

Interaction Knowledge: Understanding the ‘Mechanics’ of Digital Tools

Miguel A. Renom
Université Paris–Saclay, CNRS, Inria
Laboratoire Interdisciplinaire des
Sciences du Numérique (LISN)
Orsay, France
renom@lri.fr

Baptiste Caramiaux
Sorbonne Université, CNRS
Institut des Systèmes Intelligents et
de Robotique (ISIR)
Paris, France
baptiste.caramiaux@sorbonne-
universite.fr

Michel Beaudouin-Lafon
Université Paris–Saclay, CNRS, Inria
Laboratoire Interdisciplinaire des
Sciences du Numérique (LISN)
Orsay, France
mbl@lisn.fr

ABSTRACT

User interfaces typically feature tools to act on objects and rely on the ability of users to *discover* or *learn* how to interact with them. Previous work in HCI has used the Theory of Affordances to explain how users understand the possibilities for action in digital environments. A complementary theory from cognitive neuroscience, Technical Reasoning, posits that users accumulate abstract knowledge of object properties and technical principles known as *mechanical knowledge*, essential in tool use. Drawing from this theory, we introduce *interaction knowledge* as the “mechanical” knowledge of digital environments. We provide evidence of its relevance by reporting on an experiment where participants performed tasks in a digital environment with ambiguous possibilities for interaction. We analyze how interaction knowledge was transferred across two digital domains, text editing and graphical editing, and conclude that interaction knowledge models an essential type of knowledge for interacting in the digital world.

CCS CONCEPTS

• **Human-centered computing** → **HCI theory, concepts and models.**

KEYWORDS

learnability, discoverability, tool use, mechanical knowledge, technical reasoning

ACM Reference Format:

Miguel A. Renom, Baptiste Caramiaux, and Michel Beaudouin-Lafon. 2023. Interaction Knowledge: Understanding the ‘Mechanics’ of Digital Tools. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI ’23)*, April 23–28, 2023, Hamburg, Germany. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3544548.3581246>

1 INTRODUCTION

User interfaces usually provide digital tools that mediate users’ actions on objects [10], e.g., the styling tools (bold, italics) for formatting text in a word processor. These tools are often designed to work with a specific digital object type, e.g., a color palette for

text, a different color palette for shapes and yet another one for tables. Therefore, users need to *discover* what tools are available for a particular task and *learn* [33] their nuances before they can put them to use on objects. Designers try to facilitate learning and discovery by using *signifiers* [31], which rely on the knowledge that users already have or can readily transfer to a digital environment. A similar approach is the use of metaphors of physical objects [11] — such as the well-known “desktop metaphor” pioneered by the user interface of the Xerox Star [24] — with the expectation that users will transfer knowledge from the physical world. These cues are challenging to design [33] and can lead to misinterpretations or “mismatches” [11] with the actual *possibilities for action* that the interface offers, e.g., believing that one could *copy and paste* a window to duplicate it. Moreover, humans are able to use physical objects in ways beyond those for which they were designed, e.g., using the surface of a physical desk for rolling dough or the trash can as a door stop, whereas their digital counterparts are mostly limited in their versatility by design.

Osiurak et al. [35] posit that, besides knowledge of how to manipulate objects, human tool use relies on the ability to perform *technical reasoning*. Technical reasoning is based on *mechanical knowledge* [35], i.e., abstract knowledge of object properties and technical principles that are used to run “mental simulations” for determining the appropriate interactions among tools and objects to transform those objects. For example, mechanical knowledge would allow a user to tell that a paintbrush can be used to apply paint on a novel surface, just from having incorporated a basic understanding of the interactions between the paint, the paintbrush and the surface, i.e., paint adheres to rough surfaces and the brush transfers the excess liquid to another surface. Moreover, technical reasoning based on mechanical knowledge explains the use of objects beyond the scope of their design, i.e., for unusual uses [35] such as using the paintbrush for sweeping.

Previous work in HCI [37] has found that, similar to what happens with physical objects, users can make unusual uses of digital tools, suggesting that such behavior can be modeled by a technical reasoning process. More generally, computer users have acquired a series of principles about user interfaces that they transfer to other interfaces, resembling the transfer of mechanical knowledge. For example, both beginner and savvy computer users have expectations about digital text input, i.e., what happens when we insert a character, delete it, select it, etc. Arguably, these expectations shape the knowledge that users transfer from past experience when confronted with a novel interface. However, while previous work

in HCI has focused on the transfer of knowledge from our physical reality into digital environments, we are not aware of work that has characterized our specific knowledge about the digital world.

Inspired by the concept of mechanical knowledge from technical reasoning, we introduce the concept of *interaction knowledge* as abstract knowledge about digital tools and objects, and about the possibilities for interaction among them. With mechanical knowledge, tool use is driven by abstract knowledge of physics principles affecting physical objects; with interaction knowledge, *digital* tool use is driven by knowledge of the principles that govern digital environments. In other words, interaction knowledge is the knowledge used to perform technical reasoning in the digital world. To explore this concept, we study two expressions of interaction knowledge using a novel digital environment and an experimental protocol that investigates the effect of priming the participants' interaction knowledge of WIMP interfaces.

After reviewing related work and introducing the concept of interaction knowledge, we describe the experiment and report on the results. We conclude with a discussion on the implications of this work and avenues for future work.

2 RELATED WORK

We review literature about how users accumulate and transfer knowledge about interfaces, focusing on how to make sense of the possibilities for action with tools in the digital world using affordances, signifiers and past experience. We also review previous work on analogical reasoning and technical reasoning, which ground our work on interaction knowledge.

2.1 Designing Affordances

The Theory of Affordances [17] posits that animals can infer the possibilities for action with physical objects in relation to their body capabilities, i.e., discover *affordances* of the environment. Gibson [17] describes the perception of affordances as a mechanism directly connecting perception and action, i.e., involving no conscious effort on the part of the individual, such as when a human correctly uses a hammer without any apparent knowledge of its function. However, while affordances exist whether they are perceived or not, their perception relies on the premise that objects possess salient features that users can perceive [35]. Since their introduction to HCI [30], affordances have been the subject of numerous interpretations about their meaning and uses in the field [29]. In this regard, Kaptelinin and Nardi [25] point out the challenges raised by modern technology to this theory, even beyond digital environments. As a matter of fact, modern tools typically lack salient connections between their controls and the actuators that carry out the corresponding actions. For example, a power drill may be activated by a trigger that is internally connected to the drill bit, but such connection may not be visible to users as it is hidden under the drill's plastic case. Furthermore, Osiurak et al. [35] argue that tool use "*requires more than the mere perception of affordances provided by tools*" because, in addition to being able to manipulate them, users need to understand how a tool interacts with the target object to reach its goal state, e.g., whether the power drill can make a hole in a given wall.

In digital environments, affordances are frequently brought up to refer to the functions made available to users [29], e.g., changing the

color of text or moving up a page. These may be accessible through widgets such as buttons, palettes or scrollbars. However, digital objects and tools are not necessarily explicit about their possibilities for action, i.e., their affordances are not readily perceivable. For example, Microsoft Word provides different color palettes for text, highlight and shape color, none of which are interchangeable, and Adobe Photoshop offers an *Eraser* tool that can only erase pixel-based objects, despite the fact that vector-based objects are made of paths that are also erasable. In both examples, users are required to know whether the tool will work on a given object, besides knowing how to operate it.¹ Therefore, tool-mediated interactions — both in physical and digital contexts — require knowledge of interactions between objects beyond the user's ability to manipulate them, which affordance perception cannot readily explain.

2.2 Signifiers & Cultural Conventions

Norman [31] coined the term *signifier* to refer to the cues provided to users for "*communicating how to use the design*" [33]. Although the term has its origins in semiotics, where it is used to refer to icons, indexes and symbols [15], Norman explicitly differentiates it from this original meaning and introduces it as a communicative property of design elements. For example, the looks of a button in a user interface may *signify* that it affords clicking or tapping. Signifiers — in Norman's sense — should also help users differentiate among the possibilities for action within interface elements, so as to spot those that are relevant to the users' intention [33]. For example, a dialog box can present many buttons that afford clicking, but only one may result in closing the dialog. In this case, a designer may use additional signifiers to reveal each button's affordance besides clicking, e.g., adding an *OK* label or relying on learned conventions [33] such as a cross for a button that closes a window.

The reliance of signifiers on cultural conventions resonates with Kaptelinin and Nardi's account of the effect of culture on computer-mediated activity [25], extending it to the use of modern technology. Knowing that a trigger is what activates a power tool (even though such a connection is not salient) and that drills can be used to pierce concrete walls is a form of cultural knowledge about technology. Nevertheless, novel uses of a tool, e.g., the first use of a power drill as a screwdriver, cannot be explained by cultural background alone. Although semiotic approaches — such as the use of icons in digital environments [3] — could serve to indicate possible uses of a tool, e.g., using an "A" in a color palette button to signal that it changes the color of any text-based object, the design of icons has its limits in visual arts and culture themselves. Furthermore, while signifiers signal knowledge for action in a digital environment, they do not model such knowledge *per se* or its use in a cognitive process.

2.3 Using Knowledge from Past Experience

Besides our perceptual and our cultural knowledge, we also accumulate knowledge from our observations of and interactions with the physical world. Computer technology has capitalized on these abilities by enabling, e.g., touch-based input, free-hand gestures [4] and tangible interactions [41]. This has led to user interfaces that take advantage of our bodily capabilities, such as pinching an image to

¹ Scrollbars, on the other hand, communicate their possibility for action more effectively as they are attached to the window that they affect.

zoom it or playing games using one’s whole body. These interfaces are known as *Natural User Interfaces* (NUIs) [32, 42], in reference to their closeness with “natural” interactions in the physical world.

Several interaction models for NUIs have been introduced that are based on our understanding of the physical world. Reality-based Interaction [22] describes an interaction style based on concepts of naïve physics that aims to make HCI “*more like interacting with the real, non-digital world*”. For example, interface objects may incorporate notions such as friction or gravity. Blended Interaction [23] extends the notions of the physical world to incorporate knowledge acquired when interacting with digital artifacts, thus forming *blends* between the physical and digital realities. For example, pinch-to-zoom is a well-established concept in touch-based interfaces that we rarely encounter in the physical world.

NUIs thus rely on our existing knowledge of the world, physical and/or digital. The nature of this knowledge relates to the notion of intuition. Blackler et al. [8] investigate intuition as the basis for designing interfaces that build on users’ experience, observing that past experience is a defining factor for creating intuitive interfaces. Hurtienne and Israel [21] draw on previous taxonomies of intuitiveness to propose their own for tangible user interfaces, focusing on *image schemas*, which describe how we understand the world, and their *metaphorical extensions*, which describe how that understanding can be transferred to other situations. The implication is that users should be able to find similarities between situations that *prime* the appropriate knowledge acquired in the past.

While these models account for the users’ ability to perceive similarities and make sense of interactions, they do not address how this knowledge comes to be used in novel ways. Hence, past experience alone is not sufficient to account for unusual tool use.

2.4 Analogical Reasoning

In line with intuitiveness, humans possess the ability to reason analogically [18] by identifying similarities between current and past problems, thus being able to apply “old” solutions to new problems. Such reasoning is leveraged in interfaces that prompt users to discover their affordances [11] through analogies with previously known situations. For example, the desktop metaphor [24] relies on the analogy of a physical office, with icons representing files, folders, etc., whose possibilities for interaction can be inferred from those occurring in the physical world, e.g., documents can be *moved* into folders. The same applies to the representations of digital tools that rely on analogies with objects of the physical world to infer their uses, e.g., using *ink wells* [9] for coloring.

Learning by analogy has been studied early on in HCI with text editing environments [26, 36, 39]. Rieman et al. [38] argue that there should exist a mapping between the contexts of two digital environments in order for users to be able to construct sound analogies, e.g., transferring towards a new text editing environment. However, certain affordances can be abstracted beyond a domain. For example, users can copy and paste across a multiplicity of contexts where selection is possible. Additionally, analogies between similar contexts can lead to negative transfer, e.g., the different ways in which the Tab key works between code editors, word processors, unformatted text editors, etc.

Previous work in cognitive neuroscience [35] argues that the ability to perform analogical reasoning is essential for using novel physical objects, prompting us to look for a similar process occurring in digital tool use.

2.5 Technical Reasoning

Osiurak et al. [35] posit that humans manage to use tools based on their ability to perform *technical reasoning*, a cognitive process based on analogical reasoning from experience with physical objects and their interactions. The authors postulate that humans accumulate *mechanical knowledge* from their interactions in the physical world, which takes the form of abstractions of the mechanical principles that are at play between object properties [35]. For example, one can reason about the *cutting* principle as the act of pressing a sharp, elongated and hard object, e.g., a knife, against a softer, firm target, e.g., an apple.

Technical reasoning relies on abstract knowledge because it does not require that the tool user recognizes the objective properties of the tool and the object to recall a previous interaction between them, but rather to make sense of how the properties in each object relate to each other according to the mechanical principle in question. In other words, technical reasoning is only needed when the task at hand presents a novel component, e.g., the first time that one uses a mug as a paper weight. This ability is especially useful in situations where a recognizable tool is not available, leading the user to resort to technical reasoning so as to find an appropriate object and mechanical principle to solve the problem, thus producing novel uses of objects. For example, one could arrive at the conclusion that a knife can be used as a screwdriver because of its shape and how it interacts with the screw’s head, even though it is not designed for that purpose. This hypothesis challenges the notion that human tool use originates in our use of *procedural knowledge* [35], i.e., the learned routines that are specific to the use of tools [2]. Instead, it offers a model somewhat *half-way* between declarative and procedural knowledge.

The technical reasoning hypothesis was recently introduced to HCI in a study [37] presenting evidence that users can perform technical reasoning to carry out unusual uses of *digital* tools. In the reported experiment, the authors observed participants performing unusual uses of digital tools and found evidence that they elicited knowledge of digital objects and tools, and of principles that describe the results of their interactions, which some of them could associate with past experience. Consequently, computer users seem to develop a sort of “mechanical” knowledge of the digital world. However, this work did not focus on the form that this knowledge takes for interactions in digital environments, and did not specifically address how unfamiliar interfaces bring participants to use their acquired knowledge of digital tools.

In summary, several theories have been used to model how users find the possibilities for action in user interfaces. Technical reasoning in particular defines a form of knowledge that enables tool users to discover how to carry out unusual uses of tools, namely, mechanical knowledge. Based on existing evidence that technical reasoning is at play when using digital environments, we seek to characterize a form of “mechanical” knowledge of the digital world that enables the reasoning process to take place in unfamiliar interfaces.

3 INTERACTION KNOWLEDGE

We define *interaction knowledge* as abstract knowledge of the possibilities for interaction in digital environments. Interaction knowledge forms the basis for technical reasoning in digital environments, in the same way that mechanical knowledge does in the physical world [35]. Interaction knowledge is abstract in the sense that, when facing a novel user interface, users perceive surface cues of possible interactions, such as typing when a cursor blinks, clicking a color swatch to apply color or knowing that one can minimize a window in a GUI. These abstractions originate in the principles that users learn through experience in both physical and digital environments.

Knowledge of both physical and digital environments has been shown to help users perform *analogical* reasoning about digital interfaces [11]. However, interaction knowledge is of particular interest for tasks that involve *technical* reasoning about interactions in digital environments, i.e., when facing situations that require novel uses of tools.

Since digital objects follow artificially designed “laws of information” rather than the laws of physics, technical reasoning in digital environments must rely at least partially on abstract knowledge specific to the digital world. For example, mechanical knowledge models our understanding of “naive physics” in abstractions such as the law of gravity, which we use spontaneously even though we are not necessarily able to put it into words. Similarly, interaction knowledge models abstract “laws” of digital environments, such as text being editable when it contains a blinking cursor or windows having the ability to be minimized, which hardly have an equivalent abstraction in the physical world. Mechanical knowledge and interaction knowledge may overlap in some of the abstractions that compose them, in particular when the digital world mimics the physical one, but there is no reason to believe that one can be subsumed by the other.

Analyzing user behavior in terms of the principles that users have acquired and that they perceive as applicable from surface cues opens the door to a more rigorous approach to interface design. While some of these principles are already at play in many interactions such as cursor-based text editing, window manipulation using their sides and corners, or selection highlighting that suggests copying and pasting, many interactions are still arbitrary and differ from one environment or one application to the next. Identifying and applying these principles universally will expand users’ ability to perform technical reasoning when facing a new interface or situation, e.g., by using a tool in an unexpected way. It will also inspire better design guidelines for learnability [19] and result in more “intuitive” and powerful interfaces by capitalizing on the users’ past knowledge [8]. Interaction knowledge can be a useful resource for models such as Blended Interaction [23], Reality-based Interaction [22] or Instrumental Interaction [5], in that it can provide the basis for more universal principles for the design of interfaces, which users can more readily apply when resorting to technical reasoning.

4 STUDY: INTERACTION KNOWLEDGE IN INTERFACE DISCOVERY

This section describes an experiment designed to explore the role of interaction knowledge in the discovery of an unfamiliar interface. Our approach consists of inferring the existence of interaction knowledge from the principles underlying the tools selected by the participants to interact with objects on the screen. More precisely, our goal is to observe how different interface cues affect the users’ strategies for manipulating digital objects and draw conclusions about the kind of interaction knowledge at play.

We created an experimental editor that supports a subset of common text- and graphics-oriented tools and used separate toolbars and different selection interactions as cues of the applicability of these tools. The editor displays a canvas containing words and emojis (Figure 1, left). At first, participants see the canvas and either a text- or graphics-oriented toolbar, but not both, and are asked to perform a selection task. Then both toolbars are made available and the participants are asked to complete a series of editing tasks.

We expect that the initial toolbar, acting as interface cue, will *prime* the participants’ knowledge of how to interact with digital objects as either text or graphics. We consider that a participant is primed with a given knowledge when they perform one or more actions that require such knowledge. In other words, by first exposing participants to a specific toolbar, we expect that they will respond by demonstrating interaction knowledge about objects according to the type of that toolbar, regardless of the availability of the other toolbar. As participants progress through tasks requiring more effort, we expect them to resort to technical reasoning to solve the tasks more effectively, thereby eliciting interaction knowledge primed by the tools that are available from both toolbars.

We summarize the questions addressed by this experiment in the following hypotheses:

- H1: (*Priming*) Interaction knowledge about text or graphics is primed by the type of toolbar that the participant first sees, i.e., a toolbar with text-oriented tools primes the selection and manipulation of objects as text, while a toolbar with graphics-oriented tools primes the selection and manipulation of objects as graphical shapes;
- H2: (*Reasoning*) When facing tasks that require more effort, participants resort to interaction knowledge about both text and graphics to exert technical reasoning, i.e., when both types of tools are available, the perceived effort required by the task will prompt the participant to use knowledge primed by the available tools to complete the task more efficiently.

4.1 Task

Participants interact with a content editor that displays monospaced text characters and emojis, organized in words and visual shapes on a canvas (Figure 1). This content editor supports interacting with elements both as if they were text and graphical objects. For example, a user can select the *Highlighter* tool in the top toolbar, which will switch the mouse cursor to an *I-beam*, letting her select sequences of characters and insert and delete text as if they were part of a text document. Conversely, selecting the *Pointer* tool in the left toolbar will switch the mouse cursor to an arrow, letting

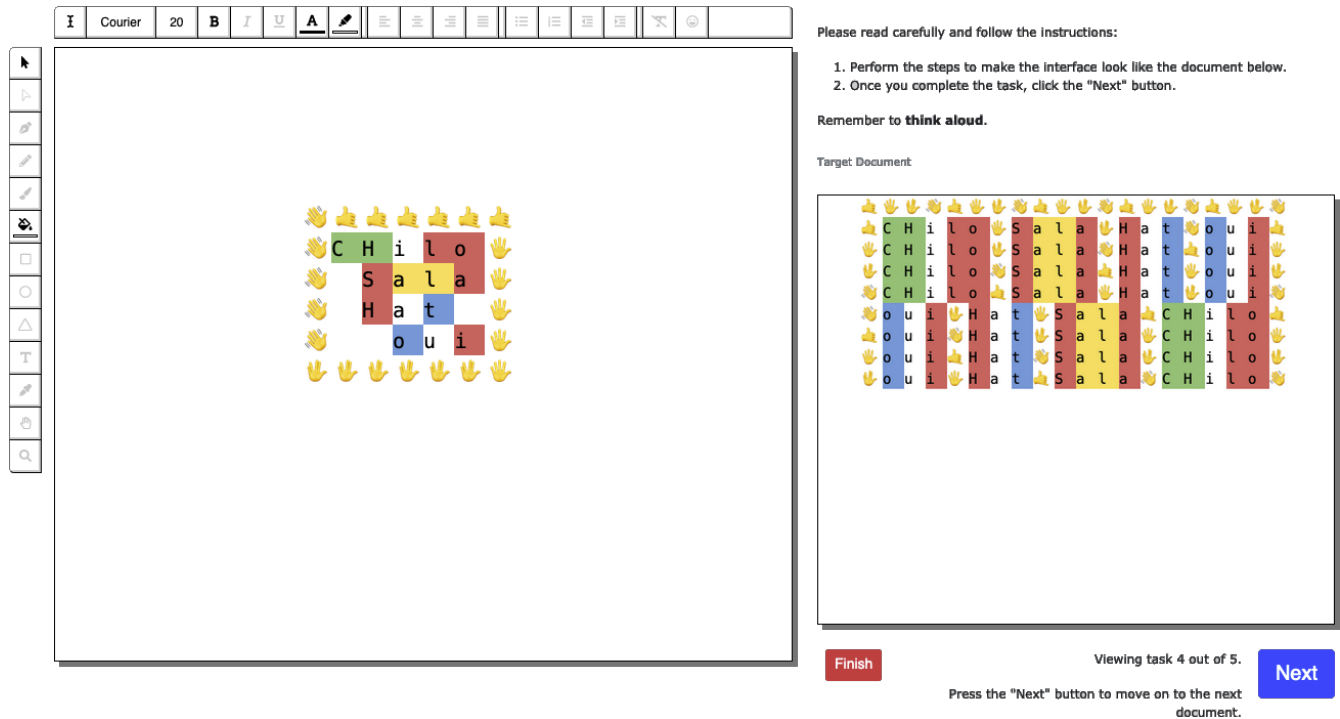


Figure 1. The full view of the experimental environment during Task 4. To reproduce the image on the right in optimal time, the participant should combine text and graphics tools.

her select shapes using rectangular selection and drag and drop objects as floating shapes on a canvas.

Participants are organized in three groups according to the initial cues that they receive: *Text Group*, *Graphics Group* and *Control Group*. Participants in the *Text Group* and the *Graphics Group* are asked to perform a selection of the objects in the canvas, i.e., they can only highlight text if they are in the *Text Group* or perform a rectangular selection or multiple object selection if they are in the *Graphics Group*. Participants in the *Control Group* are presented the canvas without any toolbar and are not asked to perform a selection. Then, all participants interact with a version of the editor that has both text and graphics toolbars enabled, and are asked to perform 5 tasks corresponding to incremental steps towards a goal state. These tasks can be carried out using either only graphics commands, only text commands or a mix of both types, which lets us evaluate the effect of the environment on the choice of tools that participants make to complete tasks.

To design the 5 tasks, we ran 11 pilot testing sessions with participants from both inside and outside our lab. We tested different object representations and layouts for the canvas' content, with the goal of inducing ambiguous interpretations about the appropriate interaction, i.e., text-based, graphics-based or other. We decided to use a mix of text characters arranged as words and emojis scattered across a grid so as to mix characteristics of both text and graphics environments, suggesting an ambiguous environment when looking at the canvas alone. This is based on the premise that participants are familiar with emojis being included in regular text as well as text being part of vector-based compositions. We chose the

tools made available in the toolbars based on two criteria: being recognizable from popular software, e.g., the pointer for moving graphical objects and the *I-beam* for typing text, and achieving similar visual results between text and graphics, e.g., the paint bucket and the text highlighter both change the background color property.

The first three tasks (Figure 3a, Figure 3b and Figure 3c) constitute small steps to familiarize the participant with the environment, such as finding out which tools and shortcuts can be used. The last two tasks (Figure 3d and Figure 3e), on the contrary, involve more effort to induce the need to devise strategies and find the tools that make them less cumbersome.

4.2 Participants

We recruited 37 computer users via calls for participation over email and social networks, as well as word of mouth from participants, until completing 12 participants for each of the three groups. Candidates were accepted if they self-reported themselves as knowledgeable about computers. We discarded data from 1 participant because of data inconsistencies found after the session². Of the remaining 36 participants, 17 self-reported as female and 19 as male. Participants reported on average between 11 and 20 years of experience with text editing, and between 5 and 10 years of experience with graphical editing. On average, the self-reported frequency of use of text editing software was “Almost daily use,” while for graphical editing software it was “A few times a month.”

²Data was stored in the participant's browser and was downloaded and later sent to the experimenter. For reasons unknown, this participant's file missed the entire action log corresponding to the last task.

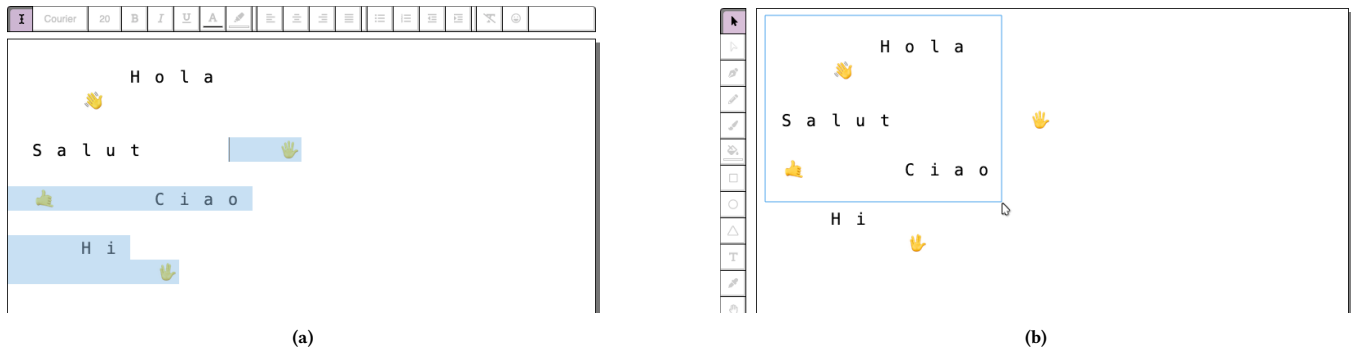


Figure 2. The editor in the second phase, when selecting elements for the first time, for the *Text Group* (a) and *Graphics Group* (b). The button corresponding to the selection tool is the only one activated in the toolbar and the participant can only perform a selection according to the tool. All other tools remain disabled until having to complete tasks (third phase).

4.3 Setup

We 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 common text- and graphics-editing commands. At the end of the session, the environment allowed downloading a JSON file containing the event logs and answers to the questionnaires, which the experimenter asked the participant to send via e-mail. The application scripts were hosted on a virtual server running on our lab's infrastructure.

When carrying out tasks, the interface (Figure 1) is comprised of a text-based toolbar at the top, a graphics-based toolbar on the left side and a rectangular canvas in the center. The interface layout borrows from popular text- and graphics-based editing environments. A number of tools are shown in a disabled state (grayed out), and are not actually implemented. These tools were included so as to make the toolbar consistent with those of familiar environments. Table 1 shows the tools required to complete the tasks. The text toolbar includes a *Text* tool that inserts a text cursor after the last element in the canvas (in top-to-bottom and left-to-right order) thus entering into *Text* mode, allowing the user to type characters as in a text editing environment. The *Highlighter* tool works in the same way as in Microsoft Word: if a selection exists before it is activated, it applies the current highlighting color to it; otherwise, it highlights any selection made with the cursor while the tool is active. Additional functions in the text toolbar resemble familiar text-oriented toolbars, and activate the *Text* mode when used: font family, font size, font style and text color. A separate panel was also added to the text toolbar to insert emoji icons at the location of the text cursor. The graphics toolbar includes a *Pointer* tool for selecting and dragging objects in the canvas, and a *Fill* tool to point and click at objects for changing their background color. Both tools put the editor into *Graphics* mode when active.

While in *Text* mode, the mouse cursor is displayed as an *I-beam* when moving across the canvas, except when the *Highlighter* tool is selected, in which case it shows the cursor corresponding to the highlighter. While in *Graphics* mode, the mouse cursor changes to that of the *Pointer* tool or a paint bucket for 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 so 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 of the environment and logging of user actions, as well as capturing more actions from the participants when they recover from mistakes.

In *Text* mode, typed characters are inserted sequentially as it would happen with a regular text editor. Insertions are wrapped at the right edge of the canvas at the character level. When in *Graphics* mode, characters can be dragged around inside and outside of the canvas. Dropping a character out of bounds will make it impossible to drag it back inside. When going from *Graphics* to *Text* mode, space and new line characters are added in front of visible characters as needed to preserve their positions and keep the behavior consistent with text input³. These characters become part of the document and are editable in *Graphics* mode. We made it possible to drag a character on top of another in *Graphics* mode. In such cases, the overlapping characters are considered at the same position in *Text* mode, meaning that they will behave as a single character, e.g., they get deleted as one character.

All elements in the canvas are padded to occupy a square slot in a grid. Dropping elements in *Graphics* mode adjusts them to the nearest slot. All elements are independent from each other, i.e. characters can be selected individually with either the *Pointer* tool or the text cursor, with the exception of overlapping characters in *Text* mode. In order to select more than one element, the user can select 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 comprised of characters arranged in words and emojis. Each goal state is depicted outside the editor interface (Figure 3) during the task. The initial state is always that of the previous goal state, regardless of whether the participant reproduced it in the previous task. This is so that every participant begins working on a given task under identical conditions. After completing a task, participants click a *Next* button

³Since we do not provide alignment tools, such as the ruler in Microsoft Word, it is not possible to have a character float in the middle of a page without inserting blank characters before it.

Table 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 graphical 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 graphics 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 graphical 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 graphics 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.

outside the editor interface that saves the action log associated with it. A *Finish* button next to the *Next* button allows to complete the session. If there are any remaining tasks, the session is considered abandoned and the data collected for that participant is discarded. This was not experienced during any of our sessions.

4.4 Procedure

We used a between-participant design with one factor (GROUP) controlling the type of toolbar and the selection type first presented to the participant. A session begins by sending the participant a unique URL corresponding to a unique Id. Using a video conferencing application with support for screen sharing, participants share a video stream of the browser window where they open the URL. Participants read a short introduction that indicates that they will use a novel digital environment without specifying its purpose.

After pressing a “Continue” button, participants observe the interface corresponding to their assigned group. All participants see the canvas with the same content. The *Text Group* also sees the text-oriented toolbar and the *Graphics Group* the graphics-oriented toolbar, while the *Control Group* does not see any toolbar. In the first phase, all participants are asked to describe how they would leave all the elements in the canvas in a selected state (without actually performing the selection). Next, in the second phase, participants in the *Text Group* and *Graphics Group* see the *Text* tool and the *Pointer* tool activated, respectively (Figure 2), without the possibility to switch to another tool nor deactivate the current one. Participants in both groups are asked to perform the steps to leave all the elements in the canvas in a selected state. Participants in the *Control Group* skip these two phases and proceed directly to the next phase.

In the third phase, all participants are presented with a fully interactive version of the editor with both its graphics and text toolbars on the top and left sides of the canvas respectively, and all implemented features in an enabled state (Figure 1). They then use the keyboard and mouse to complete 5 tasks requiring them to replicate a series of images displayed in a panel on the right. When they are done reproducing the image, they press a “Next” button to load the image corresponding to the next task. At the beginning of each task, participants have to select a tool instead of continuing with the last tool used in the previous task. Participants are asked

to think aloud [20] 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 feel stuck can be assisted. For example, if a participant attempts to select multiple elements by keeping the Control key pressed, we indicate that this is possible with the Shift key. We also give confirmation when a participant expresses that a function is not present. For example, if a participant attempts to execute an “Undo” command, we indicate that history commands are not supported and that fixing mistakes requires reversing the steps manually or refreshing the browser to start over with the task.

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, participants are 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 them. Participants can end the session at any point by pressing a “Finish” button on the side of the editor (Figure 1) 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 graphics editing environments, with demographic items at the end. Participants then download a file containing the action logs from all the tasks and the answers to the questionnaire, and transfer it to the experimenter.

4.5 Data Collection

We recorded audio from the call and video from the participants’ screen. We took notes of the participants’ responses about the type of steps they took to select content. During the task performance, we collected action logs including keystrokes of character and meta keys, toolbar interactions, tool commands on the canvas objects and clipboard commands. Every action includes a timestamp. We also took notes of the participants’ verbal protocol during the tasks.

We collected answers to the end questionnaire. The questionnaire was divided into three parts: daily experience with text and graphics editing software, experience with the experimental editor—assessing the use of 3 tools that were relevant to the task—, their perception of using text or graphics editing approaches, a self-reported measure of the prevalence of one approach over the other,

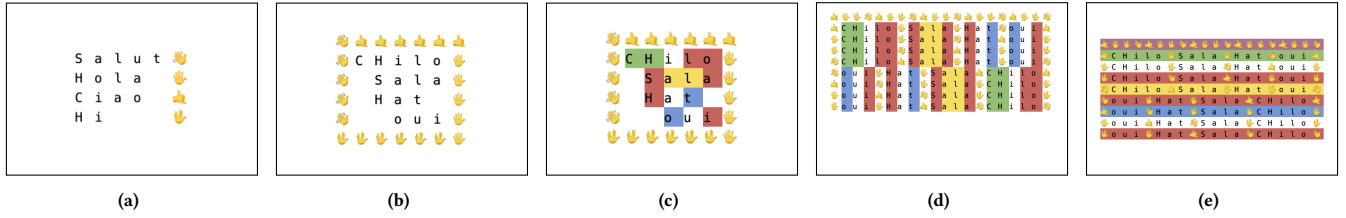


Figure 3. Screenshots of the tasks during a session.

knowledge of functions from other software that was used for completing the tasks, and demographic questions. This assessment was used in connection with the notes from the verbal protocol and action logs of their performance during the tasks.

All data were referenced by participant number. The experimental design and data collection were approved by our Institutional Review Board.

4.6 Data Analysis

We analyzed the selection techniques that the participants used (*SELTECH*) and classified them between *Text Selection* for text-based techniques, *Graphics Selection* for graphics-based techniques and *Other Selection* for alternative responses. This was coded based on their answer during the first phase of the procedure, together with the execution of the action during the second phase. We performed independence tests to determine whether the participants' GROUP (*Text Group*, *Graphics Group* and *Control Group*) associated with a particular *SELTECH* (H1).

Using the action logs of the participants' tasks, we analyzed the number of command executions by their type—"text" or "graphics"—and designated each task's *APPROACH* as based on *Graphics-only*, *Text-only* or *Mixed* commands. We considered an approach to be *Mixed* when it more than 5% of the total number of its commands were of a secondary type, so as to discard unintentional or playful uses. We did not analyze data from commands that do not modify objects or the canvas, e.g., selection or change of tool.

Next, we performed independence tests of GROUP and *APPROACH* to determine whether the interface cues associated with particular approaches to complete the tasks (H2), i.e., whether participants used one type of tool or both to solve a task. This test was carried out both by aggregating the approaches from all tasks as well as by testing individually for each task. We also tested correlations between the reported frequency of use of text- and graphics-oriented software and the number of text- and graphics-oriented actions during the tasks, respectively. Finally, we used notes from the verbal protocol and answers to the questionnaire by each participant to complement the analysis of how past experience affected the approach used to complete the tasks.

5 RESULTS

We were interested in the priming effect of the toolbar layout on the selection interaction and on the choice of tools for accomplishing the tasks. For this purpose, we analyzed the participants' description of the selection technique after their initial encounter with the environment, as well as their action logs during the execution of the tasks. We complemented our results with observations gathered

Table 2. Count of selection technique class by group.

GROUP	Selection Technique		
	Text	Graphics	Other
<i>Text Group</i>	10	2	0
<i>Graphics Group</i>	1	11	0
<i>Control Group</i>	9	3	0

from our notes of the participants' verbal protocol and answers to questionnaires (see supplemental material for additional results).

5.1 Toolbars Primed the Selection Technique

We counted the responses of each type from the description of the steps to select all the elements in the canvas, according to the GROUP. Among those in the *Text Group*, 10 (83%) described a *Text Selection* technique and 2 described a *Graphics Selection* technique; among those in the *Graphics Group*, 11 (92%) described a *Graphics Selection* technique and 1 described a *Text Selection* technique; and among those in the *Control Group*, 9 (75%) described a *Text Selection* technique and the remaining 3 (25%) described a *Graphics Selection* technique. Some participants found it difficult to identify the environment that was being presented, yet were able to describe a proper selection technique. For example, **P5** (*Control Group*) described the environment as: "maybe a chat room" before proceeding to describe a text selection technique. Therefore, all participants described either a text- or graphics-based selection technique.

Table 2 shows the counts for each GROUP. All expected frequencies are above 5. A Chi-square test of independence⁴ shows a statistically significant relation between GROUP 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 show significant differences in *SELTECH* between the *Text Group* and the *Graphics Group* ($p = .0019$) and the *Graphics Group* and the *Control Group* ($p = .0055$) but not between the *Text Group* and the *Control Group* ($p = 1.000$) — all p-values corrected with Bonferroni's technique for 3 comparisons.

These results suggest that the presence of either a text or graphics toolbar had an effect on the participants' decision to perform a text- or graphics-based selection of the objects respectively, thus supporting our priming hypothesis (H1) that the participants' knowledge of how an interaction must be carried out was primed by the type of the toolbar. However, when observing the differences between

⁴Following [1], we 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.

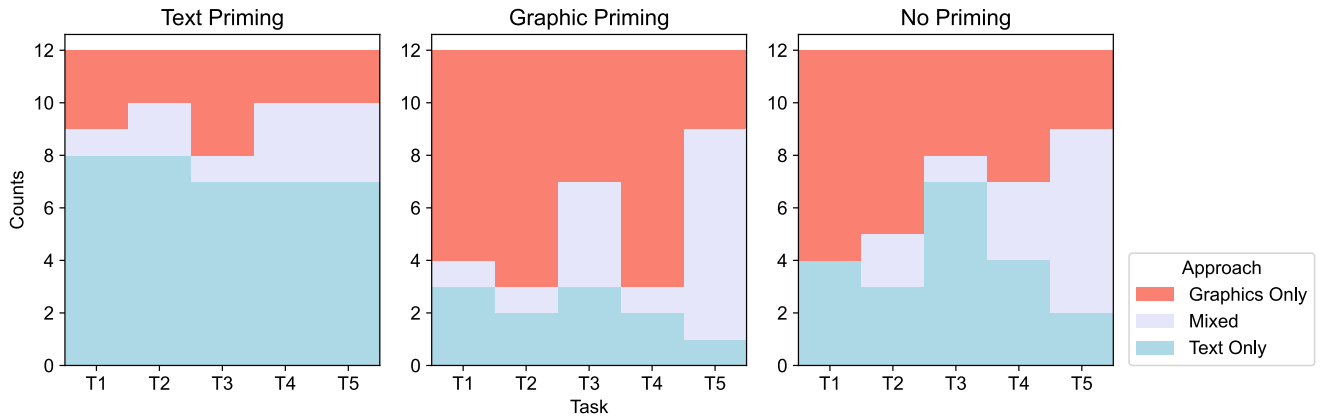


Figure 4. Frequencies of each approach among participants across tasks for each priming group (see Table 3 for the numbers).

Table 3. Count of Text-only (T), Graphics-only (G) and Mixed (M) approaches by task (see Figure 4 for a visualization).

GROUP	Task Number														
	1			2			3			4			5		
	APPROACH														
	G	M	T	G	M	T	G	M	T	G	M	T	G	M	T
Text Group	3	1	8	2	2	8	4	1	7	2	3	7	2	3	7
Graphics Group	8	1	3	9	1	2	5	4	3	9	1	2	3	8	1
Control Group	8	0	4	7	2	3	4	1	7	5	3	4	3	7	2

toolbar conditions, we found that performing a graphical selection associated with displaying the graphics toolbar, while there was no significant difference in selection technique between displaying the text toolbar and not displaying any toolbar. In other words, both in the absence of toolbars and the presence of only the text toolbar, most participants assumed that a text selection technique would be appropriate. This could be due to bias caused by the contents of the canvas resembling text more than graphics content.

5.2 Interaction Cues Primed the Choice of Tools

We analyzed the action logs from the five editing tasks in the third phase of the session to extract the types of tools per task and participant. We designated an *APPROACH* for each task based on the tool types used, classified as *Text-only* when tools were primarily text-oriented, *Graphics-only* when tools were primarily graphics-oriented and *Mixed* when tools included a significant share of both types⁵. Table 3 shows the overall counts of the approaches by task and GROUP. Figure 4 illustrates the counts in Table 3, showing that priming with a text toolbar and text interaction (*Text Group*) associated with more participants using text approaches on every task. However, there is no such visible evidence of a difference in the use of tool types between the *Graphics Group* and the *Control Group*, i.e., when users had graphics cues vs. no interaction cues before performing the tasks. This seems to contradict our previous assessment that participants in the *Control Group* may have perceived the content as text more often than as graphics.

⁵5% or more of the tools used being of the secondary type.

5.2.1 Priming Worked for Aggregated but not for Individual Tasks. We wanted to confirm our visual assessment of the difference in *APPROACH* between the *Text Group* and the *Graphics Group*. We tabulated the data from the five tasks and aggregated it for the two separate groups, the *Text Group* and the *Graphics Group* (60 observations per group, i.e., 12 participants performing 5 tasks each). All the expected values in the tabulated frequencies are above 5. The chi-square test of independence shows that there is a statistically significant association between the participant’s group 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, the *Text Group* had a large deviation in *APPROACH* (values > 2 for small tables [1]), reflected by more *Text-only* approaches (4.84) and fewer *Graphics-only* approaches, while the opposite occurred for the *Graphics Group*, for which the deviation was reflected by more *Graphics-only* approaches (3.92) and fewer *Text-only* approaches. This suggests that, in general, participants preferred using tools of a type according to the priming of their group.

Having observed an effect when aggregating all five tasks, we then analyzed the data on a task-by-task basis. After tabulating the results for each of the five tasks separately (12 observations per group), the expected values were under 5 in all tables, making the chi-square test of independence unsuitable. The Freeman-Halton’s extension of Fisher’s exact test [16] shows that in tasks 2 ($p = .013$) and 4 ($p = 0.019$) there were statistically significant associations between the participant’s group and the approach used. An analysis of the standard residuals for these two tasks shows that the deviations occurred in the same directions as for the overall case but with a less pronounced effect, due to the fact that residuals for individual tasks were smaller than for the overall tasks. Therefore, while tasks 2 and 4 followed the general pattern of using tools associated with the priming of the participants’ group, tasks 1, 3 and 5 show no significant association with a tool type, suggesting that participants sought new tool types during their execution, in line with our reasoning hypothesis (H2).

5.2.2 Participants in the Control Group were Not Primed. We analyzed the effect of priming participants on *APPROACH* by comparing

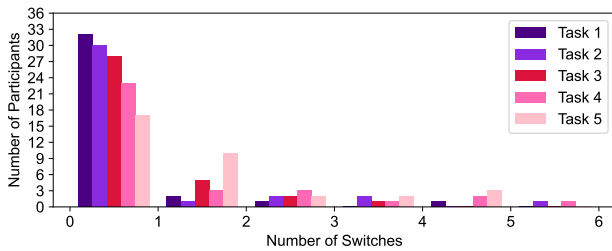


Figure 5. Histogram of the number of participants who switched tool types by the number of times and task. This chart leaves out three participants in Task 4 who switched tool types 8, 16 and 18 times respectively, and two participant in Task 5 who switched 7 and 8 times respectively.

data from the *Text Group* and *Graphics Group* combined (primed group) with the *Control Group* (non-primed group). We combined the aggregated observations of the five tasks of the *Text Group* and *Graphics Group* in one category (primed, 120 observations), while keeping those of the *Control Group* in another (non-primed, 60 observations). All expected values in the table are above 5. A chi-square test shows no statistically significant association between the primed status and the approach used by participants ($p = .582$). The same result obtains with the Freeman-Halton extension to Fisher's exact test for each task ($p > .05$ in all tests). This suggests that participants in the control group (12) did not use significantly more text, graphics or mixed approaches than participants who were intentionally primed (24), thus supporting the fact that the control group was not primed by our interaction cues.

To sum up these results, for tasks 2 and 4, as well as for all tasks aggregated, groups primed with text cues had an increased use of text-only approaches, while groups primed with graphics cues had an increased use of graphics-only approaches, in line with our priming hypothesis ($H1$)⁶. However, the differences between tasks 2 and 4 on the one hand and tasks 1, 3 and 5 on the other can be attributed to the participants' increased perceived effort⁷ of the latter, prompting them to exert technical reasoning and thus resorting to their interaction knowledge about available tools to complete them, thereby supporting our reasoning hypothesis ($H2$) for these three tasks. Moreover, the approaches used by the control group were not significantly different from those used by the other two groups combined, suggesting that when not primed by our toolbars, participants did not show a particular preference for one approach over the other. This supports the idea that participants in the control group were not exposed to interaction cues that could disambiguate the type of editing that they could carry out, and therefore did not elicit a priming effect when performing the tasks.

⁶We repeated these two 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 the two conditions and *APPROACH* ($p < .05$) for all thresholds. For individual tasks, we found similar significant results of the Fisher's exact test for all thresholds except for 49% where an additional significant association between the two conditions and *APPROACH* was found in task 1.

⁷The difficulty of the task was first associated with the number of objects to change, but was then revised after observing the participants.

5.3 Most Participants Worked on One Representation at a Time

Most participants did not switch tool type (Figure 5) during each task, i.e. most of them chose a representation at a given point in the task, e.g., edit as text, and used tools tied to it exclusively until reaching the goal state. On the other hand, two extreme cases changed tool type 16 and 17 times, both during task 4. For tasks 1, 2 and 3, most participants stayed on the same tool type from beginning to end. One possibility is that they assumed that the first representation that they found was the only possible one.

Another explanation lies in the fact that tasks 1, 2 and 3 were relatively easy to complete with either text-only or graphics-only tools, whereas tasks 4 and 5 were cumbersome to complete without using tools of both types. Figure 5 shows that task 5 has the most number of participants switching tool types once, consistent with our observation of participants using the *Fill* tool in the beginning and later finding the *Highlighter* tool more convenient to carry out the task. In summary, with the exception of task 5, most participants chose to stick to one representation, suggesting that they assumed that only one representation was correct within a task.

5.4 Some Participants' Approaches were based on Familiarity or Convenience

When asked about their motivations, **P1**, **P8**, and **P15** (who used text approaches), and **P9**, **P13**, **P16** and **P29** (who used graphics approaches) stated that the interaction cues made them think of the problem according to the environment, in line with our priming hypothesis ($H1$). However, some participants did not seem to be influenced by priming, in line with our reasoning hypothesis ($H2$).

For example, **P31** (*Graphics Group*), was the only participant primed with graphics cues who performed all tasks exclusively with text-based tools, that is, the exact opposite of the intended priming. When asked about this, he said: "I mostly use text editors [in my daily life], rather than graphics editors. That's why I'm more comfortable with text and use [it] whenever I can." In particular, during task 3, **P31** sought a tool to change the background color of text characters, stating: "I will use [the highlighter tool] because I think [the fill tool] fills the [graphics] shapes," thus identifying the task as text editing and then choosing tools according to what seemed appropriate.

Conversely, **P17** (*Text Group*) was the only participant primed with text cues who performed all tasks exclusively with graphics-based tools. When queried about 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 (*Graphics Group*) was an example in between these two. He performed tasks 1, 2, 4 and 5 using exclusively graphics tools and decided to explore an alternative way in task 3, using the *Highlighter* tool after having tested the *Fill* tool. However, although he had discovered and used the *Highlighter* tool to complete task 3, he carried out task 5 using exclusively the *Fill* tool, thus completing the task rather inefficiently (because it required pointing and clicking at every single character). When asked about this, he said: "[I] found a way that worked and stuck with it, plus, I have a big bias towards graphical design."

We found significant positive correlations between the reported frequency of use of graphics-oriented software and the number of actions executed using graphics tools in tasks 1 ($r = .49, p = .002$) and 2 ($r = .60, p < .001$), and for all tasks aggregated ($r = .42, p = .010$), but not for tasks 3, 4 and 5 ($p > .05$). Correlations between frequency of use of text-oriented software and the use of text-oriented tools were not significant ($p > .05$). Despite the significant effect of interaction cues on the approaches in general, some participants ignored these cues and used the approach that seemed familiar or convenient to them, suggesting that past experience had a significant effect on the interaction knowledge used.

5.5 Some Mixed Approaches Originated in Seeking the Right One

Some participants expressed having used tools for their perceived efficiency. For example, **P27** (*Graphics Group*) recognized the possibility to use text tools in addition to graphics tools, but preferred using the latter because “*It was just easier to stay within the graphics mode as it allowed for easier copy and paste rather than moving the hand between my trackpad and keyboard all the time.*”

P23 (*Text Group*) saw the tasks as “games,” 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 based on her experience with multiple selection, which was not supported in our environment. 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 (when switching back to a graphic tool). She performed a second text selection revealing her thought process about the environment: “*this is just text,*” after which she used the *Highlighter* tool for the remaining lines, resulting in a mixed approach.

P23 is representative of other participants such as **P10**, **P20**, **P21** and **P26** who were concerned about finding the “right” approach and explored the interface by focusing on the tools that would “solve” the task, apparently ignoring which actions were supported by the objects. This suggests that these participants paid more attention to the tools and their effects, rather than on whether the objects were supported by the tools.

5.6 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 came to graphics editing strategies, 13 mentioned Microsoft Paint, 7 Adobe Illustrator, 7 Adobe Photoshop and 5 Microsoft PowerPoint. In particular, **P7** recalled Adobe PageMaker at the end of the session because of its mixed text- and graphics-oriented tasks—referring to the *Pointer* tool as the lead cue—, although this participant used only text-based tools.

P18 was quick to associate Microsoft Word with the use of the *Highlighter* tool 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 (emphasis ours): “*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 did not mention the source of their knowledge about how to operate with text, except mentioning the highlighting and clipboard commands. However, it is evident that they did not spend much time understanding how to operate with text despite it being a novel environment, suggesting that procedural knowledge from past experience was brought in. In the next section, we discuss these results in terms of interaction knowledge.

6 DISCUSSION

Our results suggest that the experimental environment first primed the participants with their interaction knowledge of text or graphical objects. This is reflected in their ability to predict a correct selection technique simply by observing the interface, and later by their overall preference for tool types that match the priming of the first toolbar that they were presented. However, some participants mixed tools of both types and some did not even use any tools related to the priming that we intended, supporting the hypothesis that they exerted technical reasoning to solve those tasks and used interaction knowledge for both text and graphical objects. The performance of participants in the control group suggests that the interface offered no particular cues about the type of tools that could be used with the objects, evidenced by the absence of preference for a tool type in this group. In this section, we discuss the limitations of our work and implications of its results for the transfer of interaction knowledge in digital environments.

6.1 Limitations

The purpose of this work was to find evidence that participants elicit interaction knowledge about text and graphics in a simple WIMP interface. In this regard, it is but a first step to validate interaction knowledge as a concept in HCI.

As with most controlled experiments, the need to control for sources of variability resulted in a design that does not represent a standard setting, where users would find assistance by searching on the Internet or asking peers. The experimental setting may also have resulted in biases in the participants’ behavior such as trying to produce the result that they think is correct rather than what they would normally do, or acting differently because they are being observed (Hawthorne effect [27]). This is notably the case when the mismatch between cues and possibilities for action limits the expression of procedural knowledge [2], as evidenced by participants wanting to perform actions they are accustomed to but that were not measured or supported by our environment.

Also, while our design considered that the effort involved in each subsequent task was proportional to the number of changes that it required, this cannot be mapped directly to a measure of difficulty. Instead, the level of difficulty more likely depends on the tools and approach chosen by each participant.

Finally, we were interested only in the use of commands that would help participants complete the task while ignoring those that did not have an effect. We may therefore have missed interaction

knowledge that manifests itself when participants try to use a tool that has no effect in the particular context.

While the qualitative analysis of the results mitigates some of these limitations, more experimentation is required to consolidate our results with other substantial examples of interaction knowledge. In particular, further experimental designs should expand the set of available commands to those from other applications beyond text- and graphics-editing, such as photo editing, video editing, CAD, etc. Further work should also involve other modern interaction styles in order to identify additional abstract principles of interaction knowledge. For example, one could investigate the abstract interaction principles relevant to Reality-based Interaction [22] or Blended Interaction [23].

6.2 Users Accumulate and Transfer Interaction Knowledge about Digital Environments

Most participants showed a significant preference, across all tasks, for the type of tool matching the priming for their group, despite having both sets of tools available. This suggests that these participants inferred the possibilities for action on these digital objects through the cues that we initially provided in the interface (tool-bars and feed-forward from selection), which elicited interaction knowledge related to this type of tool. Although different tasks were proposed — some of which involving repetitive steps — the need to find alternative tools rarely resulted in these participants switching their tool type of choice for the remainder of the task. This is evidenced by the small number of times that participants switched tool types in most tasks. This resonates with *functional fixedness* [14], a cognitive bias that limits a participant's ability to use a tool in a non-familiar way or to use a different tool to solve a task. In our case, participants may have assigned a single function to the tools they recognized and associated it with the type of object (text or graphics) rather than with a property (e.g., color). Some participants also seemed to be stuck at personal choice, adopting a “satisficing” [12, 40] approach. They recognized that they preferred to use what they knew would solve the problem, despite it not being the most efficient method, e.g., using the *Fill* tool to change the background color of characters instead of the *Highlighter* tool. Both functional fixedness and satisficing are likely to inhibit technical reasoning, thereby reducing the participant's use of their interaction knowledge.

In a few cases however, participants found that they could use a secondary type of tool that normally would not be compatible with an object that works with the primary one, e.g., using text selection on graphics. In those cases we can infer that technical reasoning was at play. We observed participants who tried to find a tool that created the desired effect (i.e., change a property), without paying attention to the fact that such a tool would normally not be compatible with an object for which tools of a different type could be used, e.g., it would be impossible to use Microsoft Word's *Background Fill* tool to change the text's highlight color. This suggests that these participants were focusing on the property that they wanted to change rather than following learned procedures or erratically testing different tools until finding a match. Arguably, such behavior is compatible with participants exerting technical reasoning and

applying their interaction knowledge about the principles related to changing object properties that they associated with the tools.

In summary, while a majority of participants focused on the object types to make their decisions about tools, a minority had moments where they based their decisions only on what the tools do. Both behaviors are examples of knowledge of either tool or object from past experience being transferred to an unfamiliar or novel environment. The former behavior is representative of procedural knowledge, where little conscious effort is involved to execute an action [2], whereby the latter is representative of the kind of transfer of knowledge that occurs in a technical reasoning process [35]. In this case, technical reasoning comes as an assistive cognitive mechanism when facing a tool-based problem when the procedure-based solution is not available. This work therefore supports the notion that interaction knowledge is an essential form of knowledge for interaction in digital environments, similar to mechanical knowledge of the physical world, but for the digital world.

6.3 Implications for HCI

Interaction knowledge draws from mechanical knowledge, which models abstract knowledge about physical objects and principles in the physical world [35]. Mechanical knowledge is the basis for technical reasoning, which itself is based on the ability to reason analogically about object-based interactions. Humans' ability to reason analogically is not limited to physical interactions, as has already been shown in previous studies training computer users for novel interfaces [36, 38]. By modeling the knowledge about digital interactions that gets transferred analogically as interaction knowledge, we can inform the design of future interactions to build on existing knowledge that users have of the digital world and how to signify them, as well as to motivate the characterization of the principles behind this knowledge.

Following these characterizations, users of a novel interface could build on their past experiences to recognize what digital tools can and cannot do with objects. Additionally, rather than being used as a general or sparse body of knowledge, interaction knowledge could be compartmentalized according to the paradigms, platforms or devices that are being considered. For example, we could talk about WIMP interaction knowledge, touch-based interaction knowledge or VR interaction knowledge to refer to the principles and objects that are central to these interaction styles.

Furthermore, the notion that interaction knowledge describes relationships between digital tools and objects supports the development of alternatives to the application-centric paradigm [6, 34]. Rather than forcing users to accept the tool set that comes with and is limited to each application, interaction knowledge leverages the human cognitive ability to understand tool-based interactions and to use tools and objects in unusual ways, thus expanding the scope and power of interactive systems. This is notably explored by Maudet [28], who shows that limitations imposed by current design tools result in practices that extend beyond these applications' environments, e.g., using external digital images as color sources to be imported with an eyedropper tool.

Designing digital tools — rather than applications — would encourage users to engage more deeply with their digital environments by incorporating these tools into their workflows as they

see fit based on their effects and by facilitating their appropriation as we do with physical tools [13]. For example, users could create a document by switching between graphics-editing tools for images and drawings and text-editing tools for writing, all within the same editing environment. This form of *ownership* of digital tools has already been explored through concepts such as Instrumental Interaction [5], observing for example that a color picker should change the color of anything having a color property regardless of the application in which it is found [7].

Our results support the development of user interfaces centered around the existing interaction knowledge of digital tools and the possibilities for action that they create towards digital objects, as well as the development of new digital tools based on principles underlying new interaction knowledge. In other words, interaction knowledge of digital tools can serve the development of a novel property-based interaction style that lets users recognize the possibilities for action of the environment by the properties that the tools affect and whether the target objects feature these properties [37].

7 CONCLUSIONS & FUTURE WORK

Inspired by mechanical knowledge from the Technical Reasoning hypothesis [35], we introduced *interaction knowledge* as abstract knowledge of digital tools and the possibilities for interaction with objects that they enable. Interaction knowledge is the knowledge we acquire from experience in digital or physical environments and that we transfer to interact in the digital world so that we can exert technical reasoning. We designed a novel WIMP interface and an experimental protocol to explore priming interaction knowledge with interaction cues. Using different toolbars and selection interactions, we primed participants with either text or graphics cues and analyzed their strategies to solve a series of tasks in an environment where both types of interactions were possible with its digital objects.

Our results show a significant preference of participants, across tasks, for the type of tool corresponding to the specific interaction cues that were presented, regardless of the perceived effort involved to perform the task. However, we also observed some participants switching between tool types to find more efficient ways to complete the tasks, and seemingly unaware of the novelty of such a possibility. This suggests that they were able to exert technical reasoning based on their interaction knowledge about the properties that the tools changed, rather than rely on procedural knowledge of how the tools are normally used. Moreover, a control group that was not exposed to priming showed that the objects themselves had no significant priming effect, as evidenced by their mixed choice of tools.

We argue that these results are compatible with the transfer of interaction knowledge within and across digital environments, similar to the transfer of mechanical knowledge underlying technical reasoning processes [35]. While our experiment is limited to a WIMP-based interface, we believe it is a step forward in establishing the value of interaction knowledge and technical reasoning in digital environments. Future work should investigate interaction knowledge pertaining to other areas beyond text and graphics editing, for example, consolidating high-level principles that apply across platforms such as desktop, mobile, VR and AR, or in

interaction styles such as Blended Interaction and Reality-Based Interfaces. More generally, such efforts should also further study the roles of these cognitive processes in interactive behavior. Interaction knowledge and technical reasoning provide a promising theoretical ground for such studies. They also open up a rich design space for novel types of interactions that are based on humans' ability to interact with and through tools, to appropriate them, and to transfer tool-based and property-based knowledge across domains.

ACKNOWLEDGMENTS

We thank the participants of our study for their time. This work was supported by the European Research Council (ERC), Grant N° 695464 ONE: Unified Principles of Interaction.

REFERENCES

- [1] Alan Agresti. 2006. *An Introduction to Categorical Data Analysis*. John Wiley & Sons, Ltd, New York, NY, USA. <https://doi.org/10.1002/0470114754>
- [2] John Robert Anderson. 1983. *The Architecture of Cognition*. Harvard University Press, Cambridge, MA, USA. <https://doi.org/10.4324/9781315799438>
- [3] Pippin Barr, James Noble, and Robert Biddle. 2003. Icons R Icons. In *Proceedings of the Fourth Australasian User Interface Conference on User Interfaces 2003 - Volume 18* (Adelaide, Australia) (AUI '03). Australian Computer Society, Inc., AUS, 25–32. <https://dl.acm.org/doi/abs/10.5555/820086.820093>
- [4] Thomas Baudel and Michel Beaudouin-Lafon. 1993. Charade: Remote Control of Objects Using Free-Hand Gestures. *Commun. ACM* 36, 7 (jul 1993), 28–35. <https://doi.org/10.1145/159544.159562>
- [5] Michel Beaudouin-Lafon. 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)*. Association for Computing Machinery, New York, NY, USA, 446–453. <https://doi.org/10.1145/332040.332473>
- [6] Michel Beaudouin-Lafon. 2017. Towards Unified Principles of Interaction. In *Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter (Cagliari, Italy) (CHIItaly '17)*. Association for Computing Machinery, New York, NY, USA, Article 1, 2 pages. <https://doi.org/10.1145/3125571.3125602>
- [7] Michel Beaudouin-Lafon, Susanne Bødker, and Wendy E. Mackay. 2021. Generative Theories of Interaction. *ACM Transactions on Computer-Human Interaction* 28, 6, Article 45 (11 2021), 54 pages. <https://doi.org/10.1145/3468505>
- [8] Alethea Blackler, Vesna Popovic, and Doug Mahar. 2010. Investigating Users' Intuitive Interaction with Complex Artefacts. *Applied Ergonomics* 41, 1 (2010), 72–92. <https://doi.org/10.1016/j.apergo.2009.04.010>
- [9] Colin G. Butler and Robert St. Amant. 2004. HabilisDraw DT: A Bimanual Tool-Based Direct Manipulation Drawing Environment. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Austria) (CHI EA '04). Association for Computing Machinery, New York, NY, USA, 1301–1304. <https://doi.org/10.1145/985921.986049>
- [10] Susanne Bødker. 1987. Through the Interface – a Human Activity Approach to User Interface Design. *DAIMI Report Series* 16, 224 (Apr. 1987), 190. <https://doi.org/10.7146/dpb.v16i224.7586>
- [11] John M. Carroll, Robert L. Mack, and Wendy A. Kellogg. 1988. Chapter 3 - Interface Metaphors and User Interface Design. In *Handbook of Human-Computer Interaction*, Martin Helander (Ed.). North-Holland, Amsterdam, 67–85. <https://doi.org/10.1016/B978-0-444-70536-5.50008-7>
- [12] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2014. Supporting Novice to Expert Transitions in User Interfaces. *ACM Comput. Surv.* 47, 2, Article 31 (11 2014), 36 pages. <https://doi.org/10.1145/2659796>
- [13] Alan Dix. 2007. Designing for Appropriation. In *Proceedings of HCI 2007 The 21st British HCI Group Annual Conference University of Lancaster, UK 21*. British Computer Society, London, UK, 1–4. <https://doi.org/10.14236/ewic/HCI2007.53>
- [14] Karl Duncker and Lynne S Lees. 1945. On Problem-solving. *Psychological Monographs* 58, 5 (1945), i. <https://doi.org/10.1037/h0093599>
- [15] Umberto Eco. 1979. *A Theory of Semiotics*. Vol. 217. Indiana University Press, Bloomington, IN, USA.
- [16] G. H. Freeman and John H. Halton. 1951. Note on an Exact Treatment of Contingency, Goodness of Fit and Other Problems of Significance. *Biometrika* 38, 1-2 (1951), 141–149. <https://doi.org/10.2307/2332323>
- [17] J. J. Gibson. 1986. *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, USA. <https://doi.org/10.4324/9780203767764>
- [18] Mary L. Gick and Keith J. Holyoak. 1980. Analogical Problem Solving. *Cognitive Psychology* 12, 3 (1980), 306–355. [https://doi.org/10.1016/0010-0285\(80\)90013-4](https://doi.org/10.1016/0010-0285(80)90013-4)
- [19] Tovi Grossman, George Fitzmaurice, and Ramtin Attar. 2009. A Survey of Software Learnability: Metrics, Methodologies and Guidelines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) (CHI '09).

- Association for Computing Machinery, New York, NY, USA, 649–658. <https://doi.org/10.1145/1518701.1518803>
- [20] Robert R. Hoffman. 1989. A Survey of Methods for Eliciting the Knowledge of Experts. *SIGART Bull.* 1, 108 (April 1989), 19–27. <https://doi.org/10.1145/63266.63269>
- [21] Jörn Hurtienne and Johann Habakuk Israel. 2007. Image Schemas and Their Metaphorical Extensions: Intuitive Patterns for Tangible Interaction. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction* (Baton Rouge, Louisiana) (*TEI '07*). Association for Computing Machinery, New York, NY, USA, 127–134. <https://doi.org/10.1145/1226969.1226996>
- [22] Robert J.K. Jacob, 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* (Florence, Italy) (*CHI '08*). Association for Computing Machinery, New York, NY, USA, 201–210. <https://doi.org/10.1145/1357054.1357089>
- [23] Hans-Christian Jetter, Harald Reiterer, and Florian Geyer. 2014. Blended Interaction: Understanding Natural Human–Computer Interaction in Post-WIMP Interactive Spaces. *Personal Ubiquitous Computing* 18, 5 (6 2014), 1139–1158. <https://doi.org/10.1007/s00779-013-0725-4>
- [24] J. Johnson, T.L. Roberts, W. Verplank, D.C. Smith, C.H. Irby, M. Beard, and K. Mackey. 1989. The Xerox Star: a Retrospective. *Computer* 22, 9 (1989), 11–26. <https://doi.org/10.1109/2.35211>
- [25] Victor Kaptelinin and Bonnie Nardi. 2012. Affordances in HCI: Toward a Mediated Action Perspective. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) (*CHI '12*). Association for Computing Machinery, New York, NY, USA, 967–976. <https://doi.org/10.1145/2207676.2208541>
- [26] J. Karat, 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* (Boston, Massachusetts, USA) (*CHI '86*). Association for Computing Machinery, New York, NY, USA, 67–71. <https://doi.org/10.1145/22627.22350>
- [27] H.A. Landsberger. 1958. *Hawthorne Revisited*. Cornell University, Ithaca, NY, USA.
- [28] Nolwenn Maudet. 2017. *Designing Design Tools*. Theses. Université Paris Saclay (COMUE). <https://tel.archives-ouvertes.fr/tel-01827014>
- [29] Joanna McGrenere and Wayne Ho. 2000. Affordances: Clarifying and Evolving a Concept. In *Proceedings of Graphics Interface 2000* (Montréal, Québec, Canada) (*GI 2000*). Canadian Human-Computer Communications Society, Toronto, Ontario, Canada, 179–186. <https://doi.org/10.20380/GI2000.24>
- [30] Donald A. Norman. 1988. *The Psychology of Everyday Things*. Basic Books, New York, NY, USA.
- [31] Donald A. Norman. 1999. Affordance, Conventions, and Design. *Interactions* 6, 3 (5 1999), 38–43. <https://doi.org/10.1145/301153.301168>
- [32] Donald A. Norman. 2010. Natural User Interfaces Are Not Natural. *Interactions* 17, 3 (may 2010), 6–10. <https://doi.org/10.1145/1744161.1744163>
- [33] Donald A. Norman. 2013. *The Design of Everyday Things*. Basic Books, New York, NY, USA.
- [34] Midas Nouwens and Clemens Nylandsted Klokmoose. 2018. *The Application and Its Consequences for Non-Standard Knowledge Work*. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3173574.3173973>
- [35] François Osiurak, Christophe Jarry, and Didier Le Gall. 2010. Grasping the Affordances, Understanding the Reasoning: Toward a Dialectical Theory of Human Tool Use. *Psychological Review* 117, 2 (2010), 517–540. <https://doi.org/10.1037/A0019004>
- [36] Peter G. Polson, 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* (Toronto, Ontario, Canada) (*CHI '87*). Association for Computing Machinery, New York, NY, USA, 27–32. <https://doi.org/10.1145/29933.30856>
- [37] 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* (New Orleans, LA, USA) (*CHI '22*). Association for Computing Machinery, New York, NY, USA, Article 579, 17 pages. <https://doi.org/10.1145/3491102.3501877>
- [38] John Rieman, 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* (Boston, Massachusetts, USA) (*CHI '94*). ACM, New York, NY, USA, 438–444. <https://doi.org/10.1145/191666.191816>
- [39] Brian H. Ross and Thomas P. Moran. 1983. Reminders and Their Effects in Learning a Text Editor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, Massachusetts, USA) (*CHI '83*). Association for Computing Machinery, New York, NY, USA, 222–225. <https://doi.org/10.1145/800045.801615>
- [40] Herbert A. Simon. 1956. Rational Choice and the Structure of the Environment. *Psychological Review* 63, 2 (1956), 129. <https://doi.org/10.1037/h0042769>
- [41] Brygg Ullmer and Hiroshi Ishii. 1997. The metaDESK: Models and Prototypes for Tangible User Interfaces. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology* (Banff, Alberta, Canada) (*UIST '97*). Association for Computing Machinery, New York, NY, USA, 223–232. <https://doi.org/10.1145/263407.263551>
- [42] Daniel Wigdor and Dennis Wixon. 2011. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Morgan Kaufmann, Amsterdam. <http://www.sciencedirect.com/science/book/9780123822314>