copied—while ours used vector-based interaction. Therefore, participants did not seem concerned with the specifics of which commands apply to given objects. This suggests that the knowledge primed by the toolbars was neither about the specific objects that were initially identified (e.g., text) nor about the commands of specific environments (e.g., a text editor). Rather, participants elicited knowledge about the *effects* caused by shortcuts (e.g., Ctrl+C) or buttons in the toolbars, e.g., the paint bucket changes background colors, regardless of the association of the command with an object type. Arguably, this is most visible in participants using mixed approaches, who managed to use commands of both types seemingly inadvertently.

Humans can use more than one physical tool to produce the same effect on an object, switching between them seamlessly. For example, one can cut an apple with a serrated knife by sawing with it, or with a kitchen knife by pressing the edge down on it. In a similar way, users can achieve the same effect on a digital object through different techniques, e.g., moving an icon into a folder by using clipboard shortcuts, drag-n-drop interaction, etc. For physical tool use, mechanical knowledge models the match of the knife's and the object's properties with the appropriate technique (Osiurak et al., 2010). For digital tool use, users have learned specific techniques that they apply to predefined objects. However, our results show that users elicit behavior similar to that based on mechanical knowledge: they managed to interact with novel objects supporting text and graphic actions without the need to reflect on the commands as text- or graphic-oriented. This suggests that participants using mixed approaches resorted to abstractions of these commands—learned from experience in other digital environments—applying them to the objects in the canvas according to what they have learned from environmental cues and their own interactions. These results offer additional support to our findings in Chapter 3, with participants eliciting knowledge compatible with a "mechanical" knowledge of the digital world.

4.4.3 Implications for HCI

Mechanical knowledge models the basis for how we transfer knowledge of physical tool use to new tasks, supporting the Technical Reasoning hypothesis (Osiurak et al., 2010). In the digital world, users acquire knowledge of interactions and carry it to new interfaces, as is shown in studies of text-based interfaces (Polson et al., 1986) and GUIs (Rieman et al., 1994). Our results show that these principles can be primed by presenting environmental cues about the available tools, which can have an effect on the users' strategies to perform tasks involving digital objects. Consequently, the presentation of tools in digital environments can guide but also hinder users from finding alternative strategies to complete tasks, potentially to the detriment of technical reasoning. Arguably, giving users more liberty to make their own tool sets for tasks could create the conditions for more creative solutions, based on the users' preferences and experience acquired with digital tools.

Our results support moving away from the application-centric paradigm (Beaudouin-Lafon, 2017; Nouwens and Klokmose, 2018)—which forces users to accept the choice of tools offered to them—, towards leveraging human cognitive abilities used for interaction in the physical world (Jacob et al., 2008). Having tools outside of applications would allow users to engage with digital tools they feel comfortable with, much like with a physical tool set. For example, users could create a document by switching between graphic-editing tools for images and drawings, and text-editing tools for writing. This would lead to users customizing and adapting these tools to their needs, resonating with technology appropriation (Dix, 2007). This notion of ownership of tools has already been explored in Instrumental Interaction (Beaudouin-Lafon, 2000), envisioning interfaces inspired by how humans use physical tools, where, for example, a color picker could change the color of anything having such a property (Beaudouin-Lafon et al., 2021). In this sense, our results contribute to the relevance of the Technical Reasoning hypothesis to digital environments, providing a model of how users can make sense of digital tool- and property-based interactions.

4.5 SUMMARY

Mechanical knowledge models our understanding of physical properties and technical laws in the technical reasoning model (Osiurak et al., 2010). We studied participants interacting within an editing environment to complete tasks without being informed about the possibilities for action with its digital objects. We controlled the initial visibility of a text- or graphic-oriented toolbar and the first interaction with the environment, so as to induce knowledge transfer, measuring its effects on the strategies followed by participants to complete a series of tasks. We found that some participants used both text- and graphic-based commands on the same objects seamlessly, compatible with how humans switch between physical tools to accomplish the same goal regarding an object. We argued that this is compatible with the mechanical knowledge posited by the Technical Reasoning hypothesis. Additionally, we found that interaction with the environment before performing tasks associated with using commands corresponding the interaction type, exclusively. Furthermore, the performance of a control group who was not exposed to environmental cues did not associate with an increased use of a particular command type. This suggests that our environmental cues primed the participants' knowledge about what commands to use in order to alter the digital objects, compatible with the transfer of a "mechanical" knowledge of the digital world. This supports our hypotheses that environmental cues—toolbar and selection interaction—prime the participants' knowledge of a digital environment (H1 and H2). In conclusion, this work supports the hypothesis that users possess accumulated knowledge about interactions in digital environments compatible with mechanical knowledge, which can be primed by visual and interaction cues. This extends the work I presented in Chapter 3, providing additional support to model the users' understanding of digital tool-based interactions through the technical reasoning model.

5 TEXTLETS: BRINGING NOVEL TOOLS TO TEXT EDITING

Previously, I presented evidence to support the notion that users perform technical reasoning when using *familiar* digital tools in *unusual* ways (Chapter 3), and that environmental cues prime the users' "mechanical" knowledge of digital tools and objects (Chapter 4). For the last chapter of this thesis, I switch the focus to the use of *novel* digital tools in text editing environments.

Text editing was once considered a *killer app* of personal computing (Bergin, 2006). Editing text used to be the first skill a novice computer user mastered, and all personal computers are sold with a word processor. Many professions require advanced text editing skills to ensure consistent use of terms and expressions within structured documents, such as contracts, patents, technical manuals and research articles. For example, lawyers begin each contract with a list of defined terms, and must use them consistently thereafter. This is critical, since *minor* wording changes can have serious legal implications. For example, some patent specifications use the expression "comprises" to indicate that the invention consists *at least* of the elements listed afterwards; whereas the expression "consists of" is used before listing exactly and exclusively the list of elements that are considered in the invention, therefore indicating a significantly different scope of protection.

Some institutions impose official constraints to the contents of certain or all parts of a document, e.g., the US Patent and Trademark Office (USPTO) does not accept "new," "improved" or "improvement of" at the beginning of a patent title. Word limits are also common, such as the European Patent Office's 150-word limit for patent abstracts. Despite their many features, standard word processors cast a wide net in terms of use cases but lack specific functions for professional needs. For example, although spell checking is common, flagging forbidden words or ensuring the consistent use of particular terms requires the user to check "manually," e.g., by running individual searches for specific words, going back and forth between the highlighted results; When facing word count limits on specific parts, word processors offer real-time counts of words and characters for the whole document, but for a single selection may require displaying the count on-demand.

In this chapter, I present a revision of joint work carried out with Han L. Han,[1] which includes variations from the terms used in the original article (Han et al., 2020). Our motivation was to increase the power of expression in text editing settings, while preserving simplicity. We focused on a group of authors of technical documents (that we considered *extreme* users) and sought to answer the following questions:

1. How do current software tools support professional technical writers?
2. How do professional users manage constraints and consistency when editing technical documents?
3. How can we create tools that better support these needs?

We conducted an interview study with legal and technical document editors, which highlighted some of the issues they face with software used in their tasks. The results of this study led to introduce the concept of *Textlet*, which reifies the notion of selection in text documents (Beaudouin-Lafon and Mackay, 2000)—turning a transient concept into a persistent, interactive object. We accompanied the concept with examples of novel tools that are built on it, addressing some of the needs identified in the study. Additionally, we implemented some of these textlet examples in prototypes, one of which we used in a short observational study to better understand how textlets support a selective Find & Replace operation over text. This chapter concludes with our argument for Textlet as a generative concept for creating powerful new tools for document editing.

5.1 RELATED WORK

We review research related to both word processing and code editing tools and practices. The latter is a particularly interesting form of technical document that requires professional software developers to manage multiple internal constraints, as well as the specific tools developed to ensure internal consistency in code that may inform the design.

5.1.1 Text Editing Practices

Text editing was an active research topic in the 1980s when word processors became mainstream. For example, Card et al. (1980) modeled expert users' behavior in manuscript-editing tasks; Tyler et al. (1982) investigated the acquisition of text editing skills; and Rosson (1983) explored the effects of experience on real-world editing behavior. Others

1 With additional supervision by Wendy E. Mackay

examined paper-based editing practices to improve computer-based text editing (C. C. Marshall, 1997; K. O'Hara and Sellen, 1997; Sellen and Harper, 1997) and collaborative writing (Baecker et al., 1993; Churchill et al., 2000; Noël and Robert, 2004). More recent studies identified issues with modern word processors. For example, Srgaard and Sandahl (1997) found that users rarely take advantage of text styles, and argue that this is because styles do not impose restrictions on the document structure. Alexander et al. (2009) found that although users often revisit document locations, they seldom use the specific "revisitation" tools found in Microsoft Word and Adobe Reader. Chapuis and Roussel (2007) examined users' frustration with unexpected copy-paste results due to format conversion. The authors identify a clear mismatch between the advanced features offered by modern word processors and actual user practice, and highlight the need for new tools and concepts. While the previous work focuses on general editing tasks, we were particularly interested in how authors manage constraints and ensure consistency when editing structured technical documents.

5.1.2 Tools to Support Text Editing

Researchers have created a variety of text editing tools to support annotation (Schilit et al., 1998; D. Yoon et al., 2013; Zheng et al., 2006), navigation (Alexander et al., 2009; Laakso et al., 2000; Wexelblat and Maes, 1999) and formatting (Myers, 1991); as well as distributing editing tasks Bernstein et al., 2015; Teevan et al., 2016 and taking advantage of a document's structure (Miller and Myers, 2002a). In particular, Cut, Copy & Paste (Bier et al., 2006; Stylos et al., 2004) and Find & Replace (Beaudouin-Lafon, 2000; Miller and A. M. Marshall, 2004) are especially relevant to supporting internal document consistency.

Chapuis and Roussel (2007) propose new window management techniques to facilitate copy-paste tasks. Stylos et al. (2004) extracts structure from text, such as an address with different components, so that they can be pasted with a single operation. Multiple Selection (Miller and Myers, 2002b) offers a smart copy-paste function that is sensitive to source and destination selections, while Entity Quick Click (Bier et al., 2006) extracts information to reduce both cursor travel and number of clicks. Cluster-based Find & Replace (Miller and A. M. Marshall, 2004) groups occurrences by similarity, allowing entire clusters to be replaced at once. The Instrumental Interaction version of the Find & Replace tool (Beaudouin-Lafon, 2000) highlights all items at once so users can make changes besides in linear order as with conventional text editors.

Commercial applications such as *Grammarly*[2] check grammar and spelling by suggesting alternative wording, style and tone, among other features. However, they do not ensure a consistent use of specific terms, e.g., making sure that parties in a contract are always referred to by the same name. Other software tools automatically generate consistent references, including *Mendeley*[3] and *EndNote*[4] for researchers, and *Exhibit Manager*[5] for legal professionals. Although automated reference management seems to solve some problems, users still lack flexibility with these solutions, e.g., creating a custom citation format. Additionally, these tools do not integrate with the word processor's workflow but rather exist as separate parts within their own floating window or dialog, potentially distracting users from their document.

5.1.3 Tools to Support Code Editing

Code editing has been widely studied, especially around the use of Cut Copy & Paste commands (Kapser and Godfrey, 2008; Kim et al., 2004), online resources (Brandt et al., 2009), diagrams and drawings (Cherubini et al., 2007), and performing maintenance tasks (Amy J. Ko et al., 2005). A key challenge emerging from these studies is how to manage code dependencies, i.e., making sure that for every reference to a variable, function, class, etc., there is a declaration according to the language's rules. For example, Kim et al. (2004) found that programmers rely on their memory of copy-pasted dependencies when they apply changes to duplicated code. Amy J. Ko et al. (2005) identified both *direct* dependencies, e.g., going from a variable's use to its declaration, and *indirect* ones, e.g., going from a variable's use to the method that computed its most recent value, and proposed ways of visualizing these dependencies in the editor. While technical document constraints are less stringent than in computer code, there are arguably certain commonalities with human language to exploit.

We can see program code as an extreme case of a technical document with many internal constraints. For example, Toomim et al.'s technique (Toomim et al., 2004) supports editing duplicated code and visualizing links among duplicates. To help programmers use web examples more efficiently, *Codelets* (Oney and Brandt, 2012) treat snippets of code examples as "first-class" objects in the editor, even after they are pasted into the code. Kery et al. (2017)'s tool for lightweight local versioning supports programmers performing exploratory tasks, while *AZURITE* (Y. S. Yoon and Myers, 2015) lets programmers selectively undo fine-grained code changes made in the editor. *Barista* (Andrew J. Ko and Myers, 2006) supports enriched representations of

2 https://www.grammarly.com
3 https://www.mendeley.com
4 https://www.endnote.com
5 https://www.exhibitmanager.com

program code, while *Whyline* (Andrew J. Ko and Myers, 2004) and *HelpMeOut* (Hartmann et al., 2010) support debugging tasks. A challenge lies in how to build upon these concepts and tools but for non-programmers who manage less syntax-based constraints in technical documents.

5.2 STUDY 1: INTERVIEW WITH LEGAL PROFESSIONALS

Editing technical documents requires a complex process (Cohen et al., 1999), especially to maintain the document's constraints and internal consistency (Farkas, 1985). We conducted critical object interviews (Mackay, 2002) to better understand how professionals manage to follow such constraints and keep consistency in their technical documents.

5.2.0.1 Participants

We interviewed 12 participants (3 female, 9 male; aged between 24 and 50 years old). Their occupations included: a Contract Manager, a Legal Affairs Director, a Ph.D. Candidate in International Water Law, lawyers, Patent Attorneys, and Patent Engineers. All used Microsoft Word on either Windows (11/12) or macOS (1/12) platforms; only one participant used the latest version at the time of conducting the study (2019).

5.2.0.2 Procedure

We ran four pilot interviews with colleagues to establish the protocol, then visited participants at their workplaces and asked them to show us specific examples of their current digital and physical documents. We asked them to describe a recent, memorable event related to editing one of these documents, either positive or negative. All interviews were conducted by Han and I in English, each lasting between 45 and 60 minutes.

5.2.0.3 Data Collection

All interviews were audio recorded and later transcribed by Han. Each of us took hand-written notes which we later unified in one shared digital document. We were not allowed to take video or photographs during the interviews due to the sensitive and/or confidential nature of some of the documents that were discussed.

5.2.0.4 *Data Analysis*

Han and I analyzed the interviews performing a reflexive thematic analysis (Braun and Clarke, 2019). We generated codes and themes both inductively (bottom-up) and deductively (top-down) looking for breakdowns, workarounds and user innovations. After interviewing eight participants, we conducted the first analysis generating codes independently, later merging overlapping ones and grouping them into larger categories, focusing on the participants' editing behavior. Han also created story portraits (Jalal et al., 2015) to graphically code the data, which helped us engage with the collected notes.We agreed on the final themes after three iterations.

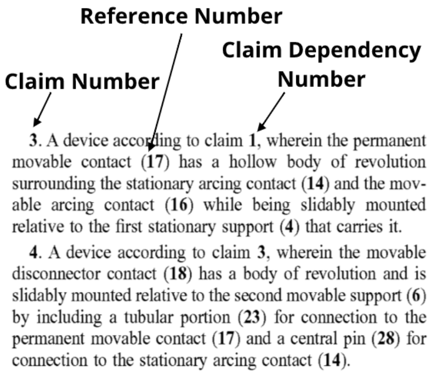
5.3 RESULTS AND DISCUSSION

We identified six themes related to the participants' practices with text documents: *maintaining term consistency*, *managing dependencies by hand*, *reusing content*, *visiting and revisiting document locations*, *managing annotations*, and *collaboration*.

5.3.1 Maintaining Term Consistency

All participants rely on their memory to maintain a consistent use of terms, which are often defined at the beginning of the document. This causes problems, e.g. when **P7** (Legal Affairs Director) struggled to use the full name of a party across the document and **P5** (Patent Attorney) often made the wrong choice between two words with highly similar meanings. Sometimes terms must be changed, e.g., shifting from British to American English or when the client prefers different wording to be used. To avoid introducing inconsistencies, lawyers must update each term and its variations, e.g., singular vs. plural, and adjust verbs (**P1**), articles (**P1, 6, 7, 9**) and pronouns (**P9**) accordingly.

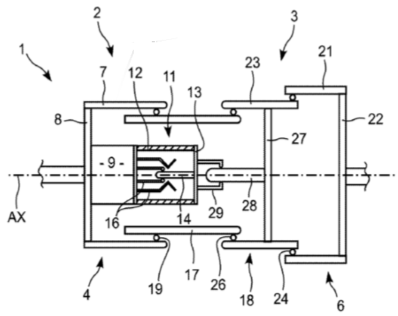
Although all participants use Word's "Find & Replace" dialog to make consistent edits, most (9/12) avoid the *Replace All* function: "*It is too risky*" (**P4**); "*I will not let the computer do it for me*" (**P6**); and "*I prefer to do it manually*" (**P5**). Instead, they search for the term, navigate the document with the *Find* button, check that they want to effectively change the occurrence and then use the *Replace* button, for each term. They ensure the correctness after replacing by assessing each new term's context: "*We have to [adjust] the verb [according to] the subject. It's a lot [of work]*" (**P4**). Reviewing the context is also essential for avoiding partial matches, i.e., when the search term matches a sub-string within a larger word (**P3, 11**), which requires performing additional Find & Replace operations.



3. A device according to claim 1, wherein the permanent movable contact (17) has a hollow body of revolution surrounding the stationary arcing contact (14) and the movable arcing contact (16) while being slidably mounted relative to the first stationary support (4) that carries it.

4. A device according to claim 3, wherein the movable disconnector contact (18) has a body of revolution and is slidably mounted relative to the second movable support (6) by including a tubular portion (23) for connection to the permanent movable contact (17) and a central pin (28) for connection to the stationary arcing contact (14).

(a) Claims



(b) Drawing

Figure 5.1: Example illustrating the numbering schemes used in a patent claim.

In summary, participants maintain consistency for terms primarily "by hand," which they find cumbersome and prone to error. Most avoid to replace all the terms in one action because they do not trust the results and cannot easily check them after they have been done.

5.3.2 Managing Dependencies by Hand

We understand a dependency as two or more sections of text that must be kept identical in whole or in part. Most participants (8/12) rely on their memory to manage such dependencies, and resort to keep them in sync by hand. We identified three areas of problems with dependency management: when managing consistency across pasted copies, when numbering items, and when managing cross-references.

All patent attorneys (4/4) copy text from the *Claims* section of the patent to the *Summary of the Invention* section while writing a draft of it. However, when they make changes to the Claims section, they often forget to update the Summary section accordingly: "*because it is not automatically updated with the claims, I can easily forget to update*" (**P6**).

Patents contain three types of numbering systems (Figure 5.1): one for claim numbers, one for claim dependency numbers (used when the current claim depends upon another claim), and one for reference numbers (used to point at parts of an illustration). Most patent attorneys (3/4) manage these numbers by hand instead of using, for example, Word's cross-reference feature, and typically number illustrations using non-consecutive ascending numbers so that they can add new references to parts in it without having to change numbers already used in the text.

Most lawyers and patent writers insisted on requiring stringent control of the text editing process—as opposed to automating parts of it—, especially when it comes to the Claims section of patents, however tedious this may admittedly be for them. Even participants who were comfortable using automatic features do not rely on automatic numbering. For example, **P7** said: "*Most of the time, I prefer if something can be automatically achieved*", yet, he avoids automatic numbering: "*I cannot really tell you why. One reason might be that if I have automatic numbering set up, this would have become paragraph 2 and all the numbering of the claims would have been changed... I would not be very happy*".

In sum, participants' key reasons for avoiding automatic numbering include (1) their inability to differentiate automatic from normal numbering, unless they select the text; (2) incorrect display of references, e.g., when items are added to a list, until a manual update is triggered; and (3) invisibility of dependencies after an update, since they lack feedback and cannot be sure that the changes are correct after they have been applied.

5.3.3 Reusing Content

All participants reuse documents to start new ones, incorporating their contents and styles as templates. When copying and pasting text for reuse, they must often perform format edits to the content before copying or adapt the format after pasting, e.g., using the brush tool (**P6**) or a macro (**P7**). Pasting text with embedded Word styles can bring *wrong* styles into the new document, resulting in a polluted style set in the destination document: "*When you copy-paste into a document, you can import the style of the [original] document. Too many unnecessary styles makes the document heavier and you have to remember which style to use. This is a mess*" (**P4**).

Most participants (10/12) use documents as templates to create new ones. Although this is useful for writing letters, filling cover pages, generating tables and managing formatting consistency, participants still struggle with formatting issues caused by style conflicts. In sum, although they benefit from the features allowing to reuse content, participants are not satisfied with the accompanying introduction of inconsistencies in formatting.

5.3.4 Visiting and Revisiting Document Locations

Participants rarely write or edit documents sequentially and they often go back and forth revisiting parts of a long text. For example, **P7** created a set of keyboard shortcuts to "*jump to different parts of the document*" to facilitate switching focus very often. This is consistent with Alexander et al.'s findings concerning users' revisiting behavior (Alexander et al., 2009).

Participants also need better support for this switch when systematically going through the whole document, e.g., incorporating edits one by one or performing Find & Replace tasks. In particular, the latter example often involves checking text replaced earlier thus calling to revisit a just visited part of the text. Unfortunately, Word imposes a sequential replace workflow, hindering users from returning to the previous replaced term without losing changes: "*The problem is that I cannot check. It made the replacement and it goes to the next occurrence, so I don't see what just happened*" (**P7**). **P8** works around this problem turning Word's *Track Changes* function on to leave a trace of each replacement. In sum, participants experience difficulty navigating documents, especially with respect to tracking recent or often-visited parts of the document.

5.3.5 Managing Annotations

Some participants (4/12) customized or re-purposed functions to make comments and annotations in documents, rather than using the features designed for such tasks included in their word processor. For example, **P5** uses footnotes with the intention to leave comments for his clients because he dislikes how the text gets smaller when using Word's Track Changes. **P7** avoids using Track Changes altogether and uses colored text between lines to encourage their active reading and convey the importance of certain of these comments to his clients.

For documents with two or more co-authors, some participants (4/12) complained that the Track Changes feature introduces more problems than it solves (**P2, 4**) and makes it difficult to understand the modifications (**P5, 7**). Instead, some (3/12) use the comparison function *after* making changes, to make modifications visible to their clients.

Interestingly, **P9** has used the comparison function to "cheat:" He does use Track Changes but in order to hide the timestamps showing that he has worked outside work hours, he creates a copy of the document without the changes, tracks changes in the copy working at home and later merges the changes into the original document using Word's Compare function at the office on Monday. Summing up, participants find comment and annotation tools frustrating and inflexible, forcing some participants to make deviant uses of existing features to replace them.

5.3.6 Collaboration

Most participants (11/12) collaboratively edit documents, creating versions that "branch" from the previous one and later merging all these versions into the main "trunk." When versioning, participants exchange documents via email and keep all successive versions of the document as separate files to keep track of changes made in each stage. They use simple suffixes to identify these versions, e.g., `_V1`, `_V2`, etc., which makes files with similar content hang around but being hard to find again. **P12** complained that she created eight versions of the same document even though she made only minor changes each time. Although a notion such as File Biography (Lindley et al., 2018) could help them manage these issues, they did not seem to know or apply it. Local versioning of files explored for code editing in Variolite (Kery et al., 2017) would also be useful in these cases, but standard word processors do not support it.

Some participants partition the master document for each co-author to edit, so that everyone has the same copy and works only in their designated section, which gets "merged" by copying and pasting at a later stage. When a version of a document is sent out and then returns with proposed changes, authors have to merge these changes into the master document. Even though they use the Track Changes feature of Microsoft Word, they usually make the changes by hand, going through each document and deciding which edits to incorporate. However, as we have identified before, this type of merge in Microsoft Word induces style pollution through simple copying and pasting which includes formatting (**P2,4**). Because the style panel in Microsoft Word is not displayed by default when users open a document, it is often hidden from users, leaving formatting and style inconsistencies undetected as a result.

A person responsible for merging may not automatically accept all the changes for various reasons: "*It might destroy the way [the text] was presented*" (**P5**), "*We do not consider all comments*" (**P6**), "*[clients'] comments are difficult to understand*" (**P7**), or the changes require other modifications to be made in other parts of the text (**P7**). In sum, we found that participants keep track of versions of their documents manually, even for minor edits, and participate in a process where these documents are merged by simple copying and pasting, incorporating changes one by one as they struggle with style pollution.

5.3.7 Summary

This study showed not only that most participants need to maintain consistent uses of terms, but also that they manage these resulting dependencies mostly by hand. They struggle to keep formatting consistent when reusing text and lack tools for frequently changing the context of their modifications within their document, flexibly generating annotations, or collaborating asynchronously. Participants elicit "mechanical" knowledge about text editing environments but seem to lack proper tools to perform repetitive editing tasks more efficiently while maintaining a sense of agency. Based on these results and the theoretical framework provided by Instrumental Interaction (Beaudouin-Lafon, 2000), we proposed a general solution to address some of their needs.

5.4 TEXTLETS: REIFYING SELECTION

General-purpose word processors such as Microsoft Word have hundreds of features. As seen in *Study 1*, users who know that a feature such as "Replace All" or automatic numbering exists may often prefer making changes by hand to stay in control. Rather than proposing specific new features to address the various use cases we observed, we seek a general approach that fits how they actually deal with text.

Word processors rely heavily on the concept of selection: the user selects a piece of text and then invokes a command in a menu, toolbar or keyboard shortcut that affects the content of the selection. However, the selection is transient: selecting a new piece of text or using an unrelated function causes the previous selection to be lost.

We introduce the concept of *textlet* as the *reification* (Beaudouin-Lafon and Mackay, 2000) of a text selection into a persistent, interactive, first-class object. A textlet represents a piece of a text document identified as interesting to the user. They can be highlighted in the document itself, listed in a side panel, or visualized through other interface elements, e.g., a scroll bar for easy access.

To create a textlet, a user simply selects a piece of text and invokes a command, e.g., `Ctrl+T`. The selected text is highlighted and the textlet is listed in the side panel. Textlets can also be created automatically and contained by higher-level textlets, which I call *host textlets*. For example, a *search textlet* (or *searchlet*) creates textlets for each occurrence of a word in a document (see Figure 5.2). The *searchlet* appears in the side panel and the user can modify the search string thus deleting the textlets associated with the previous occurrences and creating new ones. Additionally, occurrence textlets are automatically updated when editing the document, so that a corresponding textlet reflects any change that has been made to the text selection that contained its occurrence.

"Host textlets" were originally named "grouplets" (Han et al., 2020, p.7).

The power of textlets comes from the *behaviors* associated with them. The most basic behavior is to (re)select the piece of text from the textlet in the document, e.g., by double-clicking the textlet representation in the side panel. *Searchlets* carry behavior to find occurrences, create textlets associated with them and display these textlets hierarchically. Other ideas for behaviors include the ability to change or automatically generate the content of the text, to change its style or to attach annotations or additional information, such as character or word count. Creating textlets with different behaviors leverages the power of polymorphism (Beaudouin-Lafon and Mackay, 2000) because a single object (i.e., the reified text selection) addresses a variety of commands (searching, counting, referencing), providing users with a unifying concept to operate with text documents. This slightly extends the definition by Beaudouin-Lafon and Mackay (2000), which focused on polymorphic instruments.

The rest of this section illustrates the power of textlets by describing how different behaviors address issues arising from *Study 1*. We propose solutions for use cases (summarized in Table 5.1) based on the design of textlets.

5.4.1 Textlets for Consistent Reuse

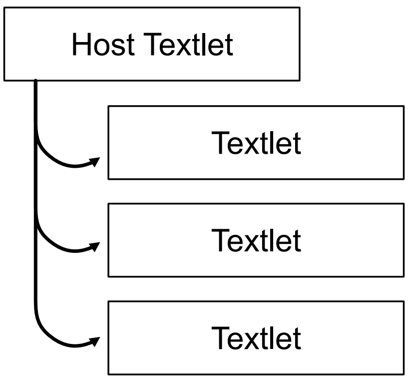
Study 1 showed that technical writers often reuse portions of text or entire templates when creating new documents. They rely on copying and pasting to incorporate parts of other documents, but this requires precisely (re)selecting the text to be copied.

With textlets, users can create text snippets specifically for reuse, such as common vocabulary and phrases, list templates, or pre-written paragraphs with placeholders. Reusing a snippet simply involves a drag-and-drop- or click-based interaction with the textlet. Placeholders can themselves be textlets to highlight the parts that need to be filled in, so that they can be easily identified, selected, and replaced with the proper text.

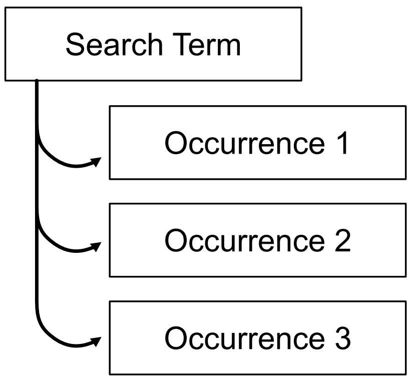
These snippets can be collected in dedicated documents or embedded into other documents. *Study 1* identified collaborative practices where users share a set of constraints and consistency criteria. By collecting reusable textlets in separate documents, they can easily share these documents and facilitate consistency across users and documents.

5.4.2 Textlets for Term Consistency

We observed that technical writers need to go back and forth on the text to make consistent changes across the document. To that



(a) Structure of a host textlet generating textlets.



(b) The structure of a *searchlet* hosting textlets for each occurrence of a word in a document.

Figure 5.2: Host textlets form a hierarchy generating and keeping references to other textlets in a document.

end, they often use the search command, but they do not trust the Find & Replace tool enough to replace all occurrences in one action carelessly. Instead, participants prefer to check the term and its context before each replace action, sometimes even typing over the occurrence instead of using the *Replace* button.

Searchlets, briefly introduced earlier, can address these use cases by automatically searching for all the occurrences of a text in the document. A *searchlet* creates *occurrence textlets* for each match of its text found in the document. In our design, these occurrences are listed under the *searchlet* in a side panel and automatically updated when the document changes. This supports fast navigation to each occurrence in the document, e.g., with a click on the occurrence in the side panel.

Searchlets support flexible Find & Replace. After specifying a replacement text for the *searchlet*, the user can replace all occurrences at once, or replace them one by one, in any order. At any time, including after replacing all occurrences, it is possible to revert individual occurrences, giving users full control and visibility over their actions. Multiple *searchlets* can be active simultaneously so that users can keep earlier searches around and get back to them later.

When users navigate the document to check for consistency and to make changes, they often lose track of where they were in the first place when they started checking. *Searchlets* facilitate navigation among occurrences, but do not address the need for location tracking in the document. Building on previous work such as *Read Wear* (Alexander et al., 2009; Hill et al., 1992) and *Footprints* (Wexelblat and Maes, 1999), a *history textlet* can record recent selections and let the user navigate among them. Previous selections can appear as individual textlets generated by the history textlet in a side panel or, to save space, the history textlet can display arrows to navigate the history of selections.

5.4.3 Textlets for Reference Consistency

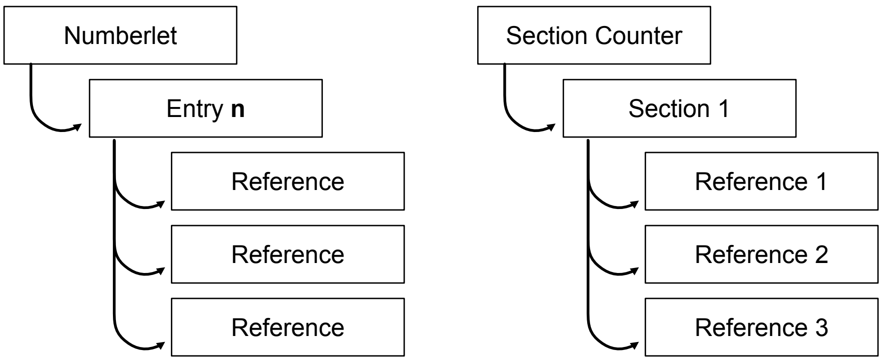
Standard word processors include tools for managing certain types of dependencies automatically, most notably numbered lists and cross-references. *Study 1* showed that participants distrust and struggle with automatically numbered lists and thus avoid automated cross-reference management tools.

Documents often include numbered items such as sections, figures, patent claims or references. Both the numbered items and the references are good candidates for textlets: Both are *computed textlets*, i.e., their content is computed and updated as the document changes, but the user can still interact with them. We propose *numberlets* as a textlet that creates numbered items and ensures that the number sequence matches the document's item order (see Figure 5.3). Each numbered item is itself a host textlet that creates and manages *reference textlets* representing references to that item. *Numberlets*, numbered items and references are listed in the side panel for easy navigation. Creating new numbered items and new references involves a simple drag-and-drop or clicking on the corresponding textlet.

This design may seem complex compared to the automatic numbering and cross-referencing features of standard word processors, but it leaves users in control by turning numbered items and references into objects that they can see and manipulate individually while the system maintains proper consistency as the document changes. It is also more powerful and flexible than the predefined types of references offered by standard word processors. For example, Word for Mac in its version number 16 can cross-reference *Headings*, *Bookmarks*, *Footnotes*, *Endnotes*, *Equations*, *Figures* and *Tables*, but not *Articles* or *Claims*, which are used extensively by contract and patent writers. *Numberlets* let users control what types of numbered items they need, providing flexibility within a unified interface.

5.4.4 Textlets for Length Constraints

Word count and character count limits are common in technical documents. For example, patent offices limit the number of claims in a patent, the number of words in the abstract, and the number of characters in the patent title. Standard word processors include tools to count words and characters in a selection, but they require users to re-select the text after every modification to verify the count. Word shows the total word count of the entire document and current selection in real time, but counting the characters in, for example, a section



(a) Structure of a *numberlet*.

(b) Example of a *numberlet* used to create a count of sections, with one section storing different cross-references to it across the document.

Figure 5.3: *Numberlets* form a hierarchy of three levels, where the top level identifies the counter, the second defines the entries that were added from that counter, and the third contains cross-references to each entry.

of the document, requires selecting the text and bringing up a modal dialog.

We propose *counting textlets* (or *countlets*) which count the number of words or characters in a segment of the document and display it in the document itself and on the side panel. As the user edits the text, the counter updates, avoiding the need for special commands or re-selection. The user can set a threshold above which the textlet will signal that the text is too long. Additional metrics could easily be included, such as the text's estimated reading time, namely, a *timelet*. Such *timelets* would be useful, e.g., for journalists and authors of video subtitles. For example, DaVinci Resolve[6] provides a visible metric of characters per second for each subtitle and lets the user define a threshold after which the value changes its color.

5.4.5 Textlets for Exploratory Writing

Study 1 showed how professional technical writers often need to manage multiple alternatives for parts of a document, before deciding or agreeing on which one to keep. Although standard word processors support change tracking, this is insufficient, since it tracks final edits and not the intermediate versions that users may want to keep but have not accepted at any point. Participants must either make copies of the entire document or use colored text or comments to list alternatives within the document.

We propose *Variant textlets* (or *variantlets*) to let users keep track of the changes made to a selection rather than the entire document. We were inspired by *Explorable Multiverse Analyses* (Dragicevic et al., 2019), where alternative analyses can be embedded in a research paper and selected by the reader to view them in context. A *variantlet* saves the original content of the selected text. After editing the text, the user can swap it with the original version for immediate comparison and swap again with the edited version. More sophisticated behaviors can be added to manage multiple alternatives, such as displaying the alternatives side by side or displaying the changes in a manner similar to the track changes mode of word processors. *Variantlets* provide greater control on version management by supporting local versioning rather than traditional document-level versioning. A similar concept is featured in *Variolite* (Kery et al., 2017) for code editing.

5.4.6 Generative Power

The previous examples show the power of textlets to support a variety of tasks. We have also identified other behaviors for textlets that could be useful for a wider range of use cases:

6 https://www.blackmagicdesign.com/

Table 5.1: How different Textlets address issues observed in *Study 1*.

Use Case	Issue	Solution
Consistent Reuse	Recurrent copy-paste to start new documents from scratch requires re-selecting the text in one or more documents.	All textlets save their text, which can be reused using simple actions such as drag-and-drop.
Term Consistency	Repeatedly navigating across a document using search terms leaves no traces of scroll positions, making it hard to go back and forth.	*Searchlets* create *occurrence textlets* that let users navigate by interacting directly with them on the side panel.
Reference Consistency	Automated numbered lists and cross-references take control away from users. Numbered items and references do not update automatically.	*Numberlets* are counters that can be manipulated and applied to numbered lists, sections, figures, etc. References to *numberlets* can be created by copy-pasting them in the document. Item numbers and references are always up to date.
Length Constraints	Standard word processors require selecting text each time to count words in a specific area and get other metrics.	*Countlets* add a persistent decoration to a text of interest that displays a word count and updates it as its contents change.
Exploratory Writing	Keeping track of alternatives is difficult. Undo/redo is not adapted to go back and forth between versions.	*Variantlets* store alternative versions of textlets that can be easily retrieved, compared and edited.

- Attaching comments, summaries, translations, word-scale graphics (Goffin et al., 2017) or emojis, and adding decorations to a textlet, e.g., highlighting or badges, to annotate the document;
- supporting arbitrary computed content, such as Victor's Reactive Documents,[7] where a textlet is defined by a formula that refers to other textlets, as in a spreadsheet;
- controlling the style and formatting of the text by associating style attributes with the textlet;
- crowdsourcing the text of a textlet or a collection of textlets for reviewing or grammar checking, as in Soylent (Bernstein et al., 2015); and

7 http://www.worrydream.com/Tangle/

- organizing textlets freely in a canvas to help analyze or annotate the content of a document (Han et al., 2022).

The generative power (Beaudouin-Lafon, 2004; Beaudouin-Lafon et al., 2021) of textlets comes from the combination of a set of behaviors:

- Navigating to the text of the textlet in the document;
- Selecting the text of the textlet, leveraging all the existing commands that act on the selection;
- Replacing/modifying text either based on user edits or automatically;
- Modifying the style of the text;
- Adding decorations that are not part of the text itself; and
- Representing and manipulating textlets in a separate view, such as a list in a side panel.

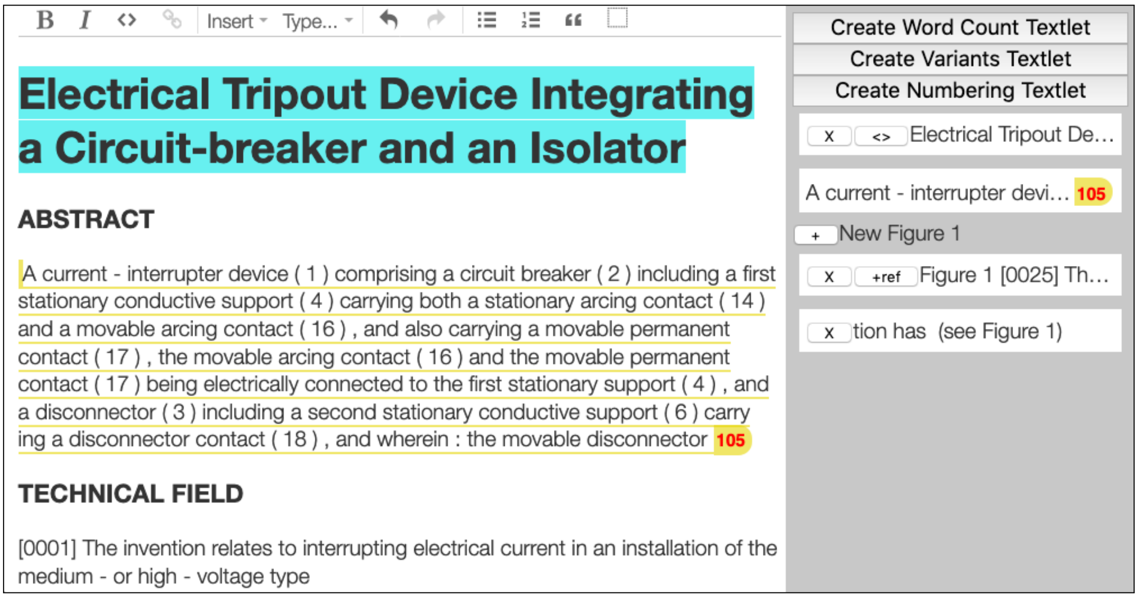
This power also comes from the ability to create textlets not only directly, by selecting text in the document, but also automatically, by using host textlets that identify and live-update a set of matching textlets. These textlets let users deal with dynamic collections of text in a concrete way, whereas standard word processors typically offer advanced, specialized commands that users hesitate to learn and use. Although textlets may involve more actions than these specialized commands, we argue that users are more likely to try them, and will save time compared to the manual solutions users resort to.

5.5 PROOF-OF-CONCEPT IMPLEMENTATION

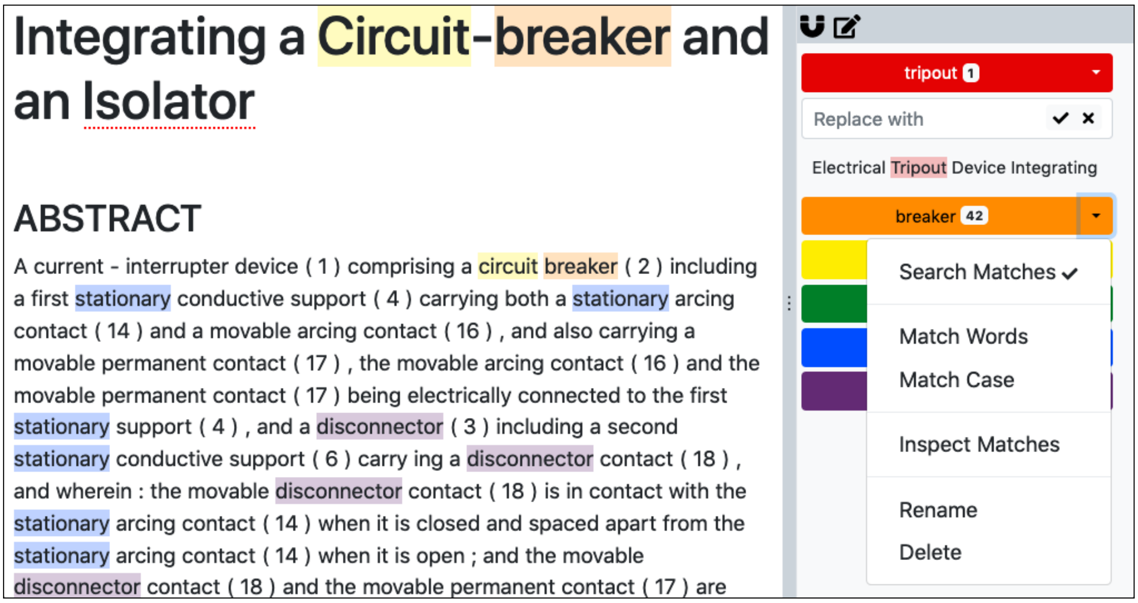
In order to demonstrate the concept of textlets, we created a proof-of-concept implementation of a text editing environment with four types of textlets: word count (*countlets*), text variants (*variantlets*), numbered references (*numberlets*), and Find & Replace (*searchlets*). These textlets address multiple use cases described in *Study 1*.

We created two prototypes as plugins to the ProseMirror[8] web-based word processing toolkit. Both were developed simultaneously. The first prototype (Figure 5.4a) was developed by Michel Beaudouin-Lafon as a test bed for *countlets*, *variantlets* and *numberlets*. I was in charge of implementing the second prototype (Figure 5.4b), which involved the development of *searchlets* through an iterative process, first internally—with Han—and then externally with testers and participants of the next study.[9]

8 http://prosemirror.net
9 A video describing how the prototype works can be found among the supplemental materials of the publication at https://dl.acm.org/doi/10.1145/3313831.3376804



(a) First prototype with the side panel showing a *variantlet*, a *countlet* and a *numberlet* containing a numbered item and a reference, and their visualization in the document.



(b) Second prototype with multiple *searchlets* that highlight corresponding occurrences in the document.

Figure 5.4: The two prototypes that were designed and implemented.

5.5.1 Overall Interface

The main window contains the text document with a traditional toolbar for basic formatting at the top and a side panel dedicated to textlets on the right. The panel features a toolbar for creating new textlets and the list of textlets themselves. It also features *searchlets*, with their list of occurrence textlets below them. A textlet is created using any of three techniques:

(a) Selecting the text content in the document and clicking a creation tool in the toolbar;

(b) clicking a creation tool in the toolbar and selecting the text content in the document; or

(c) entering a keyboard shortcut.

These techniques are also used to create host textlets (e.g., *countlets*), depending upon their type: some require a text selection, others not, and some may require additional information. Each textlet has a context menu that lets users navigate to the original text in the document, select that text, and delete the textlet. The menu also contains textlet-specific behaviors, such as *search* and *inspect* for the *searchlet*.

5.5.2 Countlets

Our implementation of *countlets* (Figure 5.5) decorates the selected text with a handle at each end. These handles let users change the scope of the textlet. The right handle also displays the word count of the text in the textlet, which is updated in real time as the user edits the content. A right-click on the *countlet* lets users set a threshold. The counter is displayed in red when its value is higher than the threshold. Deleting the textlet simply removes the word count.

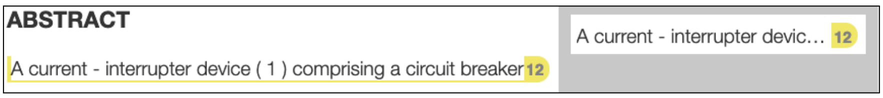


Figure 5.5: *Countlet*: a textlet for counting words.

5.5.3 Variantlets

Our implementation of *variantlets* (Figure 5.6) supports a single alternative text. When the user creates the *variantlet*, its content is stored. The user can edit the content, and swap it with the stored one by clicking a button in the side panel representation of the *variantlet*. The user can thus easily view and edit the two variants. Combining a *variantlet* with a *countlet* lets the user instantly compare the two lengths by switching between the two alternatives. A more complete implementation of the *variantlet* should include an additional button to save additional versions and a way to navigate through the versions and swap any one of them with the selection.

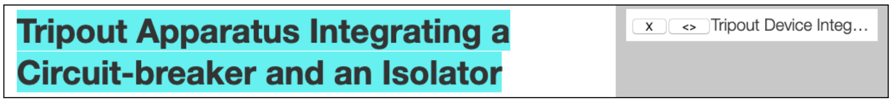


Figure 5.6: *Variantlet*: a textlet for editing local versions.

5.5.4 Numberlets

Our implementation of *numberlets* (Figure 5.7) generates a new counter with textlets to create new numbered items (*numberlets*) for that counter and new references to a given numbered item. The user creates a new counter by selecting a piece of text that contains a number or a hash sign (#), e.g., Article #. This text serves as a template for the numbering scheme. The new counter appears in the side panel as a button. Clicking this button inserts a new numbered item (the *numberlet*) at the cursor position, with the proper number. This *numberlet* is added to the side panel and is also a host textlets: clicking it inserts a reference to that item in the text at the cursor position, as well as the corresponding *reference textlet* in the side panel.

Numbered items and references are updated when the content of the document changes. The numbering of items follows their order of appearance in the document, and is therefore updated when moving text around. If a numbered item is removed and there are dangling references to it, these references show the error. All updates are immediately visible in both the text and the side panel, ensuring consistent numbering at all times. A non-implemented feature should let users drag a reference textlet below another numbered item to change the reference to that item. This would make it possible to re-attach dangling references.

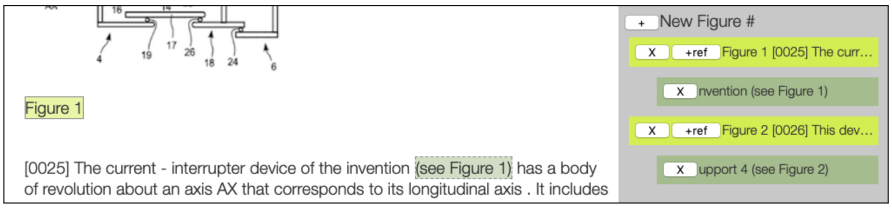


Figure 5.7: *Numberlet*: a textlet for numbering and referencing.

5.5.5 Searchlets

Our implementation of *searchlets* (Figure 5.8) supports a flexible Find & Replace task by extending the "*Search and replace instrument*" implemented by Beaudouin-Lafon (2000). A *searchlet* is created by clicking the creation tool then specifying the search text, or selecting the search text in the document and clicking the creation tool. Users can also create a blank *searchlet* and then enter the search string. Enabling the *search* behavior finds all occurrences of the search text, highlights them in the document and displays the number of occurrence in the panel. The usual "word matching" and "case sensitive" options become available in the menu to refine the search (Figure 5.4b).

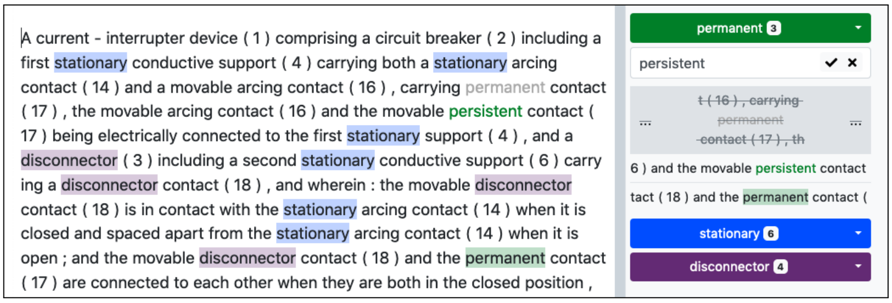


Figure 5.8: *Searchlet*: a textlet for searching and replacing text.

5.5.5.1 *Navigating Occurrences*

Enabling the *inspector* behavior generates the list of occurrences below the *searchlet* in the panel, highlights them in the document and gives access to the replace capability (Figure 5.8). Changing the search string or the search settings re-runs the search and updates the list of occurrences underneath it. Editing the document also dynamically updates the list of occurrences: typing the searched text in the document creates a new occurrence and changing the text of an occurrence in the document removes it from the list of textlets if it does not match anymore.

Each occurrence is a textlet that displays the text surrounding the match in the document and updates it in real time. An occurrence can be expanded by clicking it to better show the context (Figure 5.8). The user can then click the ellipsis buttons to show more context.

Occurrences can be moved, including under another *searchlet*, giving users flexibility to organize the search results as they see fit. For example, occurrences of different misspellings of a word can be identified with different *searchlets* and then grouped under one *searchlet*, after which they can all be replaced at once. When moved, occurrences adopt the color of their new host *searchlet*. They also "belong" to their new host for the purposes of the *Replace All* action. In the current implementation, they disappear from the list when the search string or the search settings of the new host are changed.

5.5.5.2 Replacing Text

Selecting *Replace Matches* in the *searchlet* context menu (Figure 5.4b) shows a text input field for typing a replace string and a button for replacing all occurrences in the list. Each occurrence textlet also includes three buttons that replace only that occurrence, revert to the previous text, or ignore this occurrence from future *Replace All* operations. These actions can also be performed in the document itself using keyboard shortcuts.

Replaced occurrences stay in the *searchlet*'s occurrence list until its search string is updated. This lets users work with the occurrences and make changes to the document after they perform a replace operation without losing track of the positions that were originally matched.

Searchlets extend Beaudouin-Lafon's previous work (Beaudouin-Lafon, 2000) by supporting multiple simultaneous searches. Each occurrence is reified as an item in the side panel, which supports additional functions such as disabling an occurrence in a global replace action, or moving an occurrence to another *searchlet*. Our design is also grounded in our observations of real-world challenges experienced by the professionals in *Study 1*.

5.6 STUDY 2: FIND & REPLACE TEXTLET

We used our second prototype to evaluate *searchlets* with an observational study. We did not run a comparative study with, e.g., Microsoft Word as a baseline because many features that we implemented do not exist in Word or are clearly faster, e.g., a persistently-displayed character count with *countlets* vs. highlighting text and invoking Word's word count command.

Our goals were to gather feedback, identify potential novel and unexpected uses and discuss new ideas with the participants in order to refine our design. The study focused on *searchlets*, but we also showed the participants the other textlets from the first prototype. We incorporated suggestions incrementally so that successive participants used slightly different versions of the prototype.

5.6.0.1 Participants

We recruited eight participants: three Patent Attorneys, one Patent Inventor (1 female, 3 male; aged between 29 and 50 years old, who use various versions of Microsoft Word) and 4 researchers (1 female, 3 male; aged between 24 and 26 years old who use LaTeX). Three of the patent attorneys had participated in *Study 1*. We included researchers because we believe that textlets address the needs of a wider range of users than those in *Study 1* and authors of research articles must also manage consistency in their papers.

5.6.0.2 Setup

The prototype was a web application (*Textlets*) accessed with a browser of the participant's choice on their own computer if possible. We provided a 13" MacBook Pro laptop running macOS 10.14 and Firefox 68.0 for participants who did not have a computer at hand. We created two sets of documents to match the participant's background: two patents and two research papers.

5.6.0.3 Procedure

All sessions were carried out by Han and I in English. We started by describing the features of the *Textlets* prototype and gave participants 10 minutes to experiment with it. We used a think-aloud protocol and asked participants to perform two similar tasks on two documents: one using the editor of their choice and the other using the *Textlets* prototype. We counterbalanced for document order across participants.

Each task consisted of three small exercises with increasing difficulty:

1. replace a word by another and then change it back;
2. replace a word by another but only in certain contexts; and
3. replace two words with similar meanings with another word, including all relevant variations. Thus replacing "mouse" with "rodent" also requires changing "mice" with "rodents."

The two tasks—each with three exercises—took approximately 20 minutes. After an interview, participants completed a short questionnaire. The session ended with a debriefing to identify additional use cases and discuss ideas for improvement. We also showed participants *countlets* and *variantlets* from the first prototype and asked them if they could describe scenarios for which these might be useful.

5.6.0.4 Data Collection

We recorded audio, took hand-written notes during the session and collected the answers to the questionnaire.

5.6.1 Results & Discussion

All participants successfully interacted with the *Textlets* prototype and found the tasks representative of their everyday work. Participants found that the textlets side panel was "*faster to use*" (**P1, 3**). Participants preferred because it avoids jumping to the main text (**P1, 2, 3, 6**), so that they can focus on the relevant document parts, thus reducing mental workload. Most participants (6/8) preferred making changes directly with a *searchlet* over Word's non-interactive side panel. Two participants (**P1, 2**) asked for even greater interactivity with *searchlets*, such as one-by-one replacement directly from the panel—which we added in a later version—, and merging two *searchlets* to apply the same replacement to their occurrences. We added other small improvements based on participants' feedback—including better colors and icons— and de-cluttering the textlets' widget by using a menu for its actions instead of a series of buttons.

5.6.1.1 Replace-Then-Correct Strategy

Most participants (7/8) used a *search-check-replace* strategy for each occurrence found with both Microsoft Word and LaTeX: they search for the word, go to each occurrence in the main document to check the context and then perform the replace action, either by clicking a button or retyping on the document.

Participants used a different strategy with textlets, which we denominate *search-overview-replace*. They started by creating one or more *searchlets*, scanned the overview of the occurrences to see the variations and assessed which ones to replace. P1 said: "*I can see immediately what variations are in the text [from the side panel]. So I see it will work by replacing all matches.*"

The combination of overview and contextual information around each match encouraged participants to spontaneously develop two different strategies for the final search-overview-replace step: Six participants used a replace-all-then-correct strategy, first replacing all occurrences, then checking each replaced occurrence in the overview list for errors, which they corrected either with the *Revert* button or by retyping in the document. The other two participants (**P6, 7**) used an *ignore-replace-all* strategy, first pressing the *Ignore* button to skip outliers, then using the *Replace All* action, similar to the "perfect selection" strategy by Miller and A. M. Marshall (2004). In summary, although participants were reluctant to replace all occurrences with their regular word processor, they felt comfortable using the *searchlets*' *Replace All* action and quickly developed strategies for selective replacement.

5.6.1.2 Persistent Selection: Keep Track & Individual Undo

Although both Microsoft Word and TexWorks (LaTeX editor) provide an overview list of all search occurrences, they do not track them by position. By contrast, *searchlets* create persistent occurrences that help users keep track of what happened. **P5** felt confident with the prototype, saying: "*Here (pointing at the side panel) I can see the changes in context. It helps me [and] reassures me that I did the right thing.*"

Furthermore, *searchlets* let users check the results of their previous replacements. The overview of occurrences persists in the panel even as users edit the document. This differs from other word processors that clear the search results whenever the user types in the document, which forces users to tediously re-enter the search text. For example, P3 said: *"I have this list of all the occurrences. When I want to do some replacements, I choose some of them and I keep the whole list that I can always check [in the side panel]. This is quite important... I do not need to proof-read the whole text."*

Because each occurrence is also a textlet with its own history of changes, it can be undone individually and ignored in a *Replace All* command while still remaining in the overview list. These novel behaviors contributed to most participants (6/8) spontaneously adopting a strategy where they replaced all occurrences an then corrected those that needed it. For example, **P3** said while performing a task: *"Maybe it is better to replace all and check the ones that do not work."* This suggests that making changes persistent, visible and reversible increases users' trust in the system.

5.6.1.3 Re-purposing a Searchlet

One participant suggested embedding a group of *searchlets* as *"a highlight [feature] for forbidden words"* (**P4**), arguing that making co-authors aware of these words as the document circulates would help them maintain consistency and improve collaborative editing. Textlets could thus embody constraints and serve as an active guideline when embedded in a document. This shows that after the participant acquired knowledge of the interaction of a *searchlet* with the text, she thought of a use deviating from what we had described about the design or what was supported in the prototype. In other words, the use of a *searchlet* was extended to become a permanent attachment to the document, rather than existing as a separate tool that is used on demand, thereby re-purposing the *searchlet*. While we did not assess her thought process, this arguably required her to understand the *principle* governing the relationship between the *searchlet* and the text, i.e., when used, it highlights occurrences of its selection, so that she could transfer the behavior to an analog situation where the textlet is persisted with the document. Therefore, I argue that this finding resonates with the notion of technical reasoning applied to a digital environment, supporting the unusual use of a novel digital tool.

5.6.1.4 Feedback of Countlets and Variantlets

Participants also described situations in which they wanted to use *countlets* and *variantlets*. For example, **P3** wanted to count the words in patent abstracts: *"I think this could be very useful because many times you are going to count words and [the system] does not keep it."* **P4** wanted to use *variantlets* as a local versioning tool: "*If you can version one paragraph [instead] of the whole document, it could be very useful. In that case, you can track which part you have changed.*"

5.6.2 Scalability and Limitations

A potential limitation of our approach is scalability: *Searchlets* that generate large numbers of matches (i.e., large numbers of textlets) in the side panel could cause problems when dealing with large documents. We did not observe such problems during the study, probably due to its short-term nature. Several features mitigate scalability issues: users can collapse host textlets, e.g., search results, to save space or disable them to remove highlighting in the main text. Scrolling between the document and the side panel could also be synchronized, and future textlets could combine behaviors in one widget to save space, e.g., *countlet* + *variantlet*.

P3 found that *searchlets* might be less useful in simple cases with few matches or variations of the same word: "*[With] only 3 matches, I would like to change it directly in the main text.*" Another participant wanted *searchlets* to support regular expressions. Both features could easily be supported in a future prototype.

5.6.3 Summary of the Discussion

This study showed the value of *searchlets*, the most complex textlet we developed, as well as the potential of other textlets. By turning search matches into persistent objects that users can manipulate directly, participants were willing to use functions, such as *Replace All*, that they would otherwise avoid with traditional word processors. They also spontaneously devised novel strategies to re-purpose the textlet concept in unexpected ways, such as embedding *searchlets* for forbidden words. This study provides evidence for the validity of the textlet concept and encourages to further develop and assess the textlets we have developed, as well as to design new ones.

5.7 SUMMARY

Writing technical documents frequently requires following constraints and keeping consistency using domain-specific terms. In this chapter, I presented our interviews with 12 legal professionals showing that technical writers are reluctant to use advanced features of their word processors and must instead rely on their memory to manage dependencies and maintain consistent vocabulary within their documents. As a result, we introduced a simple but powerful concept called *Textlets* to refer to interactive objects that reify text selections into persistent items.

Textlets are a deceptively simple but powerful idea, based on the premise that creating persistent, interactive objects to represent abstract or transient concepts such as the selection, can empower users. This work showed five use cases where textlets can be applied to support consistent reuse, term and reference consistency, word count constraint, and exploratory writing. It described two prototypes that implement a proof-of-concept of four textlets and used one as part of an observational study to assess a search-and-replace textlet. All participants successfully used the prototype to perform advanced tasks and most spontaneously generated a novel *replace-all-then-correct* strategy. Several also suggested novel uses and ideas for new textlets, among which, one involving re-purposing *searchlets* as attachments to documents that highlight forbidden words. This suggests that users can quickly pick up on the effect of textlets over text objects, thereby leveraging their ability to perform technical reasoning. Beyond the examples illustrated in this work, textlets offer a generative concept for creating powerful new tools for document editing. Furthermore, they draw on the "selectable" property of objects to define their effects, thereby demonstrating the increased power of a property-oriented digital tool.

6 CONCLUSIONS

"We shape our tools, and thereafter our tools shape us."

—Marshal McLuhan

Tool use is an inherent ability of humans, who frequently engage with objects to use them as mediators of their actions within physical environments (Osiurak et al., 2010). The Theory of Affordances (J. J. Gibson, 1986) posits that animals *can* perceive the possibilities for action offered by objects to them, according to their body capabilities, and in a *direct* way, i.e., without the need to invoke memories of past experience or cultural background. Although previous studies support the notion of direct perception of affordances for how humans engage with tools (Buxbaum and Kalénine, 2010; Osiurak and Badets, 2016), our ability to understand how large objects interact with each other without being able to manipulate them (Osiurak et al., 2010), e.g., a plane crushing a car, as well as the use of modern technology for which there is no clear association between handling and function (Kaptelinin and B. Nardi, 2012), e.g., the invisible connection between the trigger and the drill part in an electrical drill, cannot be readily explained by the Theory of Affordances. The Technical Reasoning hypothesis (Osiurak et al., 2009) puts the focus back on the *interactions* between objects during tool use, rather than on their manipulation. Under the technical reasoning model, humans simulate interactions between objects based on abstractions of their properties, which are matched against knowledge of technical principles, powering the use of these objects as tools upon other target objects. In consequence, technical reasoning models our understanding of tool use centered on the tools' relationships with the target objects, rather than the direct perception of affordances relative to the user's body capabilities.

In digital environments, Bødker (1987) argues that applications are not the target object of users' work, but are rather "computer-based artifacts" through which they act on objects, i.e., "users act through the interface" (Bødker, 1987) rather than focus on it. Bødker's analysis is based on Activity Theory (Leontiev, 1978) situating the user interface as a mediator of human work. Based on this approach, one can infer that digital and physical tools are equivalent psychological entities, carrying meaning as mediating means, i.e., the meaning given by the user as to what the tool is for. As an approach to tool use, Activity

Theory offers a perspective that is quite different from the direct perception of affordances of a tool, incorporating culture in the way that an object is used, e.g., one can see that a knife can be used as a screwdriver yet recognize at the same time that such use is not its function. This forms one of the arguments for extending the notion of affordances as introduced in HCI (Gaver, 1991; Norman, 1988)—and its original concept (J. J. Gibson, 1986)— to account for the users' socio-cultural background (Kaptelinin and B. Nardi, 2012) for how they interact with technology, i.e., how they choose to use digital tools for given tasks. Conversely, certain perceived affordances of tools will influence deviations from culture leading users to including them in new activities, e.g., having a regular pair of scissors at home used to slice pizzas even though we know of specific tools for that purpose. This is put forward as the developmental process behind instrumental genesis (Béguin and Rabardel, 2000), which situates re-purposing at the center of the evolution of tools, leading for example to the development of a pair of scissors with added modifications for slicing pizzas.

Based on the empirical observation that users do manage to re-purpose digital tools, this thesis addressed the question of a cognitive model of human tool use that is compatible with the use of digital tools as well as their re-purposing. In this sense, the Technical Reasoning hypothesis provides a model for the re-purposing of physical tools that bridges the socio-cultural approach of instrumental genesis with the cognitive approach to human tool use, based on knowledge about object properties acquired from experience. To that end, I focused on the relevance of the Technical Reasoning hypothesis for the re-purposing of digital tools. The results I presented suggest that users manage to use and re-purpose familiar and novel digital tools by reasoning about their effects on target digital objects, rather than remaining stuck using learned procedures meant for conventional tasks. Technical reasoning then stands as a candidate for a cognitive framework in HCI, supporting the design of user interfaces that let users use digital tools in unusual ways, towards their appropriation in the long term, such as it happens with physical tools. I do not mean to suggest that re-purposing should be the main goal of the design of a user interface, but rather that it offers a path to create interfaces that embrace the human ability to perform technical reasoning, giving users more power in the way to approach tasks. In this chapter, I close my work by summarizing its contributions, discussing implications and limitations, and laying out avenues for future work.

6.1 CONTRIBUTIONS

This thesis brings both theoretical and methodological contributions. First, I bring in the Technical Reasoning hypothesis (Osiurak et al., 2009) as a cognitive framework to re-think the way in which users re-purpose digital tools. The relevance of this approach to human tool use is reflected in the discussion by Osiurak and Badets (2016), who contrast it with other prominent cognitive theories around this question, notably, the Theory of Affordances (J. J. Gibson, 1986), frequently cited in HCI research. Additionally, the Activity theory approach to HCI offered by Bødker (1987) puts digital tools at the center of users' activity with computers, establishing their psychological proximity with physical tools, and setting the basis for the first two studies presented in this manuscript.

In the first study (Chapter 3), I presented work investigating the relevance of the Technical Reasoning hypothesis to model the re-purposing of digital tools. It includes the design of an experimental protocol based on a custom text editing environment, aiming at forcing users to re-purpose one or more commands in order to complete a text layout task. As a result, most participants managed to carry out at least one solution that involved re-purposing a command. Additionally, a thematic analysis of the participants' verbal protocols suggests that some of them engaged in a process compatible with technical reasoning, based on their knowledge of the mechanics involved in text editing and how it could be applied to devise a solution. While both the participant's self-reported experience with text editing and creative personality scores associated with finding more alternative solutions to the task, only creativity associated with carrying out a re-purposing-based solution. This is compatible with previous findings of individuals experiencing functional fixedness (Duncker and Lees, 1945), expressed in the difficulty to come up with unusual uses for familiar commands.

In the second study (Chapter 4), I presented work focusing on the priming effect that toolbars and interactions with digital environments have on the users' knowledge of what is possible with its objects. It includes the design of a digital environment whose objects support both graphic- and text-based commands, each accessible through toolbars organized in a way similar to popular commercial graphic- and text-editing software. This environment is used in an experimental protocol where participants are introduced to it with either only graphic-based or only text-based commands. Participants then select the objects according to the type of toolbar that they see (e.g., text selection when observing a text-oriented toolbar). In the last stage, participants complete several tasks with both toolbars available, i.e., with access to the commands of both types. The results of this study show that participants solved the tasks using significantly more commands of the type with which they were primed. These suggest that the environment cues that we provided (i.e., toolbars and selection feedback) primed their knowledge of what was possible with the digital objects, therefore, leading them to use only some types of "mechanical" knowledge to interact with the content.

The last portion of this work focused on a novel digital tool for text editing, stemming from a study of extreme users of word processors (Chapter 5). The results of the interview study suggest that professional document writers develop practices for keeping consistency and following constraints that involve keeping a tight control over the content, seldom handing it over to automated features. In particular, participants of the study showed that they frequently re-use—in whole or in part—previous documents as the basis for new ones, thus engaging in modifications that are subject to the same consistency requirements as the original document, e.g., modifying a patent submission's summary. These findings formed the basis for creating the concept of a *textlet*: a reification (Beaudouin-Lafon, 2000) of text selection into a persistent object that supports different behaviors with respect to the document. Different examples of textlets are presented according to different behaviors that offer support to some of the constraints and consistency requirements found during the interview study. Additionally, some of these textlets were implemented in a prototype, notably, a Find & Replace textlet that was later used for a short evaluation. Participants of this evaluation generally preferred it over the traditional Find & Replace function included in their word processors. Of particular relevance to this thesis is the case of one participant who devised an unusual use for the Find & Replace textlet as a tool for flagging forbidden words, thus contributing an additional example of users re-purposing a novel digital tool designed according to the principles of Instrumental Interaction (Beaudouin-Lafon, 2000).

6.2 IMPLICATIONS

The results of the studies presented in this thesis support the idea that users accumulate knowledge about their interactions in digital environments, building notions that they manage to transfer to other digital environments (Gilbert and Cordey-Hayes, 1996). The Technical Reasoning hypothesis presents a compelling cognitive model of a mechanism through which users can make sense of the possibilities for interaction among objects. The evidence of users performing technical reasoning in digital environments supports the notion of knowledge of "digital principles" compatible with a form of "mechanical" knowledge of the digital world. Furthermore, the technical reasoning model allows interaction designers to capitalize on "real-world" cognitive abilities (Jacob et al., 2008) that underlie our understanding of interactions among physical objects. As such, the Technical Reasoning hypothesis offers the theoretical foundation for interaction models based on objects and tools whose interactions produce consistent results that users can abstract, transfer and effectively apply to other digital tool-object interactions, based on past interactions.

Technical reasoning supports moving away from the application-centric paradigm, which encloses digital objects in workflows and tool sets that are not necessarily the users' choice (Beaudouin-Lafon, 2017; Nouwens and Klokmose, 2018). Under this paradigm, the appropriation of digital tools is hindered by blocking the users' ability to bring their learned practices to other applications. This could be partly addressed by designing digital tools that have less friction with the objects with which they can interact, so as to let users abstract out the effects of these tools rather than memorize the objects with which they can interact. For example, we could design digital tools to have effects on one or more *properties* of an object, rather than based on the object *type*, thus allowing users to use the tool on any object that carried such properties (Beaudouin-Lafon et al., 2021). On the same note, the notion of a "mechanical" knowledge of the digital world makes it all the more important to convey the "nature" of digital objects to users, so as to let them devise their own creative uses of digital tools.

Technical reasoning complements existing theoretical work in HCI. Its presentation as a cognitive model that builds upon the user's experience with the environment offers an alternative to the direct perception of affordances (J. J. Gibson, 1986) by way of reasoning. This reasoning is the product of learned uses of objects and tools which compose the abstractions that form the users' mechanical knowledge. On the side of Activity Theory approaches to HCI (Bødker, 1987; Kaptelinin and B. Nardi, 2012), technical reasoning offers a cognitive model enabling instrumental genesis processes (Béguin and Rabardel, 2000), grounded in the ability to re-purpose objects. Consequently, the Technical Reasoning hypothesis offers theoretical support for existing interaction models that embrace the mediated action perspective (Kaptelinin and B. Nardi, 2012), such as Instrumental Interaction (Beaudouin-Lafon, 2000). As such, the Technical Reasoning hypothesis adds to the pool of grounding theories that can be used to form generative theories of interaction in HCI (Beaudouin-Lafon et al., 2021).

6.2.1 Technical Reasoning in HCI

The technical reasoning model contributes new grounds for studying how users discover and make sense of their possibilities for interaction in digital environments. The ability to capture and transfer abstract principles of interactions among objects offers support for the development of interfaces that account for the users' "mechanical" knowledge of the digital world. One can point at such knowledge today when, for example, users have incorporated principles from interactions with WIMP interfaces, such as double-click to open or drag and drop to move. Furthermore, a "mechanical" knowledge of the digital world offers support for the larger notion of "unified principles of interaction" (Beaudouin-Lafon, 2017). The vision behind this project is to consolidate the design of interactions to make them consistent across digital environments, for instance, developing principles that can be learned in a desktop environment and applied in a mobile environment, e.g., double-tap on the phone, and double-click on the computer to select a word in a sentence. Caring for such consistency details could help consolidate the user experience in the digital world.

In regard to the perception of affordances (J. J. Gibson, 1986; Norman, 1988, 2013), technical reasoning provides a perspective that separates the perception of affordances offered by tools to users, from those offered by tools to act on other objects. Technical reasoning then models the latter, describing how humans make sense of the possibilities for action in terms of the technical principles that drive such actions, i.e., which tool-user interaction causes the appropriate tool-object interaction. The technical principle defines the effects of the tool when used on an object, e.g., reshaping, changing color, etc., which are the concern of technical reasoning, while the manipulation of the tool still requires the perception of its affordances to succeed. In a digital environment, the manipulation of tools is limited to certain degrees of freedom offered by input devices and the limitations built into the digital tool's design. In this sense, the *signifiers* (Norman, 2013) offered by digital tools such as clicking, tapping, pinching, etc., can be seen as the "manipulations" that users carry out on digital tools and objects, which translate into particular technical principles in the digital world, e.g., click to select with pointer, drag to erase with eraser, double-click to select word with i-beam, etc.

The mediated-action perspective (Kaptelinin and B. Nardi, 2012; Kaptelinin and B. A. Nardi, 1997) is enhanced by the Technical Reasoning hypothesis because it looks at the way in which humans use mediating means as more than the direct perception of affordances. In this regard, the technical reasoning model provides a generative angle to the Activity Theory approach to HCI, describing how a mediating mean becomes an instrument, on the basis of its properties making them able to interact with other objects. Furthermore, its modeling of tool re-purposing complements the developmental process around deviant uses for the evolution of tools (Béguin and Rabardel, 2000), i.e., the instrumentation and instrumentalization processes.

Finally, technical reasoning brings new elements to the discussion around the change in perception offered by tool use based on the embodied cognition perspective (Kirsh, 2013). In line with Osiurak and Badets (2016), Kirsh (2013) points at the lack of discussion around the "relation between tool and affordance," which refers to the process of understanding how a tool can cause an effect on a given object. The technical reasoning model describes this as a reasoning process based on mechanical knowledge. In this sense, mechanical knowledge could be interpreted as the experience that changes our perceptual learning (E. J. Gibson, 1963), expanding our action repertoire in the environment through the perception of a tool's properties. In turn, mechanical knowledge resonates with Kirsh's idea of skill being a factor beyond the embodiment occurring when holding a tool. After all, technical reasoning describes a process of abstraction of principles and transfer, regardless of the particular shapes or acquired meaning (Leontiev, 1978) of the tools and objects involved, leading to, for example, interpreting that one can shave off layers of a wood beam through the same principle as peeling a carrot, i.e., two situations that involve completely different objects. In sum, the Technical Reasoning hypothesis shows promising features as a complementary tool for contemporary theories in HCI.

6.2.2 What Technical Reasoning Brings to Interaction Design

For designers, the straightforward implication of this thesis is that users could benefit from having digital tools that apply principles consistently over the digital objects of an environment. For example, observing that a *resize tool* allows to resize any objects with a *size* property could lead users who have acquired such knowledge of the tool to seamlessly apply this principle with novel digital objects in a novel environment, provided that they have access to such tool, e.g., resize an event in a calendar, resize the selected text, etc. This is not to say that complex and black-box approaches followed by existing digital tools should be replaced by techniques involving simpler digital tools requiring more steps to be used. Arguably, users of all levels of expertise would benefit from made-for-purpose digital tools that resolve complex tasks. My argument is that some users will also benefit from coming up with their own ways to carry out tasks, even when they may not be the most efficient, because these uses will consolidate principles that will come handy in situations for which a made-for-purpose tool has not been implemented. In the same sense, users being able to re-purpose and appropriate digital tools will inform the design of future complex digital tools, following the logic behind the developmental approach to the evolution of instruments (Béguin and Rabardel, 2000). In conclusion, digital tools should be explicit about the properties that they affect as well as digital objects should be explicit about the properties that they expose for modification, so that users are able to incorporate and transfer the principles that they learn from repeated interactions with them.

6.3 LIMITATIONS & FUTURE WORK

Several limitations and avenues for future work arise from this work. First, the experimental setup used for the study in Chapter 3 represents an artificial environment with the intention to force participants to re-purpose its commands. While similar situations can be encountered with real-world digital tools, e.g., when editing with a text editor that lacks a ruler tool, it would be interesting to conduct studies of digital tool re-purposing in a more ecologically-valid context, including longitudinal studies focusing on the type of spontaneous re-purposing observed in Chapter 5. Second, we highlighted the correlations between the performance of participants in Chapter 3 and their creativity, the latter limited to self-reported scores based on a standardized questionnaire. Although it lied outside of the scope of the original study, further research should inspect other measures of creativity, as well as other personality traits that could associate with creative uses of tools. In the same direction, other factors could be investigated as potentially affecting the ability to perform technical reasoning in digital environments, along the lines of what is presented in Chapter 4.

For Chapter 4, while we observed that the overall users' choice of commands associated with the toolbar that they first encountered, this is only reproduced for two tasks analyzed individually, both involving repetitive actions such as moving and copying multiple objects. A new version of the experiment could focus on a greater number of repetitive tasks so as to understand the extent to which the complexity of the task induces participants to focus on it rather than on exploring the interface for alternatives. Additionally, we developed an environment limited to graphic- and text-based commands because of their widespread familiarity. Other digital affordances should be considered to be integrated and combined into single digital objects, in order to allow for more commands to apply, e.g., by integrating spreadsheet cell behavior, thereby gathering more evidence in favor of a digital "mechanical" knowledge at play.

On the whole, I see the potential in this work by its *formative* role for design (Rogers, 2004), emphasizing the computer-mediated activity perspective (Kaptelinin, 1996) in generative models such as Instrumental Interaction (Beaudouin-Lafon, 2000; Beaudouin-Lafon et al., 2021), and establishing guidelines for the design of interfaces that leverage technical reasoning. For instance, such a model could incorporate and define a "mechanical" knowledge of digital environments so as to build the notions off of which its digital tools would resolve interactions with digital objects. Ultimately, this work represents but a first step towards investigating the technical principles of the digital world.

APPENDICES

a EXPLORING TECHNICAL REASONING IN DIGITAL ENVIRONMENTS

A.1 QUESTIONNAIRES

Table a.1: Questionnaire before the session begins, assessing the experience with text editing software.

Text	Type	Options
I edit and format text documents on the computer...	5-point Likert	*Very Rarely* to *Very Often*
When I get stuck using the computer I...	5-point Likert	*Give up right away* to *Try until I find the solution*
	5-point Likert	*Look up the web or ask for help right away* to *Try until I solve it on my own*
I see my knowledge about text editing with computers as...	5-point Likert	*Basic* to *Expert*
The number of 'hacks' or 'tricks' I know to get things done with my text editor(s) is...	5-point Likert	*Very Small* to *Very Large*

Table a.2: Questionnaire after every trial ends.

Text	Type	Options
How well did you do?	5-point Likert	*1* to *5*
Have you achieved a similar result in the same way before?	Single Choice	*Yes* or *No*
Did you use a method that you saw somewhere else but hadn't used yourself yet?	Single Choice	*Yes* or *No*
Did you devise a method to use before the trial started?	Single Choice	*Yes* or *No*
Did you look up the interface for something that could help you solve it?	Single Choice	*Yes* or *No*
Did you try random things on the interface?	Single Choice	*Yes* or *No*

Table a.3: IPIP items for creativity assessment. Participants answer about their level of agreement with each statement. Items are keyed (+) or (-) to indicate whether they count for or against creative personalities. Items were presented in randomized order for every participant.

Text	**Type**	**Options**
I like to solve complex problems (+)	5-point Likert	1 to 5
I love to read challenging material (+)	5-point Likert	1 to 5
I love to think up new ways of doing things (+)	5-point Likert	1 to 5
I have a vivid imagination (+)	5-point Likert	1 to 5
I know how things work (+)	5-point Likert	1 to 5
I am not interested in abstract ideas (-)	5-point Likert	1 to 5
I am not interested in theoretical discussions (-)	5-point Likert	1 to 5
I avoid difficult reading material (-)	5-point Likert	1 to 5
I try to avoid complex people (-)	5-point Likert	1 to 5
I do not have a good imagination (-)	5-point Likert	1 to 5

A.2 ACTION LOGS

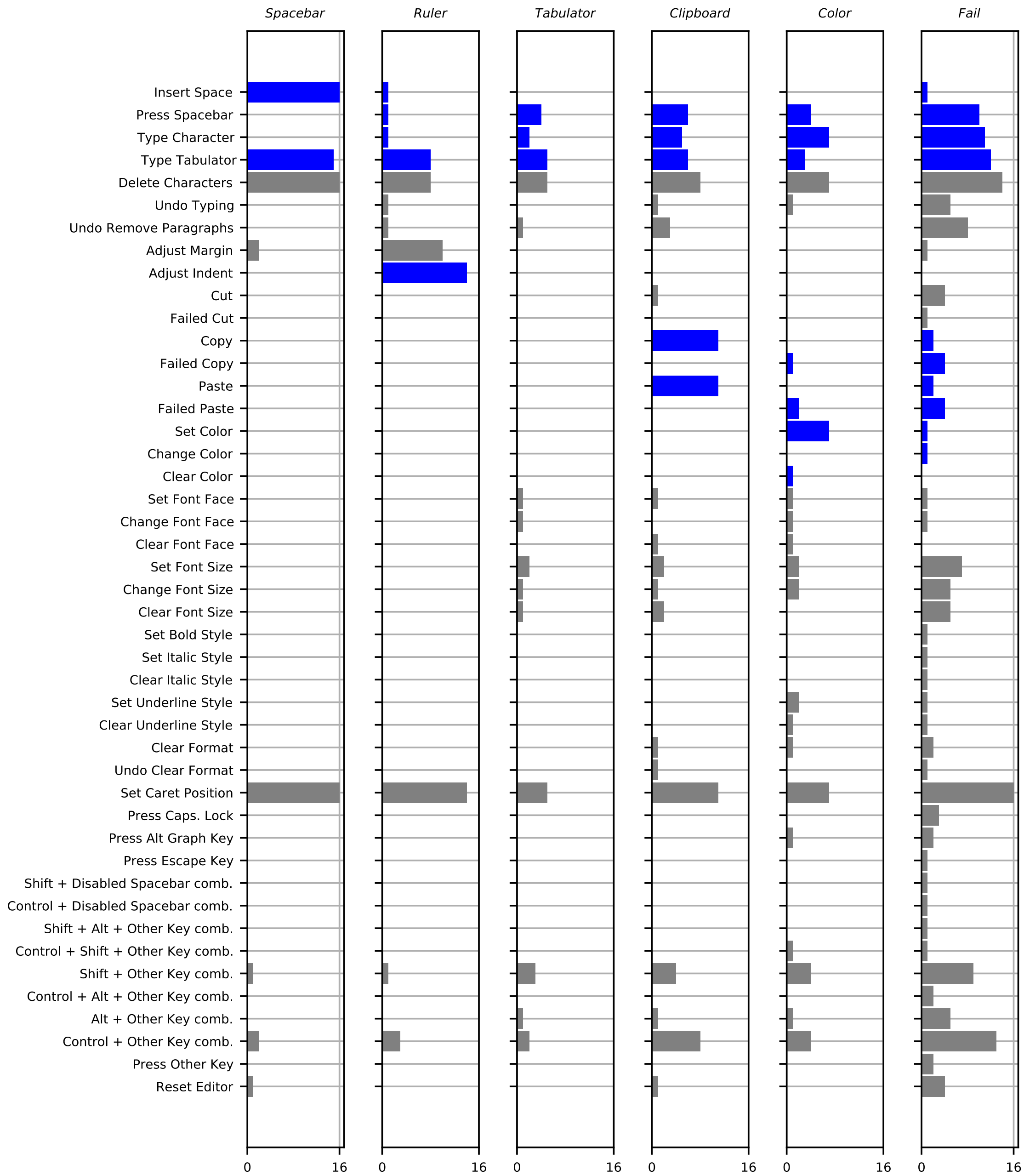


Figure a.1: Number of participants registered producing the event on the left using the editor, grouped by *TECHNIQUE* and including the last trial. Includes all the command types counted in #*TYPES* and other key combinations and cursor updates. Primary events are highlighted to indicate their association with re-purposing techniques.

Table a.4: List of actions captured for each participant.

Action	**Description**
Insert Space	Insert a space character using the spacebar
Press Spacebar	Press the spacebar while it is disabled
Type Character	Type any character key
Type Tabulator	Type a tabulator character
Delete Characters	Delete or backspace on characters
Undo Typing	Undo typing action
Undo Remove Paragraphs	Undo actions removing text or entire paragraphs
Adjust Margin	Adjust the ruler's paragraph margin
Adjust Indent	Adjust the ruler's paragraph indent
Cut	Cut selection from the document
Failed Cut	Attempt to cut selection while clipboard commands are disabled
Copy	Copy selection from the document
Failed Copy	Attempt to copy selection while clipboard commands are disabled
Paste	Paste text in the document
Failed Paste	Attempt to paste text in the document while clipboard commands are disabled
Set <property>	Set the selection's <property>
Change <property>	Change the selection's <property>
Clear <property>	Revert the selection's <property> value to its default
Clear Format	Clear the current selection's format properties
Undo Clear Format	Recover the format properties applied to a text before they were cleared
Set Caret Position	Put the text caret at a new position
Press <key>	Press the non-character key <key> (non-standard keys are identified as "Other")
[Shift | Control | Alt] + <key>	Key combination that does not produce modifications to the document as a result
Reset Editor	Resets the environment to the trial's initial state

A.3 STUDY RESULTS

Table a.5: Quantitative measures of our study with 16 participants. **R**, **S**, **T**, **C** and **X** stand for `Ruler`, `Spacebar`, `Tabulator`, `Clipboard` and `Color` respectively. **Trial #** cells indicate the trial number at which the technique was performed by the participant.

			Trial #					
P#	***EXPERIENCE***	***CREATIVITY***	**R**	**S**	**T**	**C**	**X**	***#SOLUTIONS***
1	20	40	1	2		3	4	4
2	19	34	2	1				2
3	20	43	1	2	3	4		4
4	9	34	2	1				2
5	25	37	1	2	4	3		4
7	19	37	1	2	3			3
9	16	35		1				1
10	21	44	1	2		3	4	4
11	8	39	2	1				2
12	17	43		1		2	3	3
13	16	42	2	1		5[a]	3	4
14	20	37	1	2		3		3
15	19	45	1	2	3	4	5	5
16	20	40	1	2	4	3	5	5
17	11	38	1	2		3		3
18	17	42	1	2		3	4	4
			Median Trial #					
			1	2	3	3	4	

a Trial 4 was discarded because the participant used a small font to make characters invisible (which was not supposed to work by design) and stated that he could not see them on his screen, likely due to low resolution. We accepted the technique as valid to continue the session but did not include this trial in the analysis.

b EXPLORING CUES FOR MECHANICAL KNOWLEDGE

B.1 AGGREGATED DISTRIBUTION OF COMMANDS FOR DIFFERENT MIXED APPROACH THRESHOLDS

The plots in Figure b.1 show the percentage of graphic-based commands used by participants in a given task. These allow to visualize the change in the percentage of text- and graphic-based commands between participants according to the task for any given Priming. From these plots, we settled on 5% as the threshold to characterize a *Mixed* approach

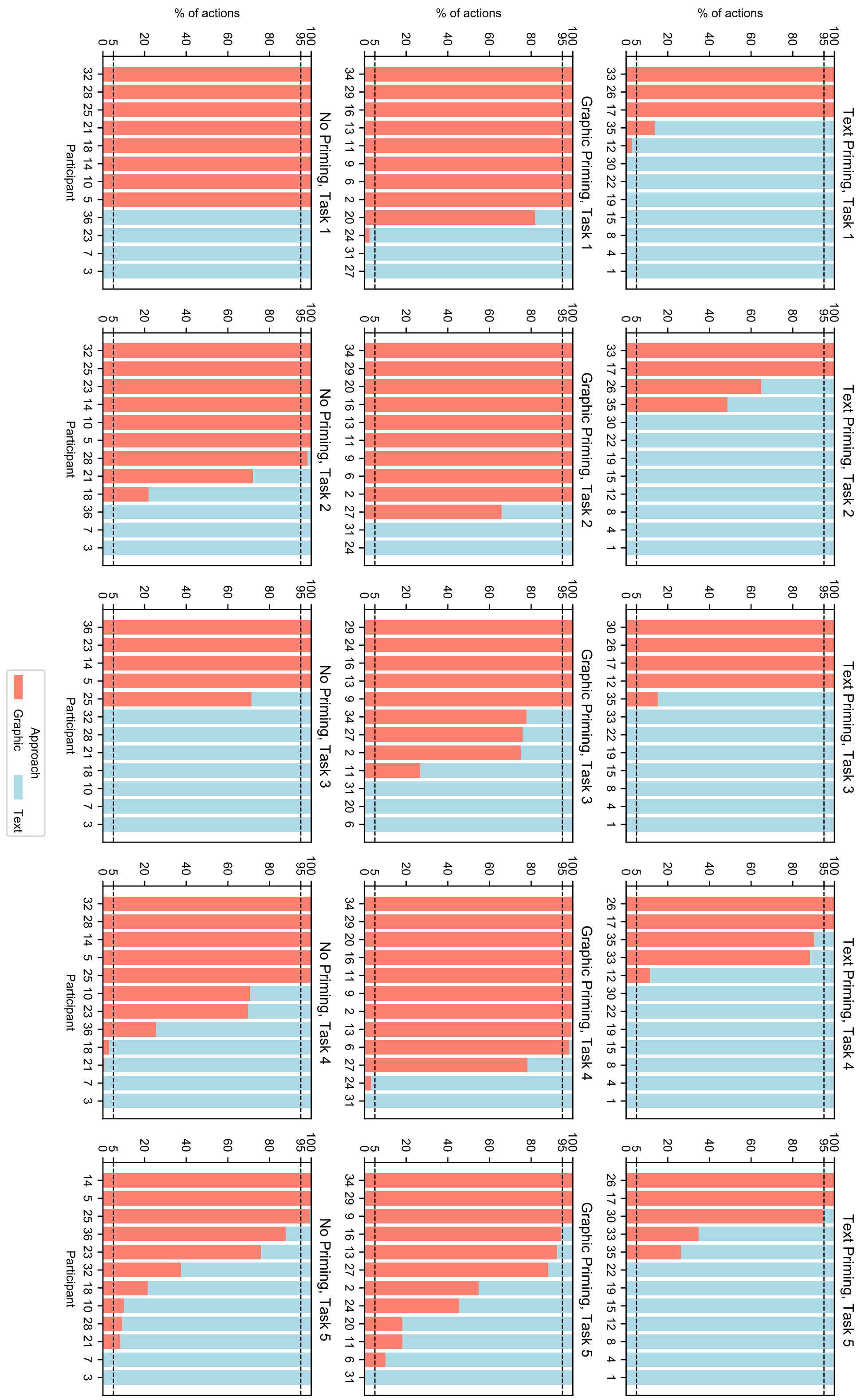


Figure b.1: Percentage of text- or graphic-based commands by participant in descending order by percentage of graphic-based commands. This provides a more detailed visualization of the distribution of graphic and text commands across tasks shown in Figure 4.4. This visualization shows that

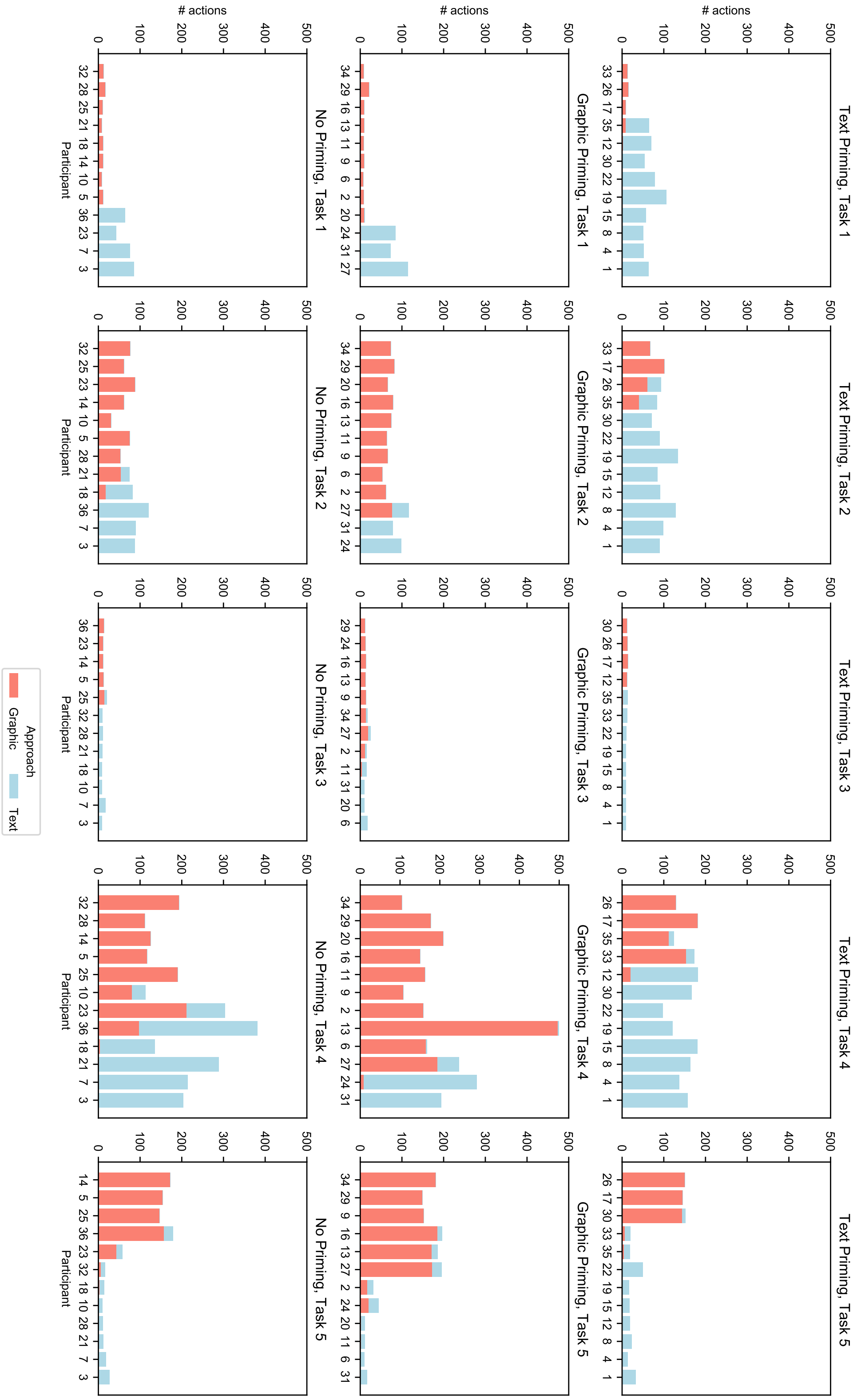


Figure b.2: Absolute number of text- and graphic-based actions for each participant.

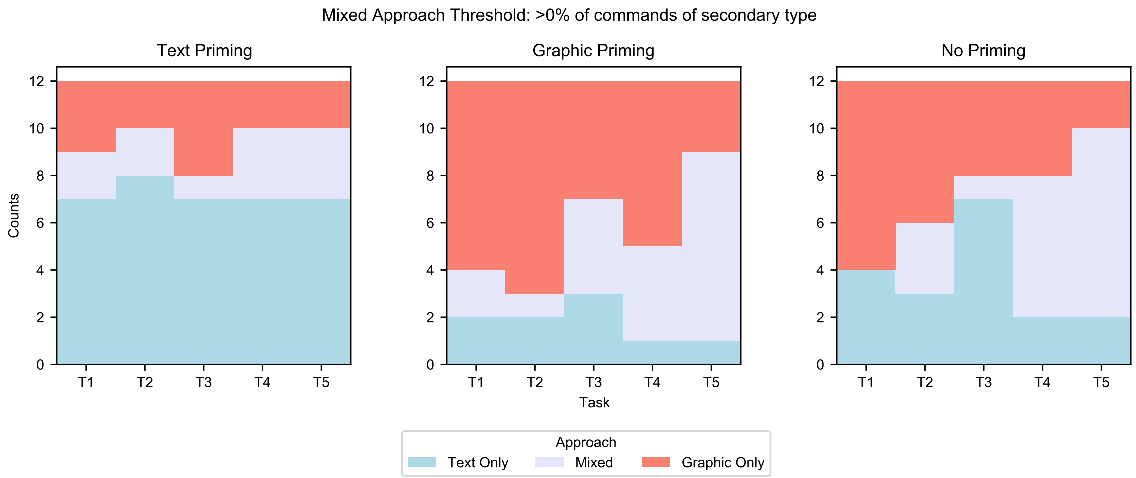


Figure b.3: Defining a *Mixed* approach as using at least one command of a secondary type.

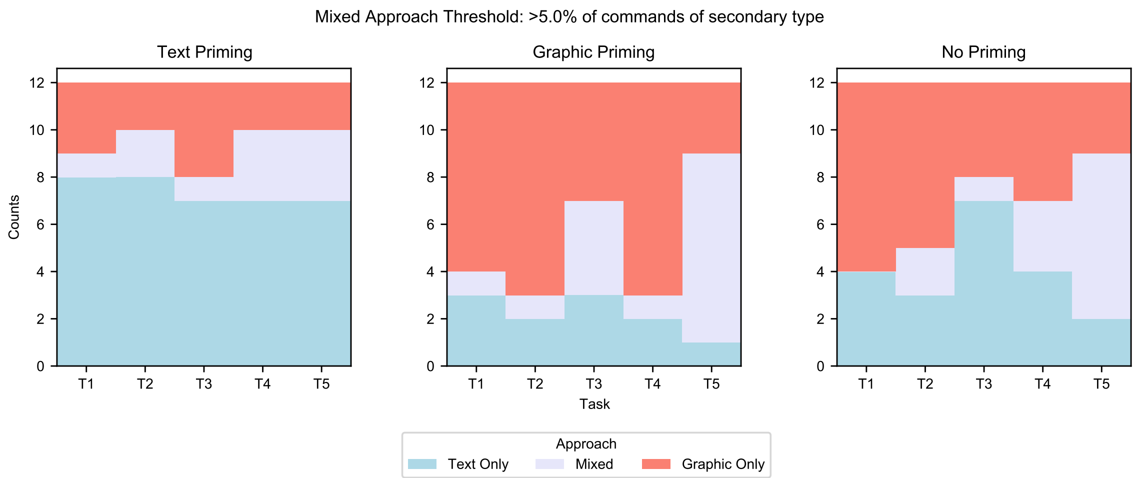


Figure b.4: Defining a *Mixed* approach as more than 5% of commands of a secondary type (used for the analysis).

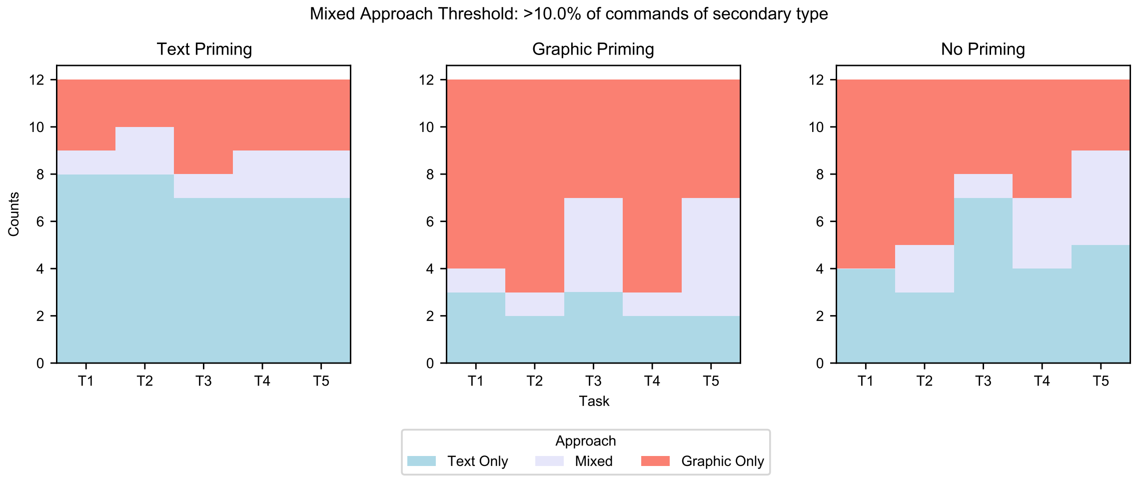


Figure b.5: Defining a *Mixed* approach as more than 10% of commands of a secondary type.

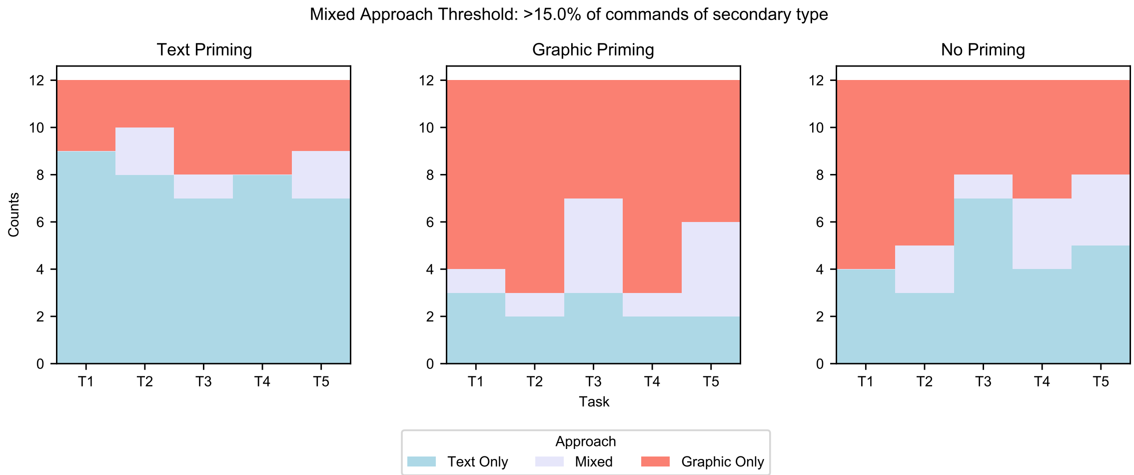


Figure b.6: Defining a *Mixed* approach as more than 15% of commands of a secondary type.

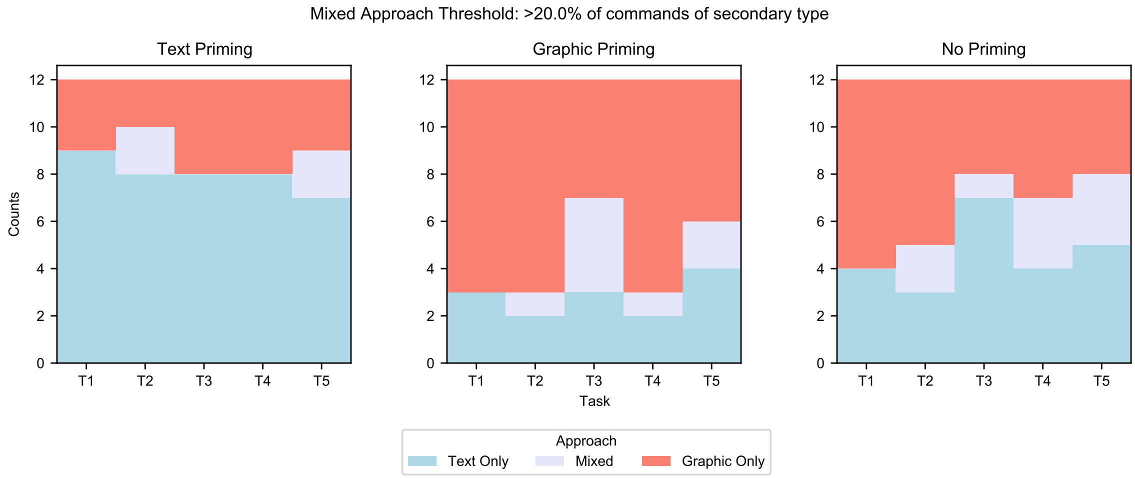


Figure b.7: Defining a *Mixed* approach as more than 20% of commands of a secondary type.

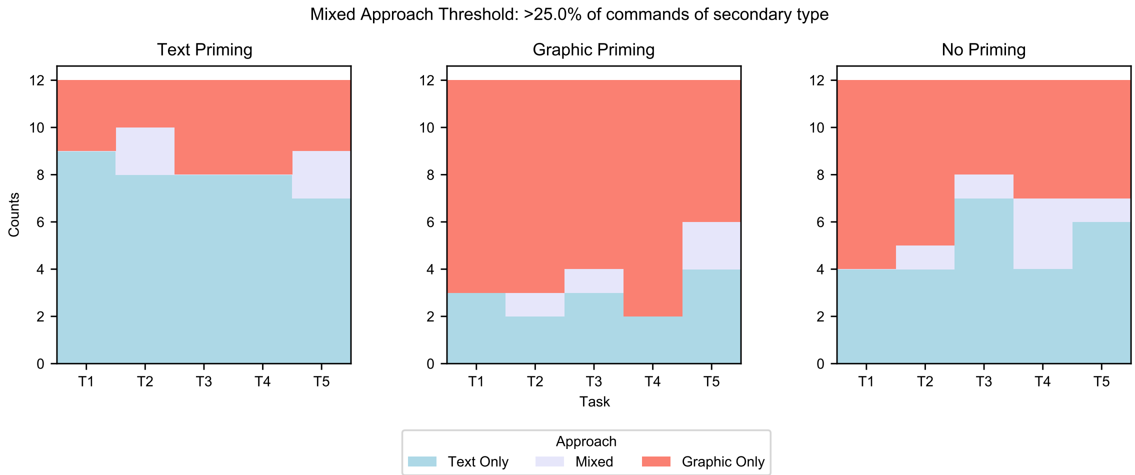


Figure b.8: Defining a *Mixed* approach as more than 25% of commands of a secondary type.

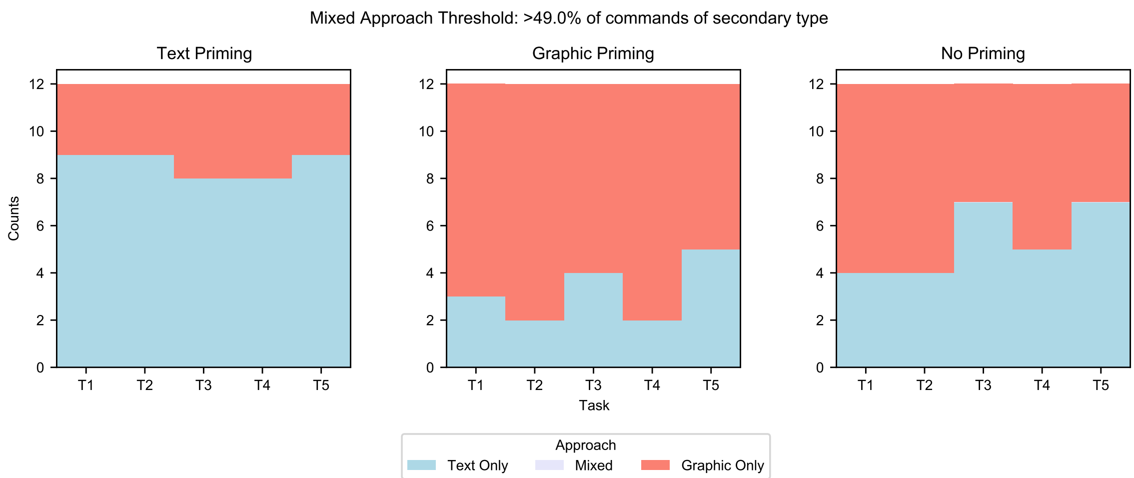


Figure b.9: Defining a *Mixed* approach as more than 49% of commands of a secondary type.

B.2 ANALYSIS OF RESIDUALS

Table b.1: Standardized residuals in overall tasks and tasks 2 and 4.

	Overall			**Task 2**			**Task 4**		
	APPROACH								
Priming	**G**	**M**	**T**	**G**	**M**	**T**	**G**	**M**	**T**
Text Priming	-3.93	-1.12	**4.84**	-2.87	0.62	**2.48**	-2.87	1.10	**2.11**
Graphic Priming	**3.93**	1.12	-4.84	**2.87**	-0.62	-2.48	**2.87**	-1.10	-2.11

An analysis of the standardized residuals for the overall tasks indicates that the number of text approaches increased significantly (> 2 for a small table (Agresti, 2006)) with text priming while it decreased with graphic priming. Likewise, the number of graphic approaches increased significantly with graphic priming while it decreased with text priming. This same pattern repeats in tasks 2 and 4.

B.3 COMMAND TYPES PER PARTICIPANT

The visualizations in this section show a detailed picture by participant of the use of commands of different types. Some tasks could be completed with a smaller action footprint when using certain types. For example, task 5 could be completed using a minimum of 10 actions (9 *text highlight* + 1 *graphic move*) if using a mixed approach, 9 of which would be text-based and 1 graphic-based. However, using only graphic-based commands would imply several times more actions by using the fill tool to color each character background individually.

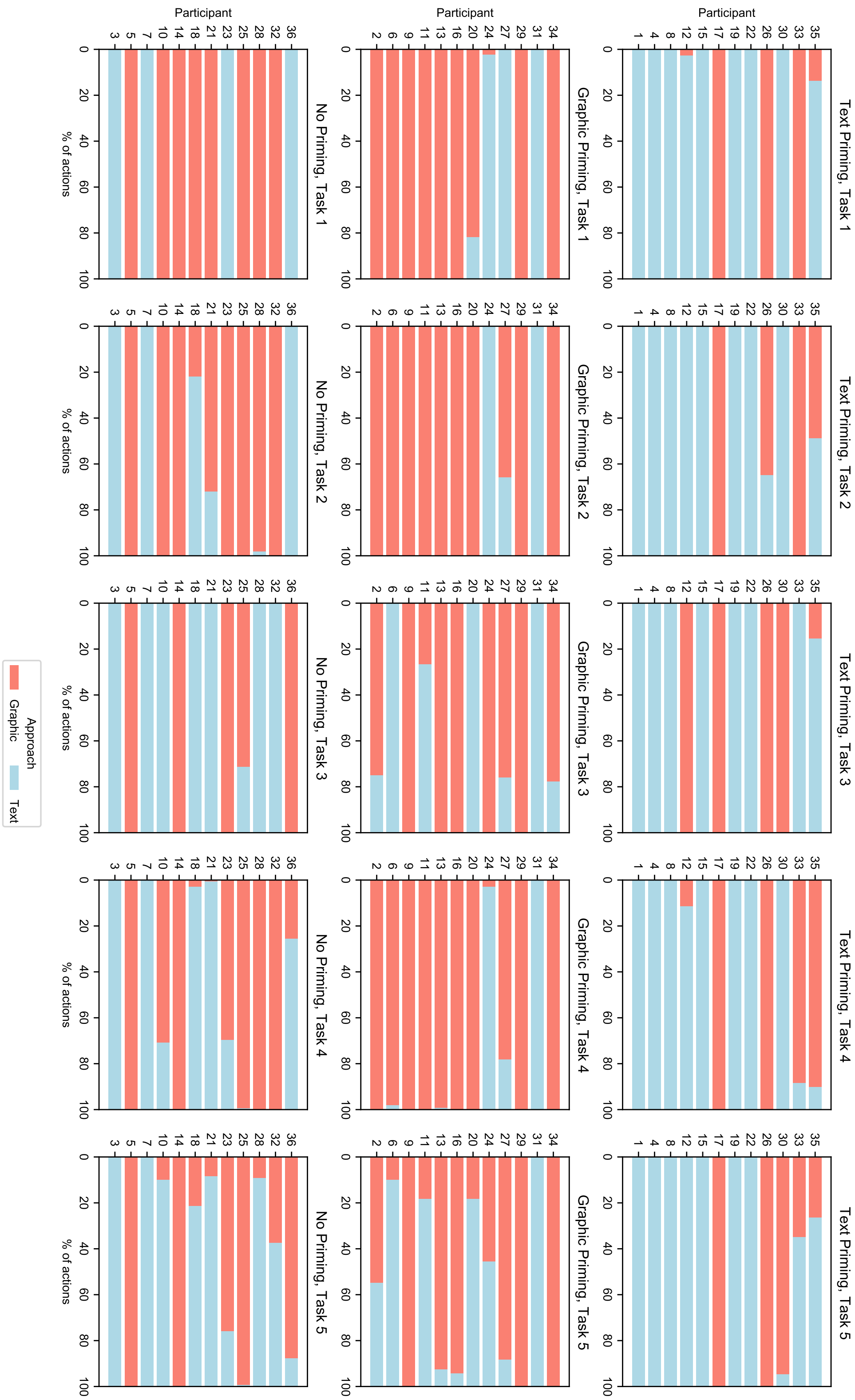


Figure b.10: Distribution of command types per participant. Thin bars adding up to 5% or less of the commands of the secondary type are not considered as mixed approaches.

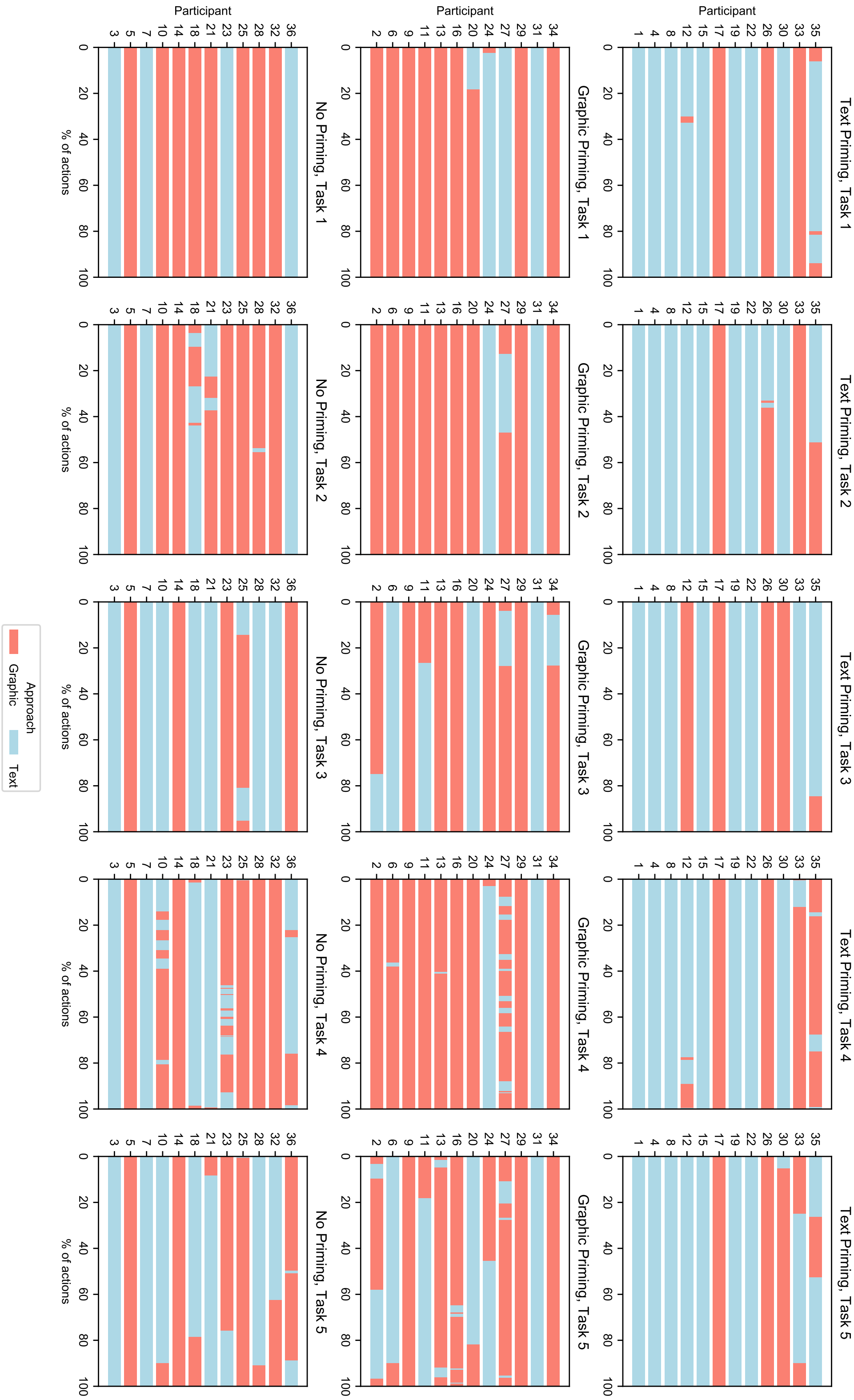


Figure b.11: Data from Figure b.10 displaying the distribution of the percentages in time for each participant. From observation, few participants alternate between text and graphic commands and rather switch and stay in one type until completing the task.

Table b.2: List of actions captured for each participant.

Action	Tool	Type	Description
Cut	Pointer	Graphic	Cut the current graphic selection.
Copy	Pointer	Graphic	Copy the current graphic selection.
Paste	Pointer	Graphic	Paste the contents of the graphic clipboard.
Rectangular Selection	Pointer	Graphic	Select objects contained in a rectangular area defined by dragging the mouse cursor.
Single Selection	Pointer	Graphic	Select a single object by clicking on it.
Move Objects	Pointer	Graphic	Drag and Drop selected objects with the mouse cursor, or drag and drop a single object.
Delete Objects	Pointer	Graphic	Delete selected objects.
Paint Background	Paint Bucket	Graphic	Set the background color of an object.
Cut	I-beam	Text	Cut the current text selection.
Copy	I-beam	Text	Copy the current text selection.
Paste	I-beam	Text	Paste the contents of the text clipboard.
Select	I-beam	Text	Select text by highlighting with either the mouse cursor or Shift + Keyboard Arrows.
Insert Character	I-beam	Text	Insert new visible character.
Insert Newline	I-beam	Text	Insert new line character.
Delete Characters	I-beam	Text	Delete current text selection (or single character).
Highlight Background	Text Highlighter	Text	Set the background color of a text selection.

BIBLIOGRAPHY

Agresti, Alan (2006). *An Introduction to Categorical Data Analysis*. Wiley Series in Probability and Statistics. John Wiley & Sons, Ltd. ISBN: 9780471226185. DOI: 10.1002/0470114754. URL: https://doi.org/10.1002/0470114754.

Alexander, Jason, Andy Cockburn, Stephen Fitchett, Carl Gutwin, and Saul Greenberg (2009). "Revisiting Read Wear: Analysis, Design, and Evaluation of a Footprints Scrollbar." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA: Association for Computing Machinery, pp. 1665–1674. ISBN: 978-1-60558-246-7. DOI: 10.1145/1518701.1518957. URL: https://doi.org/10.1145/1518701.1518957.

Anderson, John R. (1976). *Language, Memory, and Thought*. New York, NY, USA: Psychology Press. DOI: 10.4324/9780203780954. URL: https://doi.org/10.4324/9780203780954.

Anderson, John R. (1983). *The Architecture of Cognition*. New York, NY, USA: Psychology Press. DOI: 10.4324/9781315799438. URL: https://doi.org/10.4324/9781315799438.

Baecker, Ronald M., Dimitrios Nastos, Ilona R. Posner, and Kelly L. Mawby (1993). "The User-centered Iterative Design of Collaborative Writing Software." In: *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*. CHI '93. Amsterdam, The Netherlands: Association for Computing Machinery, pp. 399–405. ISBN: 0-89791-575-5. DOI: 10.1145/169059.169312. URL: https://doi.org/10.1145/169059.169312.

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. New York, NY, USA: Association for Computing Machinery, pp. 446–453. ISBN: 1581132166. DOI: 10.1145/332040.332473.

Beaudouin-Lafon, Michel (2004). "Designing Interaction, Not Interfaces." In: *Proceedings of the Working Conference on Advanced Visual Interfaces*. AVI '04. Gallipoli, Italy: Association for Computing Machinery, pp. 15–22. ISBN: 1-58113-867-9. DOI: 10.1145/989863.989865. URL: https://doi.org/10.1145/989863.989865.

Beaudouin-Lafon, Michel (2017). "Towards Unified Principles of Interaction." In: *Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter*. CHItaly '17. Cagliari, Italy: Association for Computing Machinery. ISBN: 9781450352376. DOI: 10.1145/3125571.3125602. URL: https://doi.org/10.1145/3125571.3125602.

Beaudouin-Lafon, Michel, Susanne Bødker, and Wendy E. Mackay (Nov. 2021). "Generative Theories of Interaction." In: *ACM Trans-*

actions on Computer-Human Interaction 28.6. ISSN: 1073-0516. DOI: 10.1145/3468505. URL: https://doi.org/10.1145/3468505.

Beaudouin-Lafon, Michel and Henry Michael Lassen (2000). "The Architecture and Implementation of CPN2000, a Post-WIMP Graphical Application." In: *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*. UIST '00. San Diego, California, USA: Association for Computing Machinery, pp. 181–190. ISBN: 1581132123. DOI: 10.1145/354401.354761. URL: https://doi.org/10.1145/354401.354761.

Beaudouin-Lafon, Michel and Wendy E. Mackay (2000). "Reification, Polymorphism and Reuse: Three Principles for Designing Visual Interfaces." In: *Proceedings of the Working Conference on Advanced Visual Interfaces*. AVI '00. New York, NY, USA: Association for Computing Machinery, pp. 102–109. ISBN: 1581132522. DOI: 10.1145/345513.345267.

Béguin, Pascal and Pierre Rabardel (2000). "Designing for Instrument-mediated Activity." In: *Scandinavian Journal of Information Systems* 12.1, p. 1. URL: https://aisel.aisnet.org/sjis/vol12/iss1/1.

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. Crossings, Seoul, Korea: Association for Computing Machinery, pp. 1315–1324. ISBN: 9781450331456. DOI: 10.1145/2702123.2702359. URL: https://doi.org/10.1145/2702123.2702359.

Bergin, Tim (Nov. 2006). "The Origins of Word Processing Software for Personal Computers: 1976-1985." In: *Annals of the History of Computing, IEEE* 28, pp. 32–47. DOI: 10.1109/MAHC.2006.76. URL: https://doi.org/10.1109/MAHC.2006.76.

Bergström, Joanna, Aske Mottelson, Andreea Muresan, and Kasper Hornbæk (2019). "Tool Extension in Human-Computer Interaction." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, pp. 1–11. ISBN: 9781450359702. DOI: 10.1145/3290605.3300798. URL: https://doi.org/10.1145/3290605.3300798.

Bernstein, Michael S., Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich (July 2015). "Soylent: A Word Processor with a Crowd Inside." In: *Commun. ACM* 58.8, pp. 85–94. ISSN: 0001-0782. DOI: 10.1145/2791285. URL: https://doi.org/10.1145/2791285.

Bier, Eric A., Edward W. Ishak, and Ed Chi (2006). "Entity Quick Click: Rapid Text Copying Based on Automatic Entity Extraction." In: *CHI '06 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '06. Montréal, Québec, Canada: Association for Computing

Machinery, pp. 562–567. ISBN: 1595932984. DOI: 10.1145/1125451.1125570. URL: https://doi.org/10.1145/1125451.1125570.

Bødker, Susanne (Apr. 1987). "Through the Interface – a Human Activity Approach to User Interface Design." In: *DAIMI Report Series* 16.224, p. 190. DOI: 10.7146/dpb.v16i224.7586. URL: https://doi.org/10.7146/dpb.v16i224.7586.

Bødker, Susanne (1997). "Computers in Mediated Human Activity." In: *Mind, Culture, and Activity* 4.3, pp. 149–158. DOI: 10.1207/s15327884mca0403_2. URL: https://doi.org/10.1207/s15327884mca0403_2.

Bødker, Susanne and Clemens Nylandsted Klokmose (2016). "Dynamics, Multiplicity and Conceptual Blends in HCI." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: Association for Computing Machinery, pp. 2538–2548. ISBN: 9781450333627. DOI: 10.1145/2858036.2858530. URL: https://doi.org/10.1145/2858036.2858530.

Brandt, Joel, Philip J. Guo, Joel Lewenstein, Mira Dontcheva, and Scott R. Klemmer (2009). "Two Studies of Opportunistic Programming: Interleaving Web Foraging, Learning, and Writing Code." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA: Association for Computing Machinery, pp. 1589–1598. ISBN: 978-1-60558-246-7. DOI: 10.1145/1518701.1518944. URL: https://doi.org/10.1145/1518701.1518944.

Braun, Virginia and Victoria Clarke (2006). "Using Thematic Analysis in Psychology." In: *Qualitative Research in Psychology* 3.2, pp. 77–101.

Braun, Virginia and Victoria Clarke (2012). "Thematic Analysis." In: *APA Handbook of Research Methods in Psychology* 2, pp. 57–71. DOI: 10.1037/13620-004. URL: https://doi.org/doi/10.1037/13620-004.

Braun, Virginia and Victoria Clarke (2019). "Reflecting on Reflexive Thematic Analysis." In: *Qualitative Research in Sport, Exercise and Health* 11.4, pp. 589–597. DOI: 10.1080/2159676X.2019.1628806. URL: https://doi.org/10.1080/2159676X.2019.1628806.

Buxbaum, Laurel J. and Solène Kalénine (2010). "Action Knowledge, Visuomotor Activation, and Embodiment in the Two Action Systems." In: *Annals of the New York Academy of Sciences* 1191.1, pp. 201–218. DOI: 10.1111/j.1749-6632.2010.05447.x. URL: https://doi.org/10.1111/j.1749-6632.2010.05447.x.

Card, Stuart K., Thomas P. Moran, and Allen Newell (1980). "Computer Text-editing: An Information-processing Analysis of a Routine Cognitive Skill." In: *Cognitive Psychology* 12.1, pp. 32–74. ISSN: 0010-0285. DOI: 10.1016/0010-0285(80)90003-1. URL: https://doi.org/10.1016/0010-0285(80)90003-1.

Cardinali, Lucilla, Francesca Frassinetti, Claudio Brozzoli, Christian Urquizar, Alice C Roy, and Alessandro Farnè (2009). "Tool-use Induces Morphological Updating of The Body Schema." In: *Current* *biology* 19.12, R478–R479. DOI: 10.1016/j.cub.2009.05.009. URL: https://doi.org/10.1016/j.cub.2009.05.009.

Carroll, John M., Robert L. Mack, and Wendy A. Kellogg (1988). "Chapter 3 - Interface Metaphors and User Interface Design." In: *Handbook of Human-Computer Interaction*. Ed. by MARTIN HELANDER. Amsterdam: North-Holland, pp. 67–85. ISBN: 978-0-444-70536-5. DOI: 10.1016/B978-0-444-70536-5.50008-7. URL: https://doi.org/10.1016/B978-0-444-70536-5.50008-7.

Carroll, John M. and Mary Beth Rosson (1987). "Paradox of the Active User." In: *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. Cambridge, MA, USA: MIT Press, pp. 80–111. ISBN: 0262031256.

Chapuis, Olivier and Nicolas Roussel (2007). "Copy-and-paste Between Overlapping Windows." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. San Jose, California, USA: Association for Computing Machinery, pp. 201–210. ISBN: 978-1-59593-593-9. DOI: 10.1145/1240624.1240657. URL: https://doi.org/10.1145/1240624.1240657.

Cherubini, Mauro, Gina Venolia, Rob DeLine, and Andrew J. Ko (2007). "Let's Go to the Whiteboard: How and Why Software Developers Use Drawings." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. San Jose, California, USA: Association for Computing Machinery, pp. 557–566. ISBN: 978-1-59593-593-9. DOI: 10.1145/1240624.1240714. URL: https://doi.org/10.1145/1240624.1240714.

Churchill, Elizabeth F., Jonathan Trevor, Sara Bly, Les Nelson, and Davor Cubranic (2000). "Anchored Conversations: Chatting in the Context of a Document." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '00. The Hague, The Netherlands: Association for Computing Machinery, pp. 454–461. ISBN: 1-58113-216-6. DOI: 10.1145/332040.332475. URL: https://doi.org/10.1145/332040.332475.

Ciolfi Felice, Marianela, Nolwenn Maudet, Wendy E. Mackay, and Michel Beaudouin-Lafon (2016). "Beyond Snapping: Persistent, Tweakable Alignment and Distribution with StickyLines." In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. UIST '16. Tokyo, Japan: Association for Computing Machinery, pp. 133–144. ISBN: 9781450341899. DOI: 10.1145/2984511.2984577.

Cockburn, Andy, Carl Gutwin, Joey Scarr, and Sylvain Malacria (Nov. 2014). "Supporting Novice to Expert Transitions in User Interfaces." In: *ACM Comput. Surv.* 47.2. ISSN: 0360-0300. DOI: 10.1145/2659796. URL: https://doi.org/10.1145/2659796.

Cohen, Andrew L., Debra Cash, and Michael J. Muller (1999). "Awareness, Planning and Joint Attention in Collaborative Writing: From Fieldwork to Design." In: *LOTUS CODE# 1999.02*, pp. 94–101. URL: https://www.researchgate.net/publication/239724134_Awareness_

Planning_and_Joint_Attention_in_Collaborative_Writing_From_Fieldwork_to_Design.

Coughlan, Tim and Peter Johnson (2009). "Understanding Productive, Structural and Longitudinal Interactions in the Design of Tools for Creative Activities." In: *Proceedings of the Seventh ACM Conference on Creativity and Cognition*. C&C '09. Berkeley, California, USA: Association for Computing Machinery, pp. 155–164. ISBN: 9781605588650. DOI: 10.1145/1640233.1640258. URL: https://doi.org/10.1145/1640233.1640258.

Dix, Alan (2007). "Designing for Appropriation." In: *Proceedings of HCI 2007 The 21st British HCI Group Annual Conference University of Lancaster, UK 21*. London, UK: British Computer Society, pp. 1–4. DOI: 10.14236/ewic/HCI2007.53. URL: https://doi.org/10.14236/ewic/HCI2007.53.

Domencich, Thomas A and Daniel McFadden (1975). *Urban Travel Demand: A Behavioral Analysis*. NY, USA: North-Holland Publishing Company. URL: http://www.worldcat.org/oclc/915919185.

Douglas, Sarah A. and Thomas P. Moran (1983). "Learning Text Editor Semantics by Analogy." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '83. Boston, Massachusetts, USA: Association for Computing Machinery, pp. 207–211. ISBN: 0897911210. DOI: 10.1145/800045.801612. URL: https://doi.org/10.1145/800045.801612.

Dourish, Paul (2003). "The Appropriation of Interactive Technologies: Some Lessons from Placeless Documents." In: *Computer Supported Cooperative Work (CSCW)* 12.4, pp. 465–490. DOI: 10.1023/A:1026149119426. URL: https://doi.org/10.1023/A:1026149119426.

Dragicevic, Pierre, Yvonne Jansen, Abhraneel Sarma, Matthew Kay, and Fanny Chevalier (2019). "Increasing the Transparency of Research Papers with Explorable Multiverse Analyses." In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 65:1–65:15. ISBN: 978-1-4503-5970-2. DOI: 10.1145/3290605.3300295. URL: https://doi.org/10.1145/3290605.3300295.

Duncker, Karl and Lynne S Lees (1945). "On Problem-solving." In: *Psychological Monographs* 58.5, p. i. DOI: 10.1037/h0093599. URL: https://doi.org/10.1037/h0093599.

Ericsson, K Anders, Ralf T Krampe, and Clemens Tesch-Römer (1993). "The Role of Deliberate Practice in the Acquisition of Expert Performance." In: *Psychological review* 100.3, pp. 363–406. DOI: 10.1037/0033-295X.100.3.363. URL: https://doi.org/10.1037/0033-295X.100.3.363.

Farkas, David K. (Oct. 1985). "The Concept of Consistency in Writing and Editing." In: *Journal of Technical Writing and Communication* 15.4, pp. 353–364. DOI: 10.2190/2FT6EM-UTT0-EL6J-59N9. URL: https://doi.org/10.2190/2FT6EM-UTT0-EL6J-59N9.

Fitts, Paul M and Michael I Posner (1967). *Human Performance*. Michigan, US: Brooks/Cole.

Freeman, G. H. and John H. Halton (1951). "Note on an Exact Treatment of Contingency, Goodness of Fit and Other Problems of Significance." In: *Biometrika* 38.1-2, pp. 141–149. DOI: 10.2307/2332323. URL: https://doi.org/10.2307/2332323.

Gaver, William W. (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.

Gaver, William W., Jacob Beaver, and Steve Benford (2003). "Ambiguity as a Resource for Design." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '03. Ft. Lauderdale, Florida, USA: Association for Computing Machinery, pp. 233–240. ISBN: 1581136307. DOI: 10.1145/642611.642653. URL: https://doi.org/10.1145/642611.642653.

Gibson, E. J. (1963). "Perceptual Learning." In: *Annual Review of Psychology* 14.1, pp. 29–56. DOI: 10.1146/annurev.ps.14.020163.000333.

Gibson, J. J. (1986). *The Ecological Approach to Visual Perception*. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, Inc. ISBN: 9780203767764. DOI: 10.4324/9780203767764. URL: https://doi.org/10.4324/9780203767764.

Gick, Mary L. and Keith J. Holyoak (1980). "Analogical Problem Solving." In: *Cognitive Psychology* 12.3, pp. 306–355. ISSN: 0010-0285. DOI: 10.1016/0010-0285(80)90013-4. URL: https://doi.org/10.1016/0010-0285(80)90013-4.

Gilbert, Myrna and Martyn Cordey-Hayes (1996). "Understanding the Process of Knowledge Transfer to Achieve Successful Technological Innovation." In: *Technovation* 16.6, pp. 301–312. ISSN: 0166-4972. DOI: 10.1016/0166-4972(96)00012-0. URL: https://doi.org/10.1016/0166-4972(96)00012-0.

Goffin, P., J. Boy, W. Willett, and P. Isenberg (Oct. 2017). "An Exploratory Study of Word-Scale Graphics in Data-Rich Text Documents." In: *IEEE Transactions on Visualization and Computer Graphics* 23.10, pp. 2275–2287. ISSN: 2160-9306. DOI: 10.1109/TVCG.2016.2618797. URL: https://doi.org/10.1109/TVCG.2016.2618797.

Goldberg, Lewis R., John A. Johnson, Herbert W. Eber, Robert Hogan, Michael C. Ashton, C. Robert Cloninger, and Harrison G. Gough (2006). "The International Personality Item Pool and the Future of Public-domain Personality Measures." In: *Journal of Research in Personality* 40.1. Proceedings of the 2005 Meeting of the Association of Research in Personality, pp. 84–96. ISSN: 0092-6566. DOI: 10.1016/j.jrp.2005.08.007. URL: https://doi.org/10.1016/j.jrp.2005.08.007.

Han, Han L., Miguel A. Renom, Wendy E. Mackay, and Michel Beaudouin-Lafon (2020). "Textlets: Supporting Constraints and Consistency in Text Documents." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, pp. 1–13. ISBN: 9781450367080. DOI: 10.1145/3313831.3376804. URL: https://doi.org/10.1145/3313831.3376804.

Han, Han L., Junhang Yu, Alexandre Ciorascu, Raphael Bournet, Wendy E. Mackay, and Michel Beaudouin-Lafon (2022). "Passages: Reading and Interacting with Text Across Documents." In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery. ISBN: 9781450391573. DOI: 10.1145/3491102.3502052. URL: https://doi.org/10.1145/3491102.3502052. Forthcoming.

Hartmann, Björn, Daniel MacDougall, Joel Brandt, and Scott R. Klemmer (2010). "What Would Other Programmers Do: Suggesting Solutions to Error Messages." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '10. Atlanta, Georgia, USA: Association for Computing Machinery, pp. 1019–1028. ISBN: 978-1-60558-929-9. DOI: 10.1145/1753326.1753478. URL: https://doi.org/10.1145/1753326.1753478.

Hill, William C., James D. Hollan, Dave Wroblewski, and Tim McCandless (1992). "Edit Wear and Read Wear." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '92. Monterey, California, USA: Association for Computing Machinery, pp. 3–9. ISBN: 0-89791-513-5. DOI: 10.1145/142750.142751. URL: https://doi.org/10.1145/142750.142751.

Hoffman, Robert R. (Apr. 1989). "A Survey of Methods for Eliciting the Knowledge of Experts." In: *SIGART Bull.* 1.108, pp. 19–27. ISSN: 0163-5719. DOI: 10.1145/63266.63269. URL: https://doi.org/10.1145/63266.63269.

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. URL: https://doi.org/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: Association for Computing Machinery, pp. 201–210. ISBN: 9781605580111. DOI: 10.1145/1357054.1357089. URL: https://doi.org/10.1145/1357054.1357089.

Jalal, Ghita, Nolwenn Maudet, and Wendy E. Mackay (2015). "Color Portraits: From Color Picking to Interacting with Color." 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. 4207–4216. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702173. URL: https://doi.org/10.1145/2702123.2702173.

Jetter, Hans-Christian, Harald Reiterer, and Florian Geyer (June 2014). "Blended Interaction: Understanding Natural Human—Computer Interaction in Post-WIMP Interactive Spaces." In: *Personal Ubiquitous Computing* 18.5, pp. 1139–1158. ISSN: 1617-4909. DOI: 10.1007/s00779-013-0725-4. URL: https://doi.org/10.1007/s00779-013-0725-4.

Johnson, J., T.L. Roberts, W. Verplank, D.C. Smith, C.H. Irby, M. Beard, and K. Mackey (1989). "The Xerox Star: a Retrospective." In: *Computer* 22.9, pp. 11–26. DOI: 10.1109/2.35211.

Johnson-Laird, P. N. (1989). "Mental Models." In: *Foundations of Cognitive Science*. Cambridge, MA, USA: MIT Press, pp. 469–499. ISBN: 0262161125.

Kapser, Cory J. and Michael W. Godfrey (Dec. 2008). ""Cloning Considered Harmful" Considered Harmful: Patterns of Cloning in Software." In: *Empirical Software Engineering* 13.6, pp. 645–692. ISSN: 1382-3256. DOI: 10.1007/s10664-008-9076-6. URL: https://doi.org/10.1007/s10664-008-9076-6.

Kaptelinin, Victor (1996). "Computer-mediated Activity: Functional Organs in Social and Developmental Contexts." In: *Context and Consciousness: Activity Theory and Human-Computer Interaction*, pp. 45–68.

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. URL: https://doi.org/10.1145/2207676.2208541.

Kaptelinin, Victor and Bonnie A. Nardi (1997). "Activity Theory: Basic Concepts and Applications." In: *CHI '97 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '97. Atlanta, Georgia: Association for Computing Machinery, pp. 158–159. ISBN: 0897919262. DOI: 10.1145/1120212.1120321. URL: https://doi.org/10.1145/1120212.1120321.

Karat, J., L. Boyes, S. Weisgerber, and C. Schafer (1986). "Transfer Between Word Processing Systems." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '86. Boston, Massachusetts, USA: Association for Computing Machinery, pp. 67–71. ISBN: 0897911806. DOI: 10.1145/22627.22350. URL: https://doi.org/10.1145/22627.22350.

Karlesky, Michael and Katherine Isbister (2016). "Understanding Fidget Widgets: Exploring the Design Space of Embodied Self-Regulation." In: *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*. NordiCHI '16. Gothenburg, Sweden: Association for Computing Machinery. ISBN: 9781450347631. DOI: 10.1145/2971485.2971557. URL: https://doi.org/10.1145/2971485.2971557.

Kery, Mary Beth, Amber Horvath, and Brad Myers (2017). "Variolite: Supporting Exploratory Programming by Data Scientists." In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: Association for Computing Machinery, pp. 1265–1276. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025626. URL: https://doi.org/10.1145/3025453.3025626.

Kim, Miryung, Lawrence Bergman, Tessa Lau, and David Notkin (2004). "An Ethnographic Study of Copy and Paste Programming Practices in OOPL." In: *Proceedings of the 2004 International Symposium on Empirical Software Engineering*. ISESE '04. Washington, DC, USA: IEEE Computer Society, pp. 83–92. ISBN: 0-7695-2165-7. DOI: 10.1109/ISESE.2004.10. URL: https://doi.org/10.1109/ISESE.2004.10.

Kirsh, David (Apr. 2013). "Embodied Cognition and the Magical Future of Interaction Design." In: *ACM Transactions on Computer-Human Interaction* 20.1. ISSN: 1073-0516. DOI: 10.1145/2442106.2442109. URL: https://doi.org/10.1145/2442106.2442109.

Ko, Amy J., Htet Aung, and Brad A. Myers (2005). "Eliciting Design Requirements for Maintenance-Oriented IDEs: A Detailed Study of Corrective and Perfective Maintenance Tasks." In: *Proceedings of the 27th International Conference on Software Engineering*. ICSE '05. St. Louis, MO, USA: Association for Computing Machinery, pp. 126–135. ISBN: 1581139632. DOI: 10.1145/1062455.1062492. URL: https://doi.org/10.1145/1062455.1062492.

Ko, Andrew J. and Brad A. Myers (2004). "Designing the Whyline: A Debugging Interface for Asking Questions About Program Behavior." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '04. Vienna, Austria: Association for Computing Machinery, pp. 151–158. ISBN: 1-58113-702-8. DOI: 10.1145/985692.985712. URL: https://doi.org/10.1145/985692.985712.

Ko, Andrew J. and Brad A. Myers (2006). "Barista: An Implementation Framework for Enabling New Tools, Interaction Techniques and Views in Code Editors." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '06. Montréal, Québec, Canada: Association for Computing Machinery, pp. 387–396. ISBN: 1-59593-372-7. DOI: 10.1145/1124772.1124831. URL: https://doi.org/10.1145/1124772.1124831.

Laakso, Sari A., Karri Pekka Laakso, and Asko J. Saura (2000). "Improved Scroll Bars." In: *CHI '00 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '00. The Hague, The Netherlands: Association for Computing Machinery, pp. 97–98. ISBN: 1-58113-248-4. DOI: 10.1145/633292.633350. URL: https://doi.org/10.1145/633292.633350.

Leontiev, Aleksei N. (1978). *Activity, Consciousness, and Personality*. Englewood Cliffs, UK: Prentice-Hall.

Lindley, Siân E., Gavin Smyth, Robert Corish, Anastasia Loukianov, Michael Golembewski, Ewa A. Luger, and Abigail Sellen (2018). "Exploring New Metaphors for a Networked World Through the File Biography." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 118:1–118:12. ISBN: 978-1-4503-5620-6. DOI: 10.1145/3173574.3173692. URL: https://doi.org/10.1145/3173574.3173692.

Mackay, Wendy E. (May 1990). "Users and Customizable Software: A Co-Adaptive Phenomenon." PhD thesis. Cambridge, MA: Massachusetts Institute of Technology.

Mackay, Wendy E. (2002). "Using Video to Support Interaction Design." In: *DVD Tutorial, CHI* 2.5.

Mackay, Wendy E. and Anne-Laure Fayard (1997). "HCI, Natural Science and Design: A Framework for Triangulation across Disciplines." In: *Proceedings of the 2nd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*. DIS '97. New York, NY, USA: Association for Computing Machinery, pp. 223–234. ISBN: 0897918630. DOI: 10.1145/263552.263612.

Maier, N.R.F. (1931). "Reasoning in Humans. II. The Solution of a Problem and its Appearance in Consciousness." In: *Journal of Comparative Psychology* 12.2, pp. 181–194. DOI: 10.1037/h0071361. URL: https://doi.org/10.1037/h0071361.

Marshall, Catherine C. (1997). "Annotation: From Paper Books to the Digital Library." In: *Proceedings of the Second ACM International Conference on Digital Libraries*. DL '97. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, pp. 131–140. ISBN: 0-89791-868-1. DOI: 10.1145/263690.263806. URL: https://doi.org/10.1145/263690.263806.

McGrenere, Joanna (2002). "The Design and Evaluation of Multiple Interfaces: A Solution for Complex Software." PhD thesis. Toronto, Ontario, Canada: The University of Toronto.

McGrenere, Joanna and Wayne Ho (2000). "Affordances: Clarifying and Evolving a Concept." In: *Proceedings of Graphics Interface 2000*. GI 2000. Montréal, Québec, Canada: Canadian Human-Computer Communications Society, pp. 179–186. ISBN: 0-9695338-9-6. DOI: 10.20380/GI2000.24. URL: https://doi.org/10.20380/GI2000.24.

Miller, Robert C. and Alisa M. Marshall (2004). "Cluster-based Find and Replace." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '04. Vienna, Austria: Association for Computing Machinery, pp. 57–64. ISBN: 1-58113-702-8. DOI: 10.1145/985692.985700. URL: https://doi.org/10.1145/985692.985700.

Miller, Robert C. and Brad A. Myers (2002a). "LAPIS: Smart Editing with Text Structure." In: *CHI '02 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '02. Minneapolis, Minnesota, USA: Association for Computing Machinery, pp. 496–497. ISBN: 1-58113-454-1. DOI: 10.1145/506443.506447. URL: https://doi.org/10.1145/506443.506447.

Miller, Robert C. and Brad A. Myers (2002b). "Multiple Selections in Smart Text Editing." In: *Proceedings of the 7th International Conference on Intelligent User Interfaces*. IUI '02. San Francisco, California, USA: Association for Computing Machinery, pp. 103–110. ISBN: 1-58113-459-2. DOI: 10.1145/502716.502734. URL: https://doi.org/10.1145/502716.502734.

Myers, Brad A. (1991). "Text Formatting by Demonstration." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '91. New Orleans, Louisiana, USA: Association for Computing Machinery, pp. 251–256. ISBN: 0897913833. DOI: 10.1145/108844.108904. URL: https://doi.org/10.1145/108844.108904.

Noël, Sylvie and Jean-Marc Robert (Jan. 2004). "Empirical Study on Collaborative Writing: What Do Co-authors Do, Use, and Like?" In: *Computer Supported Cooperative Work* 13.1, pp. 63–89. ISSN: 0925-9724. DOI: 10.1023/B:COSU.0000014876.96003.be. URL: https://doi.org/10.1023/B:COSU.0000014876.96003.be.

Norman, Donald A. (1988). *The Psychology of Everyday Things*. New York, NY, USA: Basic Books. ISBN: 9780385267748.

Norman, Donald A. (Nov. 2008). "The Way I See It: Signifiers, Not Affordances." In: *Interactions* 15.6, pp. 18–19. ISSN: 1072–5520. DOI: 10.1145/1409040.1409044.

Norman, Donald A. (2013). *The Design of Everyday Things*. New York, NY, USA: Basic Books. ISBN: 9780465050659.

Nouwens, Midas and Clemens Nylandsted Klokmose (2018). "The Application and Its Consequences for Non-Standard Knowledge Work." In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, pp. 1–12. ISBN: 9781450356206. URL: https://doi.org/10.1145/3173574.3173973.

O'Hara, Kenton and Abigail Sellen (1997). "A Comparison of Reading Paper and On-line Documents." In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. CHI '97. Atlanta, Georgia, USA: Association for Computing Machinery, pp. 335–342. ISBN: 0-89791-802-9. DOI: 10.1145/258549.258787. URL: https://doi.org/10.1145/258549.258787.

O'Hara, Kenton P. and Stephen J. Payne (1999). "Planning and the User Interface: The Effects of Lockout Time and Error Recovery Cost." In: *International Journal of Human-Computer Studies* 50.1, pp. 41–59. ISSN: 1071-5819. DOI: 10.1006/ijhc.1998.0234. URL: https://doi.org/10.1006/ijhc.1998.0234.

Oh, Uran and Leah Findlater (2013). "The Challenges and Potential of End-User Gesture Customization." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, pp. 1129–1138. ISBN: 9781450318990. DOI: 10.1145/2470654.2466145. URL: https://doi.org/10.1145/2470654.2466145.

Oney, Stephen and Joel Brandt (2012). "Codelets: Linking Interactive Documentation and Example Code in the Editor." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: Association for Computing Machinery, pp. 2697–2706. ISBN: 978-1-4503-1015-4. DOI: 10.1145/2207676.2208664. URL: https://doi.org/10.1145/2207676.2208664.

Osiurak, François (2020). "A World Without a Technical Solution." In: *The Tool Instinct*. John Wiley & Sons, Ltd. Chap. 4, pp. 59–88. ISBN: 9781119720348. DOI: https://doi.org/10.1002/9781119720348.ch4. URL: https://doi.org/10.1002/9781119720348.ch4.

Osiurak, François and Arnaud Badets (2016). "Tool Use and Affordance: Manipulation-based Versus Reasoning-based Approaches." In: *Psychological Review* 123.5, pp. 534–568. DOI: 10.1037/rev0000027. URL: http://dx.doi.org/10.1037/rev0000027.

Osiurak, François, Christophe Jarry, Philippe Allain, Ghislaine Aubin, Frédérique Etcharry-Bouyx, Isabelle Richard, Isabelle Bernard, and Didier Le Gall (2009). "Unusual use of objects after unilateral brain damage. The technical reasoning model." In: *Cortex* 45.6, pp. 769–783. ISSN: 00109452. DOI: 10.1016/j.cortex.2008.06.013.

Osiurak, François, Christophe Jarry, and Didier Le Gall (2010). "Grasping the Affordances, Understanding the Reasoning: Toward a Dialectical Theory of Human Tool Use." In: *Psychological Review*, pp. 517–540. DOI: 10.1037/A0019004.

Polson, Peter G., Susan Bovair, and David Kieras (1986). "Transfer Between Text Editors." In: *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*. CHI '87. Toronto, Ontario, Canada: Association for Computing Machinery, pp. 27–32. ISBN: 0897912136. DOI: 10.1145/29933.30856. URL: https://doi.org/10.1145/29933.30856.

Rabardel, Pierre (1995). *Les hommes et les technologies: approche cognitive des instruments contemporains*. Armand Colin.

Raskin, Jef (2000). *The Humane Interface: New Directions for Designing Interactive Systems*. USA: ACM Press/Addison-Wesley Publishing Company. ISBN: 0201379376.

Renom, Miguel A., Baptiste Caramiaux, and Michel Beaudouin-Lafon (2022). "Exploring Technical Reasoning in Digital Tool Use." In: *CHI Conference on Human Factors in Computing Systems*. CHI '22. New Orleans, LA, USA: Association for Computing Machinery. ISBN: 9781450391573. DOI: 10.1145/3491102.3501877. URL: https://doi.org/10.1145/3491102.3501877.

Rieman, John, Clayton Lewis, Richard M. Young, and Peter G. Polson (1994). "Why is a Raven Like a Writing Desk?: Lessons in Interface Consistency and Analogical Reasoning from Two Cognitive Architectures." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '94. Boston, Massachusetts, USA: ACM, pp. 438–444. ISBN: 0-89791-650-6. DOI: 10.1145/191666.191816. URL: http://doi.acm.org/10.1145/191666.191816.

Rogers, Yvonne (2004). "New Theoretical Approaches for Human-Computer Interaction." In: *Annual Review of Information Science and Technology* 38.1, pp. 87–143. DOI: 10.1002/aris.1440380103. URL: https://doi.org/10.1002/aris.1440380103.

Ross, Brian H. and Thomas P. Moran (1983). "Remindings and Their Effects in Learning a Text Editor." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '83. Boston, Massachusetts, USA: Association for Computing Machinery, pp. 222–225. ISBN: 0897911210. DOI: 10.1145/800045.801615. URL: https://doi.org/10.1145/800045.801615.

Rosson, Mary Beth (1983). "Patterns of Experience in Text Editing." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '83. Boston, Massachusetts, USA: Association for Computing Machinery, pp. 171–175. ISBN: 0-89791-121-0. DOI: 10.1145/800045.801604. URL: https://doi.org/10.1145/800045.801604.

Schilit, Bill N., Gene Golovchinsky, and Morgan N. Price (1998). "Beyond Paper: Supporting Active Reading with Free Form Digital Ink Annotations." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '98. Los Angeles, California, USA: ACM Press/Addison-Wesley Publishing Co., pp. 249–256. ISBN: 0-201-30987-4. DOI: 10.1145/274644.274680. URL: https://doi.org/10.1145/274644.274680.

Sellen, Abigail and Richard Harper (1997). "Paper as an Analytic Resource for the Design of New Technologies." In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. CHI '97. Atlanta, Georgia, USA: Association for Computing Machinery, pp. 319–326. ISBN: 0-89791-802-9. DOI: 10.1145/258549.258780. URL: https://doi.org/10.1145/258549.258780.

Shneiderman, B. (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. URL: https://doi.org/10.1109/MC.1983.1654471.

Simon, Herbert A (1956). "Rational Choice and the Structure of the Environment." In: *Psychological Review* 63.2, p. 129. DOI: 10.1037/h0042769. URL: https://doi.org/10.1037/h0042769.

Singley, Mark K. and John R. Anderson (1987). "A Keystroke Analysis of Learning and Transfer in Text Editing." In: *Human–Computer Interaction* 3.3, pp. 223–274. DOI: 10.1207/s15327051hci0303_2. URL: https://doi.org/10.1207/s15327051hci0303_2.

Srgaard, Pål and Tone Irene Sandahl (1997). "Problems with Styles InWord Processing: AWeak Foundation for Electronic Publishing with SGML." In: *Proceedings of the 30th Hawaii International Conference on System Sciences: Digital Documents - Volume 6*. HICSS '97. USA: IEEE Computer Society, p. 137. ISBN: 0818677430. DOI: 10.1109/HICSS.1997.665495. URL: https://doi.org/10.1109/HICSS.1997.665495.

St Amant, Robert and Thomas E. Horton (2008). "Revisiting The Definition of Animal Tool Use." In: *Animal Behaviour* 75.4, pp. 1199–1208. DOI: 10.1016/j.anbehav.2007.09.028. URL: https://doi.org/10.1016/j.anbehav.2007.09.028.

Stylos, Jeffrey, Brad A. Myers, and Andrew Faulring (2004). "Citrine: Providing Intelligent Copy-and-Paste." In: *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*. UIST '04. Santa Fe, NM, USA: Association for Computing Machinery, pp. 185–188. ISBN: 1581139578. DOI: 10.1145/1029632.1029665. URL: https://doi.org/10.1145/1029632.1029665.

Teevan, Jaime, Shamsi T. Iqbal, and Curtis von Veh (2016). "Supporting Collaborative Writing with Microtasks." In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: Association for Computing Machinery, pp. 2657–2668. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858108. URL: https://doi.org/10.1145/2858036.2858108.

Tetzlaff, Linda (1986). "Transfer of Learning: Beyond Common Elements." In: *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*. CHI '87. Toronto, Ontario, Canada: Association for Computing Machinery, pp. 205–210. ISBN: 0897912136. DOI: 10.1145/29933.275631. URL: https://doi.org/10.1145/29933.275631.

Toomim, Michael, Andrew Begel, and Susan L. Graham (2004). "Managing Duplicated Code with Linked Editing." In: *Proceedings of the 2004 IEEE Symposium on Visual Languages - Human Centric Computing*. VLHCC '04. Washington, DC, USA: IEEE Computer Society, pp. 173–180. ISBN: 0-7803-8696-5. DOI: 10.1109/VLHCC.2004.35. URL: https://doi.org/10.1109/VLHCC.2004.35.

Tyler, Sherman W., Steven Roth, and Timothy Post (1982). "The Acquisition of Text Editing Skills." In: *Proceedings of the 1982 Conference on Human Factors in Computing Systems*. CHI '82. Gaithersburg, Maryland, USA: Association for Computing Machinery, pp. 324–325. DOI: 10.1145/800049.801803. URL: https://doi.org/10.1145/800049.801803.

Van Dam, Andries (Feb. 1997). "Post-WIMP User Interfaces." In: *Commun. ACM* 40.2, pp. 63–67. ISSN: 0001–0782. DOI: 10.1145/253671.253708.

Vass, Marc, John M. Carroll, and Clifford A. Shaffer (2002). "Supporting Creativity in Problem Solving Environments." In: *Proceedings* *of the 4th Conference on Creativity & Cognition*. C&C '02. Loughborough, UK: Association for Computing Machinery, pp. 31–37. ISBN: 1581134657. DOI: 10.1145/581710.581717. URL: https://doi.org/10.1145/581710.581717.

Wexelblat, Alan and Pattie Maes (1999). "Footprints: History-rich Tools for Information Foraging." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '99. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, pp. 270–277. ISBN: 0-201-48559-1. DOI: 10.1145/302979.303060. URL: https://doi.org/10.1145/302979.303060.

Wiley, Jennifer (1998). "Expertise as Mental Set: The Effects of Domain Knowledge in Creative Problem Solving." In: *Memory & Cognition* 26.4, pp. 716–730. ISSN: 1532-5946. DOI: 10.3758/BF03211392. URL: https://doi.org/10.3758/BF03211392.

Yoon, Dongwook, Nicholas Chen, and François Guimbretière (2013). "TextTearing: Opening White Space for Digital Ink Annotation." In: *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. UIST '13. St. Andrews, Scotland, United Kingdom: Association for Computing Machinery, pp. 107–112. ISBN: 978-1-4503-2268-3. DOI: 10.1145/2501988.2502036. URL: https://doi.org/10.1145/2501988.2502036.

Yoon, Young Seok and Brad A. Myers (Aug. 2015). "Supporting Selective Undo in a Code Editor." In: *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. ICSE '15. Florence, Italy: IEEE Press, pp. 223–233. ISBN: 9781479919345. DOI: 10.1109/ICSE.2015.43. URL: https://doi.org/10.1109/ICSE.2015.43.

Zheng, Qixing, Kellogg Booth, and Joanna McGrenere (2006). "Co-Authoring with Structured Annotations." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '06. Montréal, Québec, Canada: Association for Computing Machinery, pp. 131–140. ISBN: 1595933727. DOI: 10.1145/1124772.1124794. URL: https://doi.org/10.1145/1124772.1124794.