Interface Dis/Similarities: Investigating Characteristics Influencing Perceived Differences Between GUIs

Raphaël Perraud, Sylvain Malacria

Interface *Dis*/similarities: Investigating Characteristics Influencing Perceived Differences Between GUIs

Raphaël Perraud
Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRIStAL
Lille, France
raphael.perraud@inria.fr

Sylvain Malacria
Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 CRIStAL
Lille, France
sylvain.malacria@inria.fr

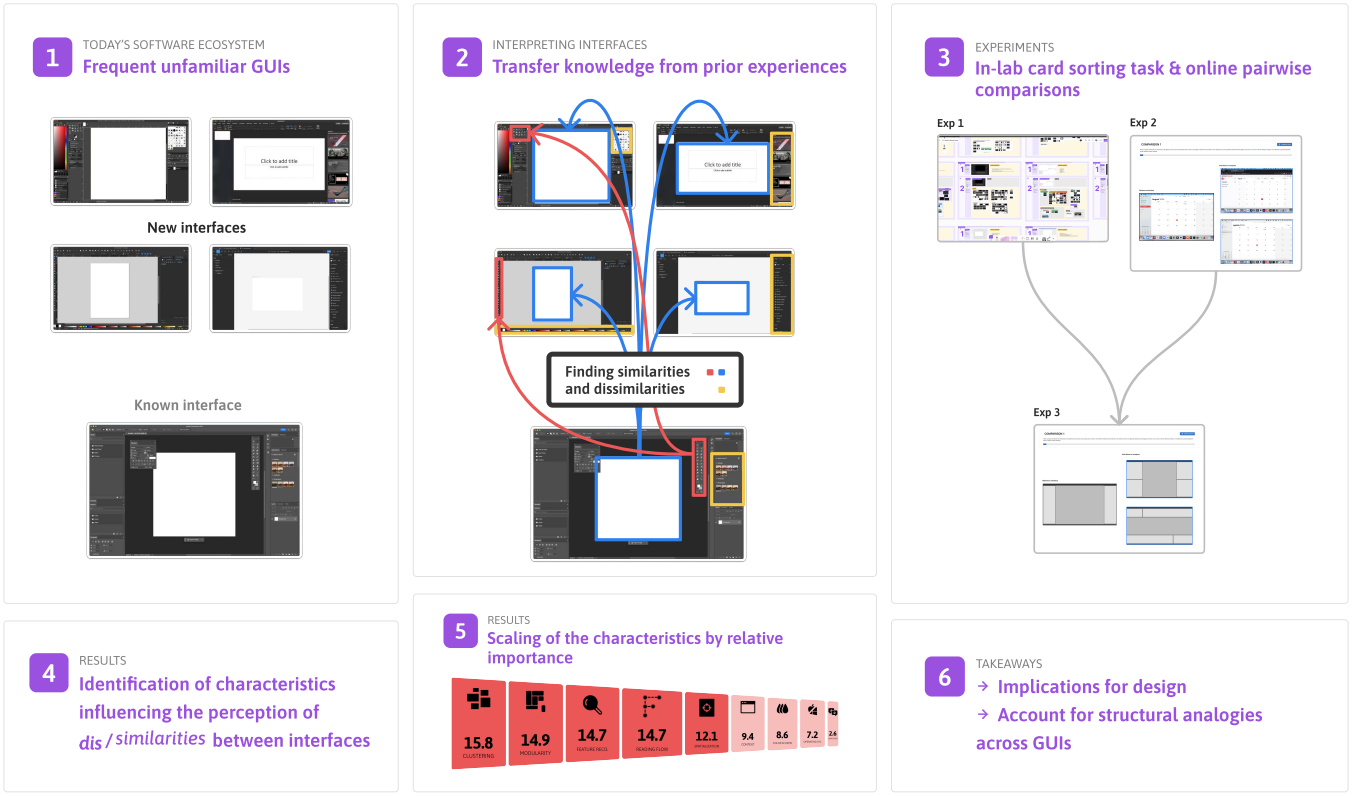

Figure 1: When facing unfamiliar Graphical User Interfaces (GUIs), users leverage prior knowledge from familiar ones (1). They do so by identifying key elements and comparing *dis*/similarities with known interfaces in order to transfer knowledge between them (2). Through a card sorting task conducted in laboratory and online surveys (3), we identify the primary interface characteristics influencing users' perceptions of interface *dis*/similarities (4). We scale these results to establish an order of importance (5). Our results provide actionable implications for design and enables further research on how users approach novel GUIs (6).

Abstract

While HCI research acknowledges that prior knowledge can be transferred from familiar interfaces to unfamiliar ones, we lack an understanding of which interface characteristics support this process. To adress this issue, we conducted three experiments to identify the interface characteristics that influence the perception of *dis*/similarities between software interfaces. The first, which involves a card-sorting activity, identifies seven intrinsic characteristics of interface. The second, conducted via an online pairwise comparison survey, identifies three characteristics inherent to the interface's display context. Finally, the third experiment contrasts the ten identified characteristics to determine their respective influence on perceived interface *dis*/similarities. Altogether, our results provide actionable guidance for understanding how users perceive differences between interfaces and how such perceptions may inform or facilitate analogical transfer of knowledge from familiar to unfamiliar interfaces.

This work was presented at *CHI '26, Barcelona, Spain* - Authors' version

Raphaël Perraud and Sylvain Malacria. 2026. Interface Dis/Similarities: Investigating Characteristics Influencing Perceived Differences Between GUIs. In Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26). https://doi.org/10.1145/3772318.3790718

CCS Concepts

• **Human-centered computing → Human computer interaction (HCI)**; *Empirical studies in HCI*; Graphical user interfaces.

Keywords

interface similarity,transfer learning,perception,Graphical User Interfaces (GUIs)

ACM Reference Format:

Raphaël Perraud and Sylvain Malacria. 2026. Interface *Dis*/similarities: Investigating Characteristics Influencing Perceived Differences Between GUIs. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3772318.3790718

1 Introduction

Users frequently face unfamiliar Graphical User Interfaces (GUIs) as software evolves across updates, platforms, and competing applications. These GUIs can differ more or less from previously encountered ones, in terms of layout, structure, visual representations, and interaction styles. As a result, users may be familiar with the intended functionality of a software, but still need to interpret a new GUI to use it effectively [24, 55]. To do so, they often rely on knowledge from previously used applications [87] to transfer GUI skills and knowledge across applications [6, 12, 80, 109]. This transfer relies on the ability to detect both similarities and dissimilarities between interfaces, in what we term in this work as *dis*/similarities.

Situations where users have to compare two interfaces are extremely frequent: when learning a new version of a software [76], when switching to a new platform [6], when using a new software that resembles one they already know [80], or when both applications rely on similar interaction metaphors [48]. These comparisons are not incidental. They directly influence how quickly users understand the software [15], which elements they recognize, and how effectively they transfer skill between GUIs [60]. Users draw on prior experience to perform these GUI comparisons [22, 43, 87]. They typically do this by forming analogies between the *familiar* GUI (the one they already know) and the *unfamiliar* GUI (the one introduced by the new software). This form of *analogical reasoning* supports such inferences by aligning new GUIs with previously learned ones based on perceived correspondences [53, 94]. When users judge two GUIs as similar, they are more likely to adopt similar strategies [39, 68]. When they judge the GUIs as different, they tend to anticipate a greater learning cost [15]. Therefore, understanding which GUI characteristics enable these mappings is critical, not only for learning and onboarding, but also for redesign, migration, and cross-application knowledge transfer.

Despite how routinely users compare GUIs in everyday computing use, we still lack a clear understanding of which interface characteristics they rely on when comparing that two interfaces feel close or distant. The literature on *consistency* is extensive on the criterion that defines it [9, 13, 39, 87, 91], yet offer little insights on what is actually *perceived* by users. Computational approaches to GUI similarity attempt to quantify distances between screenshots [23, 30, 57] or interface hierarchies [17, 40, 96]. However, these models rely on descriptive properties or deep features that are not grounded in how users actually perceive similarities and differences and lack interpretability [35]. As a result, designers lack perceptually meaningful tools to anticipate how different a redesigned interface would be perceived, how disruptive changes might be, or how easily knowledge could transfer across versions, often leaving the burden of adaptation to users.

To close this gap, we identify the characteristics that lead users to judge two interfaces as similar or different, and we articulate the following research question: Which perceptual characteristics of GUIs make them appear similar or dissimilar at first sight? We address this question through an empirical study that uncovers and quantifies these perceptual factors. Understanding this perceptual layer is a necessary step toward modelling how analogical reasoning applies to GUIs, particularly for supporting cross-interface knowledge transfer.

Through a mixed-methods approach, we first conduct an open card sorting study [34, 93] to identify key internal GUI characteristics that shape users' perception of *dis*/similarities. We then use pairwise comparisons through an online survey to explore how contextual characteristics, such as operating systems and platforms, influence these perceptions. Finally, we systematically generated GUIs that vary along the identified characteristics, to evaluate the relative impact of each characteristic on perceived *dis*/similarities. Our results clarify how users compare interfaces at first sight and yield design implications for improving knowledge transfer in interface design, while laying the groundwork for the integration of cognitive models of analogical reasoning in HCI.

2 Related work

Our work relates to how users build a mental model of a GUI, how they interpret a novel and different GUI, and how such *dis*/similarities can be characterized and estimated.

2.1 HCI research on GUI *dis*/similarities

Literature has proposed several approaches to estimate interface similarity, by comparing differences in the UI hierarchical structure [17, 40], or using convolutional networks to represent interfaces through embeddings and compute layout [30, 57] or semantic similarity [23, 52, 107]. While these methods quantify distances derived from descriptive or latent properties, they remain largely disconnected from how users themselves perceive differences between GUIs. Moreover, convolutional networks typically operate as "black boxes" [79], producing a single similarity score without interpretable insight of *why* two interfaces might be perceived similar or different. In addition, these previous solutions were mostly built on datasets restricted to websites or mobile applications, overlooking the diversity of productivity GUIs [8, 36, 40, 96]. Consequently, designers lack perceptually meaningful tools for anticipating the impact of GUI changes or for understanding cross-application similarity.

Another line of work focused instead on quantifying the impact of these *dis*/similarities [76, 88] or exploring how systems could better support the transition between two interfaces. Typically, one approach consists in displaying a translation interface (basically the interface from the software the user is familiar with) the user can interact with to operate the unfamiliar software [6, 48, 80]. These concerns are pervasive in HCI because situations in which users

must infer the possibilities of a new interface while transferring prior knowledge are ubiquitous. Such situations can be, but are not limited to, when a GUI is modified following an update [104, 106], when switching between two software with of similar domains (e.g. transitioning from Photoshop to GIMP [80]), or simply when downloading a software for achieving a goal for the first time (for instance running a 3D modelling software for the first time).

As a summary, situations where users must interpret a new interface while relying on prior knowledge are extremely common. Prior work has either attempted to quantify interface differences through computational metrics or to mitigate them through translation systems that replicate the familiar interface. Yet, these approaches do not tell designers which differences actually matter to users, nor when an adaptation mechanism is warranted. A principled understanding of how users perceive *dis*/similarities between a familiar and an unfamiliar GUI is therefore needed. Our work addresses this gap by identifying the perceptual characteristics that shape first-sight judgments of GUI *dis*/similarities.

2.2 Interpreting new GUIs and recognition of prior knowledge

When interacting with software, users progressively build a *mental image* of the GUI, a visuo-spatial representation shaped by spatial memory and salient landmarks [54]. Even when incomplete, such images provide a concrete perception of the interface [103].

Alongside this visual representation, users develop a *functional knowledge* of the interface, i.e. the domain of commands expected in relation with achievable tasks [4] (e.g., a pen tool draws a colored line). When encountering an unfamiliar GUI, users do not start from scratch: they relate it to prior *mental images* and attempt to map their *functional knowledge* onto the new context [72].

This interpretive process often relies on analogical reasoning, where users search for correspondences between the unfamiliar interface and previously learned ones [28, 86]. Analogies are more easily identified when interfaces share similar visuo-spatial schemas [67] or higher-order organizational structures [28], such as panel hierarchies or task workflows. These analogies guide initial expectations and inform early interaction attempts. They help to infer how to interact with tools by identifying visual characteristics that cue possible actions [42, 105]. In GUIs, such analogies activate prior *functional knowledge* and support trial-and-error exploration, enabling users to infer how to operate unfamiliar tools or commands even without prior experience [71].

In practice, users interpret unfamiliar GUIs by grounding them in prior mental images and functional expectations, forming analogies that help them infer how the new interface may work. To understand this process, it is necessary to identify the specific *interface characteristics* that trigger these analogies and shape early perceptions of similarity or difference. Because users can already infer possible actions and meanings from visual characteristics alone, without any manipulation [42, 105], our study focuses on experiments with static interfaces to isolate the perceptual factors that shape the perception of *dis*/similarities.

2.3 Other lines of research on the perception of GUIs in HCI

Parallel lines of research have examined how users perceive and process visual information in GUIs. Research on visual complexity highlights how information density [5, 65], visual variety of form and colors [70], spatial organization [103], and perceivability of details [41] shape early impressions of an interface. Gestalt principles, such as grouping by proximity or collinearity, help users organize screens into coherent spatial chunks [77], supporting comparisons at multiple levels of granularity. *Mental images* emerge from users' spatial memory of interfaces [89] and from characteristic components that act as landmarks [100–102], providing a stable perception of what the interface looks like. Recent work posited that certain spatial arrangements may prime functional expectations, such as interpreting a top-aligned toolbar as a text-editing environment, but provided limited justification for why such mappings hold [82].

While previous work explains how users perceive and remember individual GUIs, they do not describe which perceptual characteristics users rely on when comparing multiple interfaces side by side. In this work, we address this gap by identifying the interface characteristics that actually lead users to form such interpretations and that support cross-interface comparison, enabling us to understand how users judge *dis*/similarities at first sight.

3 Identifying the characteristics of the perception of interface *dis*/similarities

To investigate the characteristics influencing users' perceptions of *dis*/similarities between interfaces, we conduct a study consisting of three interrelated experiments (Figure 2) approved by our local institutional review board.

3.1 Approach

The overarching goal of this work is to explore how different characteristics of an interface influence users' perception of *dis*/similarities between interfaces. When comparing two interfaces, users likely construct analogies between them. To better understand this, we aim to identify the specific interface characteristics that users rely on to construct such analogies.

We therefore conducted **Experiment 1**, through an open card sorting activity [34, 93], to assess which characteristics influence users' perception of *dis*/similarities.

We then conducted **Experiment 2** to explore how the contextual conditions in which an interface is displayed, such as operating system or language, influence users' perception of *dis*/similarities between interfaces.

Finally, we combine the characteristics identified in **Experiments 1** and **2** to inform the design of **Experiment 3** in which we quantify the relative importance of these characteristics in shaping users' perceptions of GUI *dis*/similarities.

3.2 Type of interfaces tested

We focused on *productivity software* as defined by Garcia [27]. We chose them due to their large adoption among diverse user groups, which facilitates recruitment and ensures that feedback is based on relevant experiences. Additionally, productivity software are

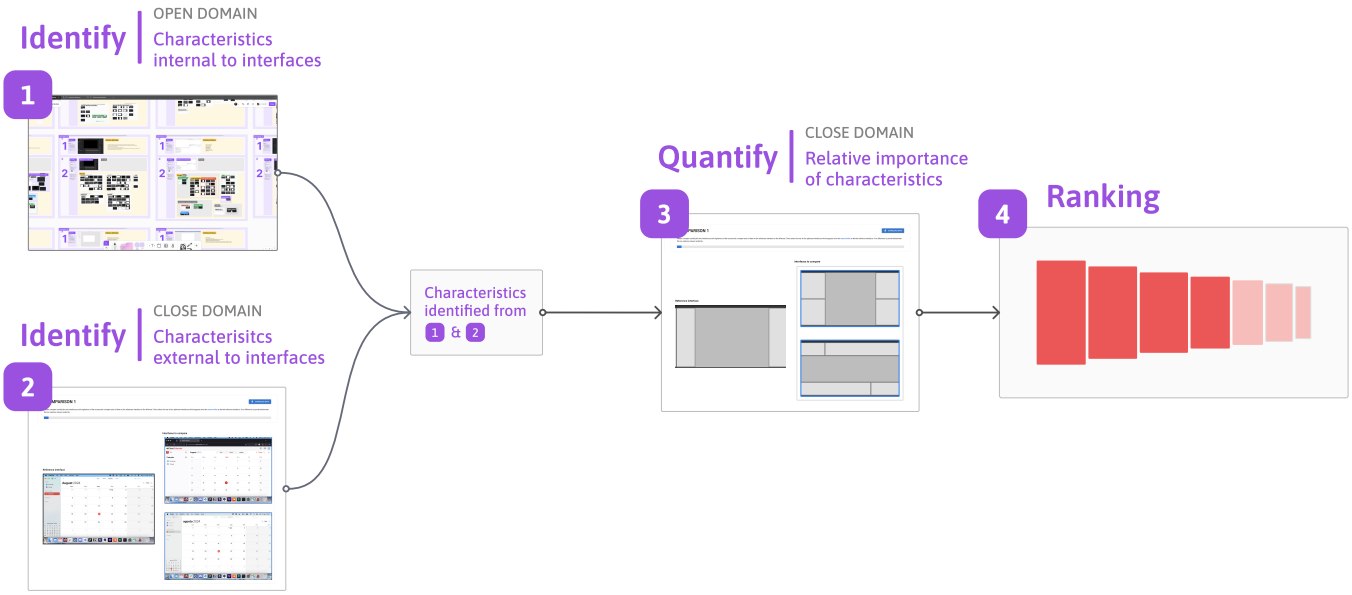

Figure 2: Overview of the studies dependencies: we (1) identified characteristics internal to GUIs through an open card sorting task, completed by (2) external characteristics previously identified through pairwise comparisons. This enabled to (3) quantify the relative importance of the characteristics identified to inform how these characteristics influence the perception of *dis*/similarities.

feature-rich and often customizable [61], offering a variety of interface elements and configurations for a detailed analysis. The diverse design approaches of application developers provide a comprehensive dataset for examining different GUI elements and their impact on user perception. Focusing on productivity software also aligns with real-world scenarios where users frequently transition between different tools, highlighting the practical relevance of understanding how users perceive interface *dis*/similarities between them.

4 Experiment 1: Identifying intrinsic characteristics

Analogical mappings require identifying characteristics between analog interfaces to establish similarities between them [10]. Because these characteristics are numerous and cannot be preselected arbitrarily, we first conducted an open card-sorting task [34, 93] to elicit the features users naturally rely on when comparing GUIs.

4.1 Rationale

To identify the characteristics that users rely on when comparing GUIs, we relied on the exploratory method of card sorting [34, 93] to reveal characteristics spontaneously. It is particularly suited for revealing the latent structure underlying first-impression judgments [2]. In our case, it is therefore suited to eliciting users' conceptual structures and uncovers the visual and structural cues they use to contrast interfaces. This bottom-up approach ensures that the characteristics identified in subsequent analyses are grounded in users' perception.

4.2 Experimental setup

4.2.1 Procedure. Participants joined a video call and were asked to first open their Web browser and complete an online preliminary questionnaire, answering various demographics and 7-points Likert-style questions on comparing software GUIs (see OSF repository [1]). They were then asked to visit a collaborative whiteboard website, shared with the experimenter. Participants first watched a video demonstration of how to use the collaborative whiteboard and its features. Then, the experiment moved to the *characterization* phase, where participants were presented with a single interface and instructed to depict it with *attributes* that characterize it. An *attribute* was defined as *an aspect, characteristic or property of an interface that defines its appearance, behavior, or functionality, and not related to the goal or type of the application (e.g. "2D map" or "navigation")*, and created by adding a text box with a name and description for it. Each participant had to create at least 3 attributes. This phase was intentionally framed as a scaffolding step: pilot studies revealed that engaging directly in full categorization was cognitively demanding without first grounding participants in attribute creation. The experiment then moved to the *sorting* phase, where 30 interfaces were revealed, including the interface to depict in the first stage. Participants were instructed to group GUIs based on the previously defined attribute they fit in. Basically, it consisted in creating group boxes for each attribute, naming them accordingly, creating a free text box in it to give a textual description and move interfaces sharing these attributes by drag-and-dropping them. A GUI could be duplicated to be included in several attribute groups. Participants were allowed, and frequently chose, to revise,

[1] Anonymized data, analysis scripts, materials and interface assessed by participants are available online: https://osf.io/cxywe/?view_only=79a720f0f86d4323ae4b33969b2aa0f1

rename, or delete their initial attributes in the following phase and create new ones during this phase. Attributes could also be nested and re-organized during the task. The experiment took on average 54min.

Participants were repeatedly reminded that the task was supposed to reflect their subjective perception, and that there was no "correct" or "incorrect" answer. When participants considered having finished grouping the GUIs with similar attribute, the experiment moved to the *conclusion* phase with a debriefing questionnaire inquiring about their understanding of the tasks, a self-retrospective of their sorting approach, and explicitly asking about the characteristics of interfaces that help them to compare interfaces and find similarities between them. Participants were encouraged to "think-aloud" [21, 51] all along the experiment.

4.2.2 Materials and apparatus. Participants were presented with a set of 30 interface screenshots drawn from a diverse range of desktop productivity software [1], including applications for video, photo, and audio editing; text, slide, and spreadsheet editing; as well as vector graphics, music production, and 3D modeling. The screenshots depicted each interface immediately after opening a new file or project in it, and using the default GUI of a newly installed version of the software.
Participants used their own computer for the experiment. We used FigJam [2] as web-based collaborative whiteboard for the sorting activity, as it allows positioning elements on a 2D canvas, draw and label rectangular sections, and adjust zoom level. Interviews were conducted remotely using the Cisco Webex video conferencing software, and online questionnaire were implemented using Limesurvey,

4.2.3 Participants. A total of 22 participants, aged 21 to 52 (M=31.9, SD=10.7), were recruited through email invitations and personal contacts for this experiment. 10 had a background in computer science or UX design. All participants reported regularly interacting with GUIs on different types of devices or operating systems (OS), with 12 of them using at least three OS every month (five of them having a background in computer science or UX design).

4.3 Data processing

4.3.1 Collected data. We recorded the interviews' audio and captured a video of each participant's interface on FigJam. One of the authors transcribed the audio. *Sorting data* exported from FigJam included attribute groups, descriptions, interfaces, and their hierarchy for analysis.

4.3.2 Data analysis. We conducted an inductive, reflexive thematic analysis [11] on participants' verbal comments and their labeling of groupings. Our goal was to identify recurring patterns in how participants conceptualized *dis*/similarities between interfaces during the sorting task and accompanying verbalizations. In the initial familiarization phase, one author transcribed audio recordings and reviewed the whiteboard artifacts. During data cleaning, we standardized attribute labels and removed any that violated the instruction not to group interfaces by application domain (e.g., "music," "video"). Coding was conducted at a semantic level, focusing on the explicit content of participants' descriptions. The primary coder then generated initial codes by tagging segments of text and sorting decisions in a spreadsheet, which allowed tracking code frequency and co-occurrence. These codes were iteratively reviewed and grouped into candidate themes based on conceptual similarity, with the goal of identifying the main characteristics influencing users' perception of *dis*/similarities. Although we entered the analysis with general expectations informed by prior literature on GUI perception (e.g., layout, complexity, semantics), we adopted an inductive stance to remain open to unanticipated forms of interface comparison. We acknowledge that the analysis was primarily conducted by a single researcher, which may limit triangulation; however, all themes were reviewed and discussed among the co-authors to ensure analytic coherence and relevance. Data excerpts provided are translated by the authors for participants whose native language is not English. We chose to report the results according to the logic of identification of characteristics expressed verbally by the participants during the experiment. We elaborate on these themes in the results in the next section of this article, and report significant differences only.

4.4 Results

We identified seven characteristics influencing the perception of interface *dis*/similarities. These characteristics were derived inductively from participants' groupings in the card sorting task. Specifically, we analyzed patterns in the labels participants assigned to their groups, identifying recurring categories across multiple participants. To refine and validate these categories, we examined participants' verbal comments, which helped contextualize and clarify the meaning of group labels. In the following sections, we present each characteristic, supported by illustrative excerpts from participants' comments. These verbatims also informed the short textual descriptions associated with each grouping, which included a title and a brief explanation.

4.4.1 ColorScheme: The analysis of colorimetry is carried out in the first instance. Participants frequently first sorted based on color schemes. Many expressed color was a primary feature, highlighting how quickly it influenced their initial impressions: *"at first sight..."* (P9), *"it's the first thing that strikes me"* (P8), *"first vision, [...] second vision"* (P15), *"at first glance"* (P17). Participants expressed a perceptual division between darker and lighter GUIs. They reported darker GUIs as complex and requiring expertise (*"dark equals expert"* (P8)) and lighter GUIs as accessible and inviting, which aligns with non-expert preferences because they look *"bright and welcoming"* (P8). Some participants even ultimately reconsidered the priority of color for comparing GUIs, and removed it from their final categorizations. As one participant remarked, *"At first, I sorted the interfaces by the color [...], but in the end I didn't follow through because I think the white theme and the black theme have no place in this exercise"* (P1). This indicates a shift from first impressions to a deeper structural understanding in the comparison process.

4.4.2 Modularity: Understanding the GUI's level of modularity is crucial for correct interpretation. To understand this underlying

[2]FigJam is available online at https://www.figma.com/fr/figjam

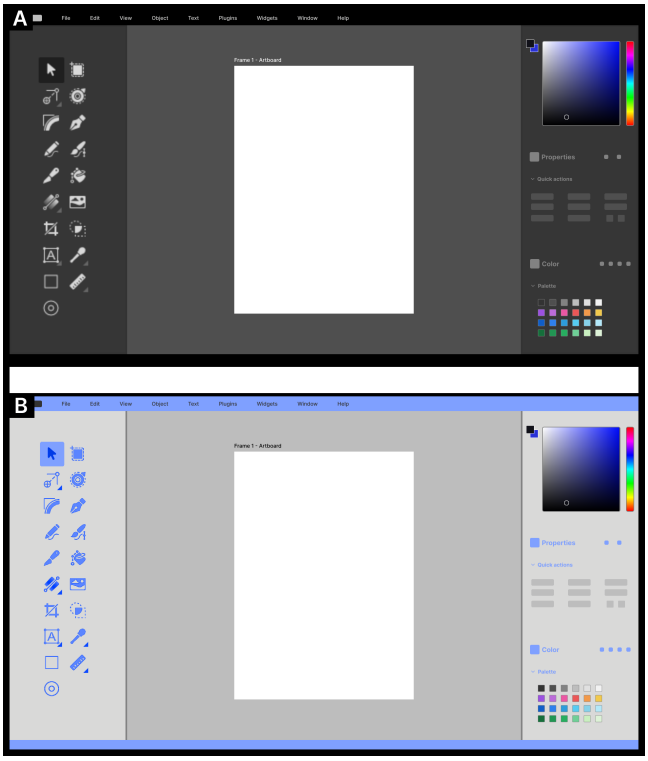

Figure 3: Illustration of colors scheme variations in dummy interface layouts. Multiple participants reported perceiving the dark themed interfaces as tailored for expert, while light themed interfaces more suitable for inexperienced users, and often starting to sort interfaces accordingly, before shifting toward deeper structural analyses.

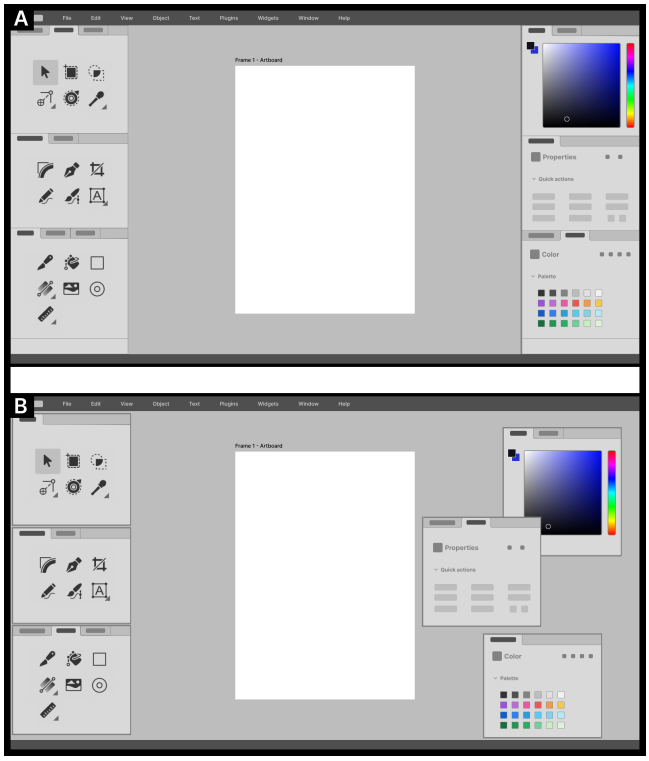

Figure 4: Illustration of modularity variations. A: static layout in which tool panels are docked and fixed within the workspace. B: flexible layout in which tool panels become floating, movable windows that users can freely reposition, resize, and reorganize according to their needs.

"interface's logic" and organization, many participants first considered the degree of modularity; whether the GUI comprises discrete, adaptable components that can be adjusted, moved, or reconfigured. Note that in this document, we will refer to a component as any visually and functionally distinct part of the interface, such as toolbars, windows, panels, or buttons that contributes to the layout and can be mentally grouped by functional similarity or spatial proximity. This was crucial to determine if spatial positioning could reliably guide their mental model of the GUI: *"Those are windows. I wouldn't be surprised if there were more apps than I think working in windows. [...] I have the impression that the sub-menu is displayed in a certain way when in fact it's not fixed, and it's a window that I could move around, [...] maybe this sub-menu can be moved around."* (P2) Knowing whether an interface is "static", such as MS Word or LibreOffice Writer, allowed participants to rely on spatial consistency, building mental models around familiar components' positions. Participants also associated modularity with increased complexity, often suggesting that GUIs *"which look more static induce possibly simpler software"* (P1). However, when interfaces appeared *"with many modular windows"* (P6), they noted that spatial location alone become limited; instead, they clustered GUI components by functionality, as P10 explained: *" When I look at the interface, the first thing I see is the overall look. [...] There are tools everywhere, you have to look everywhere [to access the application's features]. But there's a logic to the way the tools are presented [...] and so here I'm sorting out the way the tools are presented, and so here tools are on both sides of the interface".*

This interpretation suggests two approaches: relying on spatial cues for static interfaces or recognizing semantic objects like tools or parameters for modular interfaces, highlighting how modularity shapes comparison strategies.

4.4.3 Clustering: Ability to categorize the interface layout. Once participants determined whether spatial cues would guide their comparisons, they segmented GUIs into coherent functional zones. First, they performed a visual grouping of major components, identifying key areas such as the workspace (e.g., the *"main window"*), tool panels, indirect and direct manipulation panels, or content visualization regions as they built a *mental image* of each interface. These initial groupings reflected how participants perceived the interface "at a glance".

They then refined these groups by identifying what coherently binds the elements within each region, distinguishing functional units such as buttons groups (as illustrated by P5 who sorted interfaces based on *"buttons to move in the workspace, buttons to modify*

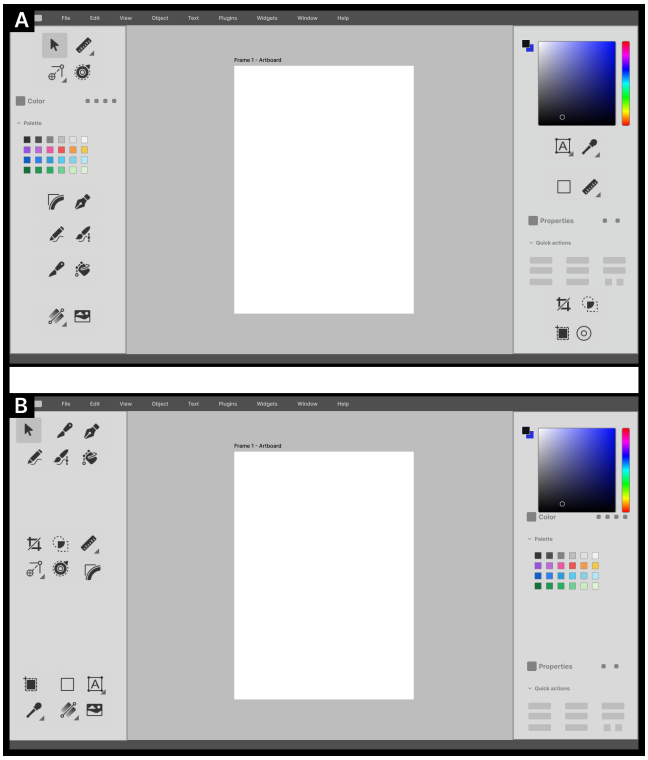

Figure 5: Illustration of clustering variations. A: interface with limited logical grouping of features, where tools and controls are redistributed across the layout, weakening the perception of distinct functional areas. B: interface showing clearly identifiable clusters, with tools, properties, color controls and workspace grouped into coherent functional zones.

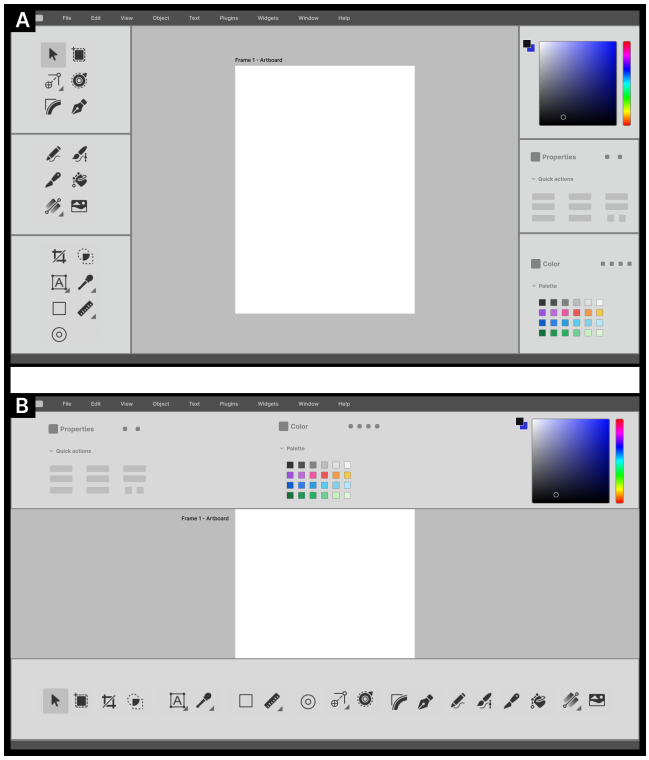

Figure 6: Illustration of reading flow variations. A: interface structured in vertical blocks, with tool groups and controls organized in separated panels indicating that they are independent to each other, guiding to a block-based reading flow. B: interface structured in horizontal layers, where properties, workspace, and toolbars form stacked "floors," supporting a horizontal and sequential interpretation of interface regions.

object's appearance"), parameter windows, and property inspectors like *"sections to edit text"* (P21), or *"file explorers"* (P19). The components of these groups share *structural similarities*, allowing participants to map perceived regions to meaningful roles within the interface, as expressed by P6: *"the overall layout, that big central space, stuff on the left, stuff on the right, toolbar at the bottom, toolbar at the top, a toolbox on the left which is fixed in my opinion"*. This approach enabled them to categorize interface regions while inferring functional relationships among GUI elements, as expressed by P4: *"Here, we have panels that allow us to move around in files. And here, those are tool palettes. I guess that's what allows us to move the panel. [...] It is often the same thing: we have a file explorer on one side and an edition [tab] on the other."*

This categorization approach allowed participants to identify shared elements, visual representations, and layout similarities across interfaces. P13 illustrated this process: *"when I analyze the interfaces, well, broadly speaking, they always look the same [...] there are always three zones. One where you can draw on your files or tools [...] Then there's another section where you can get an overview and another section where you can work on everything [...] So when I look at each interface, I feel like that there similar, so I want to group them together"* (P13). It describes three distinct zones for file access, content overview, and creation. This structural understanding enabled participants to group interfaces based on consistent functional patterns.

4.4.4 ReadingFlow: Knowing which way to "read" the interface. Participants frequently used orientation-related adjectives, such as *"vertical"* or *"horizontal"*, along with sequencing terms to describe layouts, for instance *"line accumulation, displaying by floors"* (P3), *"linear interface"* (P10), and *"more organized and sequential (e.g., left->right->left->right) layouts"* (P14). Recognizing an interface's reading flow helps establish a structured approach to interpreting relationships between elements, creating a clear hierarchical understanding. This understanding is pivotal for analyzing new interfaces and relating them to familiar ones, as P3 illustrated: *"The grid display makes me think of 3D, so I know it's 3D software. Then, if [the interface] is in floors (ndlr: orgzanized in multiple horizontal layers), I know it's either video editing or music editing software."* Such insights allow users to categorize components hierarchically and spatially.

Orientation-based reading also aids in discerning the relationships between interface regions: whether they are independent blocks or represent *"multiple views"* (P8) of a single object, each

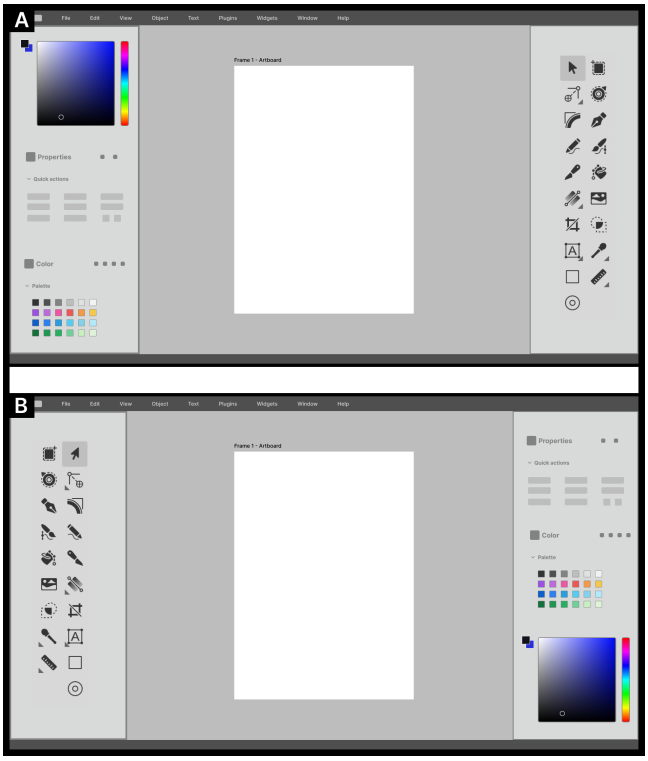

Figure 7: Illustration of spatialization variations. Both interfaces presents similar arrangements but mirrored, exposing similar controls and buttons at different locations.

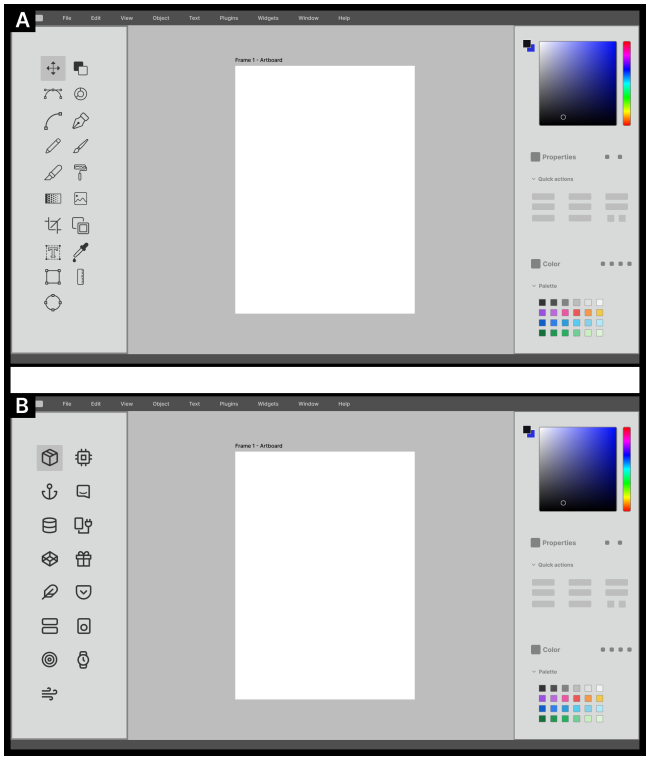

Figure 8: Illustration of feature recognition variations. A: interface displaying icons commonly associated with vector-graphics software, enabling users with prior experience of this domain to readily infer the tools' functions. B: interface presenting a different set of icons that follow alternative visual conventions, which reduces immediate recognizability and increases the effort required to interpret each tool. Icons are deliberately out of context to highlight differences in recognizability.

with varying levels of importance. By parsing interfaces in this way, users determine levels of similarity between layouts and evaluate the compatibility of interface reading flows.

4.4.5 Spatialization: Spatial anchoring of interface components. Participants showed a tendency to rely on spatial localization for precise comparisons, often expecting features to remain in consistent locations across interfaces, particularly for participants accustomed to "static" interfaces. This preference for spatial anchors is illustrated by P7: *"The first thing that comes to mind [...] are the functionalities present on one software package present in the same place as on the other? Typically in the case of Word and Office: if I click on the top left-hand corner, will I see where I want to save my data or save it, etc.? Doing so makes it easier to compare"* (P7). Rather than using semantic labels, participants frequently opted for spatial anchors terms (e.g., top, bottom, left, right, corners) to reference interface components. This reliance on spatial anchors highlights how participants mentally reconstruct and compare interfaces based on spatial cues, as exemplified by P6: *"Left border Pane, Left/Right or Right: Large, centered workspace, with side zones."* (P6)

This is not surprising as users are known to rely on spatial location to recall features by using landmarks like corners and edges [103]. Given the need to retain visual information for comparison, they segment interfaces spatially, a strategy essential for forming *mental images* and facilitating efficient comparisons: *"so the first section would be everything up there, the top bar, the second [...]"* (P19).

4.4.6 FeatureRecognition: Expectations based on software's recognized action capabilities. After segmenting the interface into zones, participants focused on the purpose and expected behavior of individual components. Feature recognition refers to users' ability to interpret tools, menus or widgets based on their visual representation and their similarity to components seen in other software. Participants identified familiar features, such as tool palettes, layer lists, or secondary menus, by interpreting iconic and textual cues that recur across many applications. They compared widgets from unfamiliar interfaces to those from familiar software to infer how to interact with them: *"like in Grasshopper[...] it's parametric, but basically, you could achieve the same result by linking blocks together to create your final product"* (P15). References such as *"the list of icons, as for example in Word"* (P12) or *"the secondary menu on the left with mostly icons and text [...] that should open windows with new options upon selection"* (P2) illustrate how recognizable visual

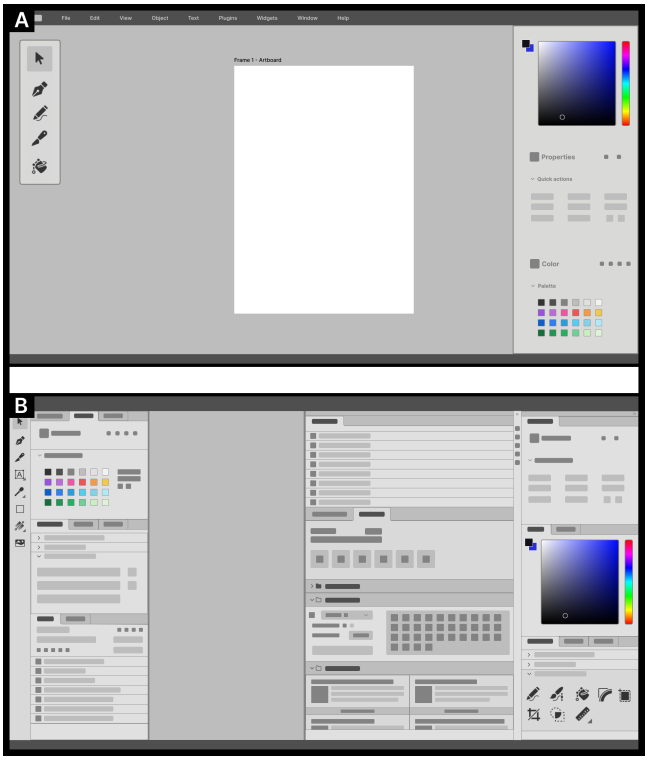

Figure 9: Illustration of visual complexity variations. A: interface containing only a small set of essential tools and minimal on-screen information. B: interface with high visual complexity, displaying numerous panels, dense parameter lists, and layered controls. The increased information load makes the interface harder to parse at a glance and can overshadow other perceptual attributes during comparison.

elements help to form analogies from prior knowledge. This capacity for deduction allows users to enrich their structural analysis by interpreting the likely functionalities of components, particularly when comparing applications of same domain (e.g., Photoshop and GIMP for photo editing).

This reasoning can have a cascade effect: recognizing familiar features helps users infer the role of adjacent or unfamiliar components, as P1 illustrated: *"the interface makes me think that blocks on the left influence the rest of the timeline on the right"*. By drawing from previous experiences with similar interfaces to interpret unfamiliar ones, users also formed expectations on the features that should be available: *"in 3D software interfaces, there must be a "control" or "management" principle in these blocks"* (P0).

4.4.7 VisualComplexity: Perceived complexity impedes analysis based on other attributes. Some interfaces were classified as *"not particularly appealing, [...] even pretty scary"* (P8), because they look *"difficult to get to grips with, that don't make you want to try them out, that aren't very accessible to the average user, and that are aimed more at professionals or experts"* (P10). This perception arises from users' interpretation skills, which are needed to understand interface elements and their potential actions. When the information load was especially dense and difficult to parse, participants often sorted these interfaces into a rebuttal *"overloaded"* (P3) category.

The visual complexity also influenced estimations of expertise and effort required, as reflected by comments on categories *"Screen too white. I don't want to read what's written"* (P3) and *"I don't understand"* (P17). Here, visual complexity acts as an "aggregative" attribute, grouping interfaces that seem too complex to tackle together and overshadowing other characteristics in the evaluation process.

5 Experiment 2: Identifying contextual characteristics

Forming appropriate analogies may also require interpreting contextual information surrounding an interface. While the previous experiment focused on intrinsic interface characteristics, this study examines whether and how contextual factors influence users' perception of *dis*/similarities through an online survey with pairwise comparisons [73]. Here, we define context as any configuration from external factors not inherent to the software itself that affects how the interface is displayed within the screen. This includes system-level settings (e.g., operating system, language), window size, or the mode of delivery (e.g., embedded in a web application versus shown in a standalone application).

5.1 Experimental setup

5.1.1 Procedure. After giving consent, participants answered various demographics, provided information regarding software usage (available on OSF [1]) in a preliminary questionaire, and then moved to the experiment itself. Each trial showed participants 3 images of interfaces: a *reference screen UI* and two *candidate screen UI*[3] (see 2 in Figure 2). For each trial, participants were instructed to specify which *candidate screen* resembled the most to the *reference* one by clicking on it. If participants could not decide between the *candidate interfaces*, they were instructed to select one randomly. The experiment then progressed to the next trial. There was a 500ms interval displaying a blank screen between each trial to minimize change detection in the *candidate screen*. Participants carried on until all pairs had been assessed. On average, the experiment took approximately 13 minutes to complete.

A *screen* corresponded to a full-screen capture of macOS displaying a single application window under a certain configuration. The reference *screen* was systematically the same: a full screen capture of Apple Calendar running natively in English, with its window maximized to fill a 16:10 display, and the macOS dock shown at the bottom (see figure 10). Candidate *screen* each differed along one given *characteristic* (see section 5.1.3).

5.1.2 Rationale. This experiment relies on a pairwise comparison, a simple task that does not require training [59]. It is well-suited for non-expert participants and avoids common calibration issues [98]. Paired comparison protocols offer 3 advantages over direct-rating tasks [73]: (1) it is less cognitively demanding [14] (selecting among two items is easier than providing an ordinal rating); (2) it avoids normalization issues which can occur, for instance, when users avoid extreme responses; (3) it provides higher sensitivity and lower

[3] In the following, we use *screen* to refer to *screen UI*

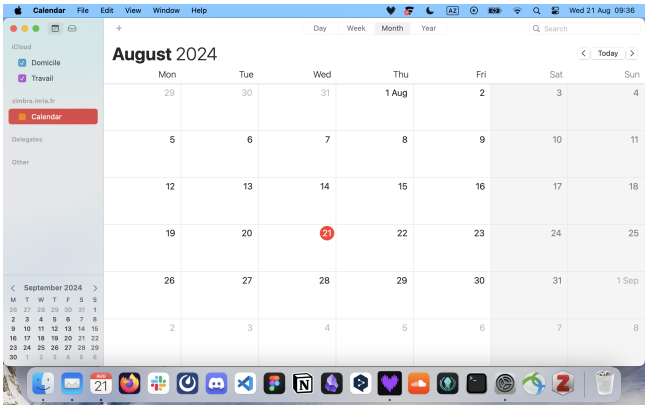

Figure 10: The reference interface presented to participants for all comparisons: a full-screen capture of Apple calendar application running natively in English, displayed with its window maximized to fill a 16:10 display, with the macOS dock visible at the bottom of the screen

measurement error [90]. It is thus well-suited for examining the impact of a specific set of defined characteristics on the perception of *dis*/similarities between interfaces. Since this experiment focuses exclusively on the context of display, the domain of characteristics is well-defined, allowing us to pre-determine the conditions to be evaluated (see 5.1.3). We excluded characteristics related to the physical context and focused on the display context of a unique software.

We selected a calendar application as the reference interface because it represents a widely familiar and visually stable category of productivity software, minimizing domain-specific biases. Its low visual complexity allowing us to control for content-related biases while isolating the effect of display context on perceived *dis* /similarities.

5.1.3 Design. We opted for a within-subjects design, where participants evaluated 55 unique pairs of *screen* with 10 candidate interfaces plus the reference interface to assess. The order of pairs was randomized for each participant. We added attention checks [4] comparisons to confirm that they honestly answered the questionnaire at fixed completion stages: 25%, 50% and 75%. We also added attention checks when the option at the same on-screen location was selected 5 times consecutively. Participants were screened out after two failed attention checks.

Candidate interfaces were similar to the reference one, except that they differed on one of the following characteristics, with two levels each: OperatingSystem (*Linux* or *Windows*; instead of macOS), ApplicationContext (running in a *Web Browser* or *Virtual Machine*, instead of native), MaximizedMode (with *visible dock and system bar* or *hidden dock and visible system bar*, instead of maximized and visible dock), WindowSize (occupying 1/3 or 2/3 of the display, instead of the maximized), Language (*Italian*, as a language using the latin alphabet or *Korean* as a language using a different alphabet, Hangul; instead of english).

[4]Showing a *screen* of Apple Mail instead of Apple Calendar in same conditions than the *reference* interface

We selected these characteristics to cover an exhaustive set of contextual influences, focusing on variables that could be controlled externally to the application interface and might significantly affect users' perception of an interface.

5.1.4 Participants. We recruited 60 participants (age 18-68, M=31.8, SD=10.3) for this web survey using Prolific [1], all different from Experiment 1. They conducted the experiment from their own device, but were pre-screened to ensure they used a desktop or laptop computer with a display of at least 1920x1080px. They were compensated for their participation (hourly rate≈10£). Participants who failed twice to attention checks stopped the experiment, their data deleted, but were still compensated.

5.2 Data analysis

5.2.1 Perceived similarity score. Our analysis focuses on practical significance, estimating what portion of the population will select one condition as more similar to the reference interface than the other, using Just-Objectionable-Differences (JOD) scaling. When 75% of observers select one condition over another, we assume that the quality difference between them is 1 JOD. We scaled pairwise comparisons results following Thurstone's model V assumptions [95] using a Maximum-Likelihood-Estimation method to estimates the quality scores following recommendations from Perez-Ortiz and Mantiuk [73] who provides a complete MatLab toolkit [58] for extensive analysis. This approach captures the order of conditions and the magnitudes of difference, making it more suitable than vote counting to compare items of similar quality (here *screen*) and express the magnitudes of difference in terms of practical significance using JOD [74].

To do so, we converted answers' table to comparison matrices, one per participant. We then performed outlier analysis to detect potential observers who performed very differently from the rest of the sample. For that, we set a customary threshold of 1.5 on the inter-quartile-normalised score L_{dist} to investigate if this score is close or above as suggested by [73]. Once we are confident there is no outlier in our dataset, we scaled the results and compute bootstrapped confidence intervals ($p < 0.05$).

5.2.2 Data filtering. In addition to attention checks that are added to comparisons, [33] propose to evaluate the individual consistency of each participant, by calculating the number of cyclic triads occurring in their choices. A cyclic triad occurs when comparisons are intransitive, (e.g. A is preferred to B, B is preferred to C and C is preferred to A). The coefficient of consistency [44] is then computed as follows for each observer: $\zeta = 1 - \frac{24c}{n^3 - n}$ where n is the number of stimuli (n=10) and c the number of cyclic triads. $\zeta = 1$ when there is no circular triads (i.e. perfect consistency) and will decrease to zero as the number of circular triads, and thus the inconsistency, increases. Participants' results were rejected if their coefficient of consistency was inferior to 0.75 [50]. This limit was decided to allow for some degree of input error (i.e. unintended clicking after page refresh) whilst still removing the most inconsistent participants. In our case, we excluded only one participant ($\zeta = 0.66$).

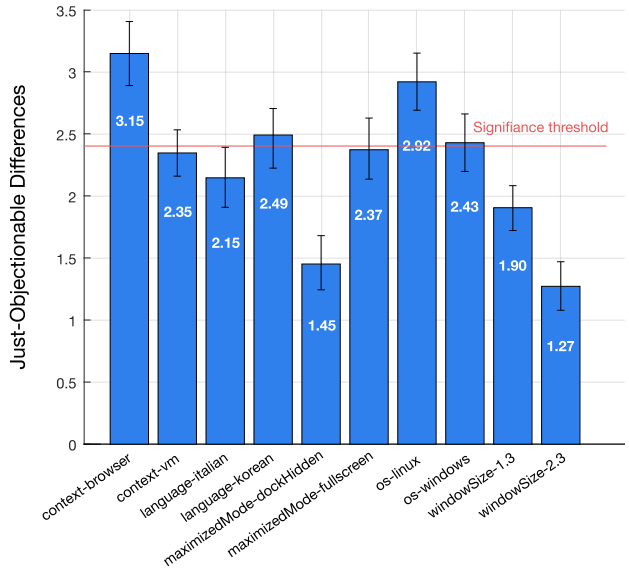

Figure 11: Scaling results and confidence intervals for pairwise comparisons. A difference of 1 JOD indicates that 75% of observers selected one condition as most similar to the reference *screen UI* than the other, 2 JOD corresponds to 91% of observers, 3 JOD 97.8%. The threshold of 95% of observers agreeing to select a condition over another is reached when at 2.4 JOD of difference.

5.3 Results

The preliminary questionnaire did not reveal significant differences in participants' prior experience with multiple operating systems, software, or devices that could allow use to identify subgroups of participants, we therefore analyzed the results as a whole. We plotted the scaling and the confidence intervals in Figure 11 illustrating the difference in JOD scores between each condition to the reference's score (calculated as $JOD = JOD_{Reference} - JOD$). Unexpectedly, all the conditions tested showed significant differences of more than one JOD from the reference, highlighting the importance of the interface display context in the perception of *dis*/similarities between two interfaces. Two groups of contextual parameters are observed. A first of weaker effect (< 2 JOD of difference) includes parameters related to application WINDOWSIZE, and MAXIMIZEDMODE. A second of stronger effect (> 2 JOD of difference) includes the interface display CONTEXT, the OS, and the display LANGUAGE.

95% (> 2.4 JOD) of observers noted an objectionable difference to the reference *screen UI* for the conditions OS $_{\text{WINDOWS}}$, OS $_{\text{LINUX}}$, LANGUAGE $_{\text{KOREAN}}$ and CONTEXT $_{\text{BROWSER}}$. Participants perceived the KOREAN condition to be more dissimilar than the ITALIAN condition. This is likely due to the significant difference between the Hangul alphabet and the latin-based alphabet. Indeed, the majority of participants (83.3%) used Latin-based alphabets while only a small proportion were familiar with non-Latin alphabets such as Greek, Gaj or Afrikaans (only one participant reported speaking Korean). Further analysis for this participant showed that the LANGUAGE $_{\text{KOREAN}}$ was perceived slightly less different from the *reference* than ITALIAN. As for CONTEXT $_{\text{VM}}$, it showed a *screen UI* of MacOS with the window of the virtual machine also running under MacOS, with very subtle visual indicators that may not be recognizable by people who are not familiar with it. This may explain the difference (0.80 JOD) with CONTEXT $_{\text{BROWSER}}$ which showed a Firefox window, with easily recognizable items (search bar, navigation buttons).

6 Experiment 3: Evaluating the relative impact of each characteristic on the perception of interface *dis*/similarities

Finally, to address our research question, **Experiment 3** operationalizes intrinsic and contextual characteristics identified in **Experiments 1** and **2**, and quantifies their relative contributions to perceived interface *dis*/similarities.

6.1 Experimental setup

6.1.1 Procedure. We replicated the procedure of Experiment 2, as it enables the results to be scaled onto a unified and interpretable metric, with the following adjustments. The reference *screen* used the same configuration as the reference in Experiment 2, presenting a GUI interface with the minimal items to fit our conditions (see Figure 12). Candidate *screen* each differed along one given characteristic (see section 6.1.3). On average, the experiment took 23 minutes to complete.

6.1.2 Characteristics. We kept the characteristics identified through the open card-sorting activity (see Section 4) and selected the ones that reached a threshold of 2.4 JOD (indicating that 95% of observers perceived an objectionable difference) in Experiment 2. Given that VISUALCOMPLEXITY, which overlaps with several other characteristics, tends to dominate interface comparisons and complicates users' ability to assess *dis*/similarities, we opted not to include this characteristic in the experiment. Instead, we rely on the extensive existing literature to address its perceived importance [5, 18, 65, 70, 81]. To ensure balanced comparisons across characteristics, we retained the two conditions of each characteristic identified from Experiment 2 (CONTEXT, OS, and LANGUAGE).

6.1.3 Design. We opted for a within-subjects design, where participants evaluated 153 unique pairs of *screen* with 18 candidate interfaces plus the reference interface to assess. We excluded the VISUALCOMPLEXITY characteristic from this experiment because it appeared to overshadow participants' evaluations of other criteria, rendering it impractical to derive an accurate measure of *dis*/similarities under such conditions. The same parameters for attention checks [5] and timing delays used in the previous experiment were applied.

Candidate interfaces were similar to the reference one, except that they differed on one of the following *characteristics*, with two levels each: OPERATINGSYSTEM (*Linux* or *Windows*; instead of macOS), APPLICATIONCONTEXT (running in a *Web Browser* or *Virtual Machine*, instead of native), LANGUAGE (*Italian*, as a language using

[5]The same *screen* was used for attention checks

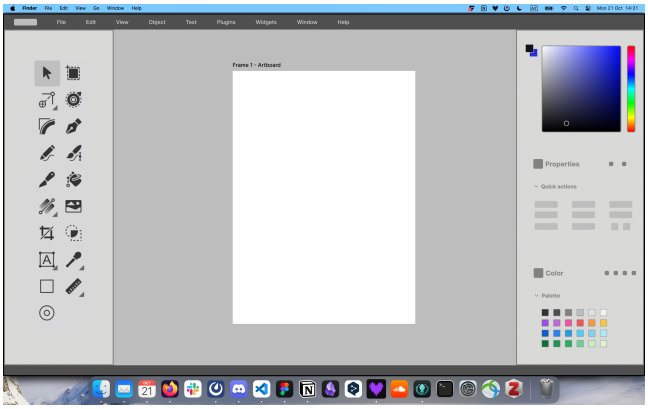

Figure 12: The reference interface presented for all comparisons was a full-screen capture of an interface application tailored for this experiment, running natively in English, displayed with its window maximized to fill a 16:10 display, with the macOS dock visible at the bottom of the screen.

Characteristic	W Score
Clustering	15.8
Modularity	14.9
Feature Recognition	14.7
Reading Flow	14.7
Spatialization	12.1
Application Context	9.4
Color Scheme	8.6
Operating System	7.2
Language	2.6

Table 1: Proportional scaling scores (W) for each characteristic quantifying their relative importance in the perception of *interface distance*.

the Latin alphabet or *Korean* as a language using the Hangul alphabet; instead of English), ColorScheme (*Dominant Color* or *Background theme*), Feature Recognition (*explicit* or *not explicit* icons), Clustering (*identifiable* or *not identifiable* clusters according to Gestalt principles), Modularity (*stackable* or *floating* windows), Reading Flow (*by blocks* or *by horizontal layers*), Spatialization (*internal* displacements of components or on *opposite* side of the interface). The variations shown to participants reused the interfaces introduced in Section 4.4 to illustrate the characteristics identified from **Experiment 1**; for the additional contextual characteristics identified as significant in **Experiment 2**, we instantiated the corresponding visual elements (such as the macOS dock or the browser frame to Figure 12). All resulting stimuli are available on OSF [1].

6.1.4 Participants. We recruited a total of 85 participants (that did not participate to experiments 1 and 2), aged from 18 to 60 (M=30.7, SD=9.25) for this Web survey using Prolific [1], all different from Experiments 1 and 2. They conducted the experiment from their own device. Once again, we pre-screened to ensure they used a desktop or laptop computer with a display of at least 1920x1080px. They were compensated for their participation (hourly rate≈10£). Participants who failed twice to attention checks stopped the experiment, their data deleted, but were still compensated. Participants had diverse backgrounds, a summary of the demographic information is available in the appendices.

6.2 Results

As in experiment 2, we computed JOD scores using Thurstone's Case V model [73]. We excluded two participants that were inconsistent ($\zeta = 0.69$ and $\zeta = 0.62$) in their answers. We also illustrated the scaling and the confidence intervals in Figure 13, showing the differences in JOD scores between each condition and the reference's score (calculated as $JOD = JOD_{Reference} - JOD$).

We found statistically significant differences between most conditions within each characteristic, suggesting participants considered *screens* as different. Only differences between the conditions of ColorScheme (0.14 JOD between ColorScheme Theme and ColorScheme DominantColor) and Context (0.15 JOD between Context VM and Context Browser) were not found significant, suggesting that the variations chosen for these two characteristics were minimal. ReadingFlow shows a large variability between its conditions, where the Linear condition displayed the greatest difference from to the Reference condition, a 86.5% higher difference from the Blocs condition.

By scaling the data by characteristic (see Figure 13), we can derive an order of importance of their influence on the perception of dis/similarities (see Table 1). The proportional score for each characteristic was computed as: $W = \frac{\text{JOD}}{\sum JOD}$. This metric quantifies the relative influence of each characteristic on the perception of dis/similarities, with higher scores indicating a lesser alignment with the reference condition and, consequently, that it can induce greater perceptual dissimilarity. Among the evaluated characteristics, Clustering (3.45 JOD), Modularity (3.25 JOD), ReadingFlow (3.20 JOD) and FeatureRecognition (3.21 JOD) exhibited the most substantial influence on participants' judgments of similarity, as reflected by their consistently high JOD scores.

Notably, external characteristics, such like Context (2.05 JOD), OS (1.58 JOD), and Language (0.56 JOD) tend to exhibit lower JOD scores than internal interface characteristics, except for ColorScheme (1.86 JOD), suggesting they generate less perceived dis/similarities. These characteristics fall below the 95% certainty threshold (2.4 JOD relative to the reference score), suggesting a diminished influence on the perception of dis/similarities. In contrast, structural interface characteristics (*e.g.*, Clustering, ReadingFlow, and Modularity) result in higher perceived dis/similarities. Spatialization of components appears to have a moderate effect on this perception of dis/similarities. These findings suggest that users rely more heavily on overarching structural elements and recognizable functionalities rather than contextual characteristics and color scheme when assessing dis/similarities between interfaces.

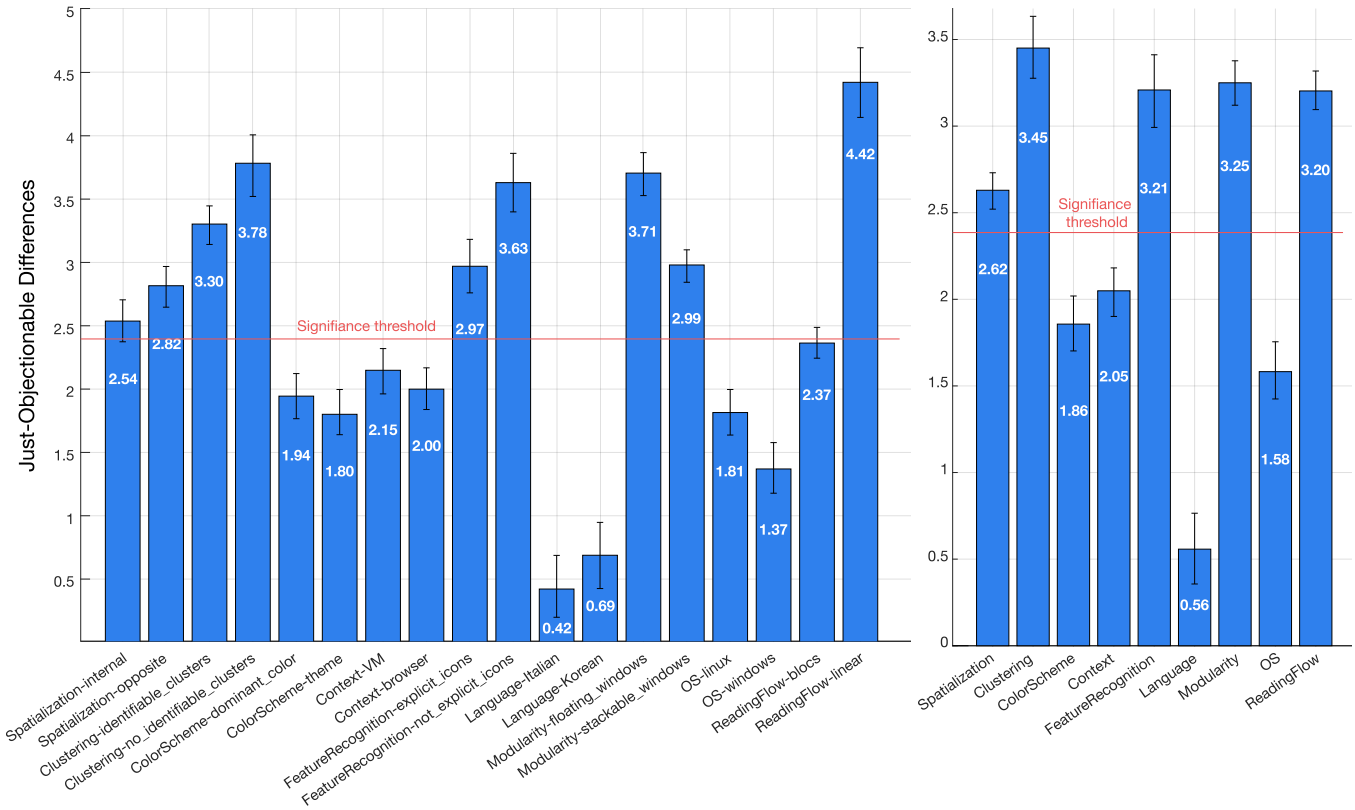

Figure 13: On the left, scaling results and confidence intervals for pairwise comparisons by condition. On the right, scaling results and confidence intervals for pairwise comparisons by characteristic.

7 Discussion

Our study demonstrates that the perception of interface *dis*/similarities is shaped by a combination of intrinsic and contextual characteristics, which collectively influence how users relate new interfaces to known ones. By integrating findings from our empirical approach, we show that users rely on overlapping layers of visual, structural, and functional information to compare interfaces, a process that plays a pivotal role in facilitating knowledge transfer across GUIs. This study is, to our knowledge, the first to both identify and weigh GUI characteristics identified in user studies, providing an empirically grounded account of their relative influence on perceived *dis*/similarities. This enables a more precise understanding of how people compare interfaces and offers a basis for perceptually aligned models of cross-interface learning and similarity computing. In the following sections, we contrast these characteristics in relation to ones identified in prior work and discuss implications for research and design.

7.1 How users perceive GUI *Dis*/similarities

7.1.1 Structural features dominate perception of interface dis/similarities. *Experiment 3* contrasts characteristics identified in *Experiments 1* and *2* and shows that structural characteristics (Clustering, ReadingFlow, and Modularity) and FeatureRecognition exert the most significant impact, compared to the contextual ones like Context, OS, and Language. This suggests that structural properties are prioritized to infer knowledge about the software and its interface functionalities, confirming insights from *Experiment 1*, where superficial cues like ColorScheme affected initial impressions but did not support deeper comparisons. This may explain why users are less disoriented in spatially consistent interfaces [88], or why multiple studies focused on interface consistency found mixed results when manipulating only part of the factors variables identified in this study [83, 87]. *Experiment 1* also revealed preferences for consistent Spatialization, yet its comparatively lower weight in *Experiment 3* suggests that spatial placement is interpreted in relation to higher-level structural organization. In other words, spatial information is useful when the interface structure is static, but becomes unreliable when GUIs are modular or customizable [76]. Overall, structural cues anchor semantic elements within a recognizable organization, enabling users to retrieve relevant *functional knowledge*, form expectations about possible operations, and construct analogical mappings from familiar to unfamiliar interfaces.

Implications for research: Our findings contribute to a clarification of the notion of interface consistency. Consistency, often described as the "look and feel" of an interface [63], has long been criticized for being too abstract [31] and missing guidelines that describe which design elements should be help consistent [88]. Our results specify the concrete factors that give substance to this

concept. Maintaining the relational organization of interface components, for instance keeping a comparable degree of modularity, a consistent clustering logic, and stable functional regions (e.g., tools on the left, parameters on the right), is what most strongly sustains users' expectations. In contrast, greater variability can be tolerated in language, visual appearance or in the precise spatial placement of elements, as long as the underlying structural schema remains recognizable.

7.1.2 Contextual characteristics have limited influence of the perception of dis/similarities. Contextual characteristics such as OS, Language, and ApplicationContext, while noticed, only moderately alter perceived interface *dis*/similarities, reinforcing that users generally expect interfaces to remain largely consistent across contexts with only minor differences. Previous observations already demonstrated that different Language limits user's understanding of interfaces [16, 32, 69, 108] and that it is challenging to maintain consistency across OS [6, 19] which can induce several *dis*/similarities between the interfaces of a same software. However, our study is, to our knowledge, the first to explicit the role of ApplicationContext as a characteristic which could induce differences between interfaces.

Implications for research: It has been showed that a software displayed on different platforms or devices can introduce *dis*/similarities between interfaces and affect user's perception [6, 19]. Our results extend these findings to the context of display, which can impacts interface perception also *within* the same device.

7.1.3 Visual complexity can overshadow other attributes in interface comparisons. The degree of modularity in an interface influences users' perception of visual complexity. Interfaces with flexible arrangements, like floating windows, are often perceived as more complex, as modularity increases cognitive load by requiring dynamic processing of spatial relationships while users tends to form spatial memory [7, 103]. However, this trade-off can be necessary in highly feature-rich software, where interface rearrangement options help mitigate the effects of *software bloat* [62]. High perceived complexity, marked by dense information and intricate layouts, can overshadow other characteristics, making it harder for users to assess *dis*/similarities. Participants in our study labeled such interfaces as "cluttered", echoing findings that high visual complexity can deter meaningful user engagement with interfaces [47, 64]. That being said, this observation should be contextualized according to cultural differences, as Eastern and Western cultures approach visual clutter differently [20, 26].

Implications for research: This suggests that above a certain threshold of visual complexity, users may stop comparing interfaces at a structural level and instead rely only on local cues. In such cases, structural anchors that normally support analogical transfer no longer operate as effective bridges between familiar and unfamiliar interfaces, highlighting the need to model visual complexity as a limiting factor on analogical reasoning in GUI learning.

7.2 Theoretical implications for HCI

7.2.1 Interface dis/similarities are processed at the mental image stage. Three characteristics identified in our study (Spatialisation, Clustering, VisualComplexity) align with dimensions described in Visuo-Spatial Working Memory research [3, 41, 65, 70, 103] (where attributes such as spatial placement, command groups, and amount of information are evoked). This partial overlap confirms that GUI comparison is mediated by the formation of a *mental image* [103] bridging the visual stimulus and the mental model of the interface built from it.

Implications for design: Our results support prior research emphasizing *mental images* in interface cognition [78, 103], enabling users to assess similarity and transfer knowledge between familiar and unfamiliar interfaces. As such, comparing interfaces involves both recognition and recall processes. Designers of new software should replicate "landmarks" [103] that are common in software of the same domain to support comparisons during onboarding and transfer of knowledge between them.

7.2.2 Users leverage structural mappings to trigger functional knowledge. Our results extend VSWM insights by identifying four novel characteristics (FeatureRecognition, Modularity, ReadingFlow and ColorScheme), revealing comparison strategies consisting in identifying layout patterns and shared features across interfaces. They also suggest that structural characteristics, such as ReadingFlow or Clustering helped them to recognize broad software categories (e.g., video editing, 3D modeling) and form expectations about possible behaviors and key features. When such similarities are present, they can facilitate analogical reasoning [38], allowing users to align unfamiliar interfaces with familiar ones and, in some cases, transfer relevant functional knowledge.

Implications for design: Users first leverage structural mappings to trigger *functional knowledge*. This means that a new productivity software that organize its interface differently from software of same domain would probably hinder the triggering of adequate *functional knowledge*. Conversely, systems that increase this perceived similarity can support knowledge transfer across interfaces and help the onboarding of an unfamiliar software by leveraging prior knowledge of a familiar software [48, 80].

7.2.3 Users form analogical mappings based on shared structural and functional patterns. Our findings indicate that users rely on a core set of characteristics that function as cognitive primitives akin to "entities" in Structure-Mapping Theory (SMT) [28]. Users grouped interfaces together when these organizational structures aligned, even when details varied, and such alignment triggered functional expectations grounded in prior experience, echoing established observations in analogical reasoning studies [10, 67, 97]. Analogical reasoning [29] posits that an unfamiliar interface structured like a familiar one tends to be expected to support comparable operations. Contextual cues can also act as primers, activating interaction knowledge (such as keyboard shortcuts or gesture-based interactions), echoing recent findings of Renom et al. [82]. Our findings explain why participants in [82] interpreted a horizontal top toolbar as "text-editing" and a vertical left toolbar as "vector-editing": such judgments rely on ReadingFlow that participants associated with specific software types. When a new interface reproduces this structural pattern, it activates expectations about available tools and corresponding interactions, supporting analogical retrieval from previously used applications.

Implications for design: Users often rely on trial-and-error when onboarding new software [45, 46]. Onboarding systems should

therefore support inference during this exploratory process by highlighting potential analogs to prior software experience, helping users recall relevant knowledge instead learning new one. Integrating analogical reasoning with established work on mental models [78, 84, 92] can inform the design of onboarding mechanisms that better align with users' expectations.

7.2.4 Addressing the gap between computational similarity and human perception. Current computational metrics [57, 111] quantify similarity from visual or structural representations but do not capture how users perceive *dis*/similarities at first sight. Our results address this gap by identifying and weighting the specific interface characteristics that users rely on.

Implications for research: This enables explainable similarity computing that is aligned with users' mental models: rather than providing a single distance value, our approach clarifies why two interfaces feel more or less similar. Integrating these weighted characteristics into computational metrics would therefore provide a path toward similarity models that are not only accurate, but also interpretable from a perceptual and cognitive standpoint.

7.3 Limitations and future work

The first experiment of our study relied on a card-sorting task. Such tasks can encourage participants to adopt a systemic and structural approach, focusing on the relationships between interfaces as a set rather than conducting an in-depth analysis of each individual interfaces. Indeed, we wanted to expose participants to a diverse range of software and interfaces, encompassing varied interaction metaphors, graphic styles, for different purposes (3D, audio, video, text, vector, etc.). Focusing on fewer interfaces might have allowed participants to engage in more detailed analyses, but at the risk of producing data specific to a single use case, limiting the broader applicability of our findings. To mitigate the problem, the experiment's initial phase required participants to analyze a specific interface in detail, during which they produced an analytical framework to reference during the subsequent card-sorting activity. That being said, some participants preferred to rely solely on this initial analysis and consequently sorted the entire set of interfaces according to the characteristics they had noted on the first interface, while others opted to revise and adjust their categories while sorting by identifying the characteristics that stand out from the whole set of interfaces. The reasonable number of participants [99] (n=22) in this card sorting study allows us to ensure a reliable structure of our interpretation made of the data collected after reaching meaning saturation (when researchers have "understand it all") [37].

The preliminary questionnaires (available on OSF) collected information about participants' software usage, but these data were insufficient to form reliable expertise groups. To mitigate expertise effects, **Experiment 1** used a broad and heterogeneous set of interfaces that maximized encounters with unfamiliar layouts. Familiar and unfamiliar interfaces were therefore processed differently: familiar ones could activate deeper functional expectations, whereas unfamiliar ones constrained participants to visual judgments. We consider this asymmetry beneficial, as familiar interfaces helped participants generate richer categories that they could extend or contrast when examining unfamiliar ones. Excluding all familiar interfaces would have reduced this categorization process and limited ecological validity, since real-world onboarding typically involves comparing a new interface with familiar ones [46, 56, 76, 85]. In **Experiment 2**, we used a simple and widely known interface, and **Experiment 3** relied on a neutral, dummy interface, limiting potential expertise biases. Across all studies, participants made purely perceptual judgments without interacting with the GUIs, meaning our findings primarily reflect first-impression processing. Expertise may nonetheless modulate moderately the weighting of characteristics, particularly for modularity [76]. Future work could stratify participants by cross-application expertise to examine how experience reshapes these perceptual factors.

In an effort to identify the roots of the user's perception of interface *dis*/similarities, our experiments were limited to visual inputs with static screen captures of interfaces. However, real-world use cases involve dynamic interfaces that react to user interactions. To build on this work, future research could explore how the perception of *dis*/similarities evolves during interface exploration, as users tend to adopt different comparison strategies based on the *dis*/similarities they encounter [76]. This would be a key step in better understanding how analogical reasoning operates during situated interaction, especially in exploratory scenarios, which represent a substantial portion of users' time with unfamiliar interfaces [46].

In terms of methodological implications, our approach demonstrates the utility of mixed-method evaluations in uncovering the complexity of interface perception. Combining a hierarchical clustering activity and pairwise comparisons allowed us to systematically identify and compare characteristics that influence on the perception of interface *dis*/similarities. Future research could extend this methodology by incorporating user behavior data to assess the efficiency and accuracy of user interface comparisons in real-time scenarios.

Our results identify and weights the interface characteristics that users rely on to compare interfaces. Future work could focus on developing metrics to quantify these characteristics, enabling the calculation of a multi-dimensional *interface distance* score that estimates how dis/similar two interfaces are perceived. A first step in this direction has been introduced in recent work on perceptual interface distance computation [75], which proposes interpretable metrics for comparing GUIs. Drawing on this groundwork, subsequent efforts could refine these metrics and develop predictive models of cross-interface knowledge transfer grounded in analogical reasoning.

Building on these findings, future research could explore how analogy-based representations can inform the design of generative user interfaces (GenUIs). Recent work positions GenUIs as AI-driven systems that adapt or synthesize interfaces dynamically in response to user goals and context [49, 66]. By encoding GUI characteristics that users perceive as making interfaces closer or more distant, analogy-based models could guide generative UIs, enabling adaptation via analogy (for familiarity and transfer) or deviation (for innovation). This direction also raises questions about how analogy-based cues can be integrated into human-AI co-creation workflows, allowing designers and end-users to steer generative adaptations through familiar patterns.

Finally, in LLM-powered support systems, analogy-driven interface representations may mitigate users' vocabulary limitations

[25, 46] by enabling extrapolation from known features to unfamiliar ones. This offers a principled way for AI systems to align with users' mental models during both interface generation and interactive help, akin to how visual analogy models abstract relational rules to bridge reasoning gaps [110].

8 Conclusion

In this work, we investigated how users perceive *dis*/similarities across GUIs and identified the interface characteristics they rely on when comparing unfamiliar interfaces with familiar ones. Through a mixed-methods approach combining hierarchical clustering and pairwise comparisons, we found that users predominantly rely on a small set of structural and visual characteristics to perceive interface similarity or difference. These characteristics explain how users form rapid judgments about unfamiliar GUIs and provide a perceptual basis for transferring prior knowledge during initial exposure. By clarifying which interface properties drive these perceptions, our findings complement existing computational approaches that quantify similarity by grounding analysis in user perception. They also offer designers a more interpretable lens for anticipating how interface changes or cross-application variations will be experienced by users, rather than relying on users to adapt on their own. Overall, this work provides a foundation for systems that aim to support cross-interface learning and for future models that integrate perceptual and cognitive mechanisms to predict how users interpret novel GUIs.

Acknowledgments

This work was supported by the Agence Nationale de la Recherche project Discovery (ANR-19-CE33-0006) and the région Hauts-de-France (France). We also thank Carl Gutwin for providing helpful feedback that informed and improved this project.

References

[1] 2024. Prolific. https://app.prolific.com/.

[2] Tessa Aarts, Linas K. Gabrielaitis, Lianne C. De Jong, Renee Noortman, Emma M. Van Zoelen, Sophia Kotea, Silvia Cazacu, Lesley L. Lock, and Panos Markopoulos. 2020. Design Card Sets: Systematic Literature Survey and Card Sorting Study. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. ACM, Eindhoven Netherlands, 419–428. doi:10.1145/3357236.3395516

[3] Bart Aben, Sven Stapert, and Arjan Blokland. 2012. About the Distinction between Working Memory and Short-Term Memory. *Frontiers in Psychology* 3 (Aug. 2012). doi:10.3389/fpsyg.2012.00301

[4] Eytan Adar, Mira Dontcheva, and Gierad Laput. 2014. CommandSpace: Modeling the Relationships between Tasks, Descriptions and Features. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. ACM, Honolulu Hawaii USA, 167–176. doi:10.1145/2642918.2647395

[5] Eren Akça and Ömer Özgür Tanriöver. 2021. A Comprehensive Appraisal of Perceptual Visual Complexity Analysis Methods in GUI Design. *Displays* 69 (Sept. 2021), 102031. doi:10.1016/j.displa.2021.102031

[6] Jessalyn Alvina, Andrea Bunt, Parmit K. Chilana, Sylvain Malacria, and Joanna McGrenere. 2020. Where Is That Feature?: Designing for Cross-Device Software Learnability. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. ACM, Eindhoven Netherlands, 1103–1115. doi:10.1145/3357236.3395506

[7] John Robert Anderson. 2000. *Learning and Memory: An Integrated Approach*. John Wiley & Sons Inc.

[8] Maxim Bakaev, Vladimir Khvorostov, Sebastian Heil, and Martin Gaedke. 2017. Evaluation of User-Subjective Web Interface Similarity with Kansei Engineering-Based ANN. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*. 125–131. doi:10.1109/REW.2017.13

[9] J. M. Christian Bastien and Dominique L. Scapin. 1995. Evaluating a User Interface with Ergonomic Criteria. *International Journal of Human-Computer Interaction* 7, 2 (April 1995), 105–121. doi:10.1080/10447319509526114

[10] Isabelle Blanchette and Kevin Dunbar. 2000. How Analogies Are Generated: The Roles of Structural and Superficial Similarity. *Memory & Cognition* 28, 1 (Jan. 2000), 108–124. doi:10.3758/BF03211580

[11] Virginia Braun and Victoria Clarke. 2022. *Thematic Analysis: A Practical Guide*. SAGE, London ; Thousand Oaks, California.

[12] Frederik Brudy, Christian Holz, Roman Rädle, Chi-Jui Wu, Steven Houben, Clemens Nylandsted Klokmose, and Nicolai Marquardt. 2019. Cross-Device Taxonomy: Survey, Opportunities and Challenges of Interactions Spanning Across Multiple Devices. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–28. doi:10.1145/3290605.3300792

[13] Nicolas Burny and Jean Vanderdonckt. 2022. (Semi-)Automatic Computation of User Interface Consistency. In *Companion of the 2022 ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. ACM, Sophia Antipolis France, 5–13. doi:10.1145/3531706.3536448

[14] Andrew P. Clark, Kate L. Howard, Andy T. Woods, Ian S. Penton-Voak, and Christof Neumann. 2018. Why Rate When You Could Compare? Using the "EloChoice" Package to Assess Pairwise Comparisons of Perceived Physical Strength. *PLOS ONE* 13, 1 (Jan. 2018), e0190393. doi:10.1371/journal.pone.0190393

[15] Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2015. Supporting Novice to Expert Transitions in User Interfaces. *Comput. Surveys* 47, 2 (Jan. 2015), 1–36. doi:10.1145/2659796

[16] Sayamindu Dasgupta and Benjamin Mako Hill. 2017. Learning to Code in Localized Programming Languages. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*. ACM, Cambridge Massachusetts USA, 33–39. doi:10.1145/3051457.3051464

[17] Niraj Ramesh Dayama, Kashyap Todi, Taru Saarelainen, and Antti Oulasvirta. 2020. GRIDS: Interactive Layout Design with Integer Programming. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–13. doi:10.1145/3313831.3376553

[18] Don C. Donderi. 2006. Visual Complexity: A Review. *Psychological Bulletin* 132, 1 (2006), 73–97. doi:10.1037/0033-2909.132.1.73

[19] Tao Dong, Elizabeth F. Churchill, and Jeffrey Nichols. 2016. Understanding the Challenges of Designing and Developing Multi-Device Experiences. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. ACM, Brisbane QLD Australia, 62–72. doi:10.1145/2901790.2901851

[20] Ying Dong and Kun-Pyo Lee. 2014. The Effects of Culture on Users' Perception of a Webpage: A Comparative Study of the Cognitive Styles of Chinese, Koreans, and Americans. In *Industrial Applications of Affective Engineering*, Junzo Watada, Hisao Shiizuka, Kun-Pyo Lee, Tsuyoshi Otani, and Chee-Peng Lim (Eds.). Springer International Publishing, Cham, 133–151. doi:10.1007/978-3-319-04798-0_11

[21] K Anders Ericsson and Herbert A Simon. 1993. Protocol Analysis: Verbal Report as Data (revised edition). *MITP, Cambridge, MA* (1993).

[22] Robert M. Fein, Gary M. Olson, and Judith S. Olson. 1993. A Mental Model Can Help with Learning to Operate a Complex Device. In *INTERACT '93 and CHI '93 Conference Companion on Human Factors in Computing Systems (CHI '93)*. Association for Computing Machinery, New York, NY, USA, 157–158. doi:10.1145/259964.260170

[23] Shirin Feiz, Jason Wu, Xiaoyi Zhang, Amanda Swearngin, Titus Barik, and Jeffrey Nichols. 2022. Understanding Screen Relationships from Screenshots of Smartphone Applications. In *27th International Conference on Intelligent User Interfaces*. ACM, Helsinki Finland, 447–458. doi:10.1145/3490099.3511109

[24] Wai-Tat Fu and Wayne D. Gray. 2004. Resolving the Paradox of the Active User: Stable Suboptimal Performance in Interactive Tasks. *Cognitive Science* 28, 6 (Nov. 2004), 901–935. doi:10.1207/s15516709cog2806_2

[25] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. 1987. The Vocabulary Problem in Human-System Communication. *Commun. ACM* 30, 11 (Nov. 1987), 964–971. doi:10.1145/32206.32212

[26] Yoluana Gamboa, Juan Jesús Arenas, and Freddy Paz. 2020. Usability Evaluation Towards a Cultural Perspective: A Systematic Literature Review. In *Design, User Experience, and Usability. Case Studies in Public and Personal Interactive Systems*, Aaron Marcus and Elizabeth Rosenzweig (Eds.). Vol. 12202. Springer International Publishing, Cham, 608–617. doi:10.1007/978-3-030-49757-6_44

[27] Manuel B. Garcia. 2024. Factors Affecting Adoption Intention of Productivity Software Applications Among Teachers: A Structural Equation Modeling Investigation. *International Journal of Human–Computer Interaction* 40, 10 (May 2024), 2546–2559. doi:10.1080/10447318.2022.2163565

[28] D Gentner. 1983. Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science* 7, 2 (June 1983), 155–170. doi:10.1016/S0364-0213(83)80009-3

[29] Dedre Gentner, Jeffrey Loewenstein, and Leigh Thompson. 2003. Learning and Transfer: A General Role for Analogical Encoding. *Journal of Educational Psychology* 95, 2 (June 2003), 393–408. doi:10.1037/0022-0663.95.2.393

[30] Samuel Goree, Bardia Doosti, David Crandall, and Norman Makoto Su. 2021. Investigating the Homogenization of Web Design: A Mixed-Methods Approach. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–14. doi:10.1145/3411764.3445156

[31] Jonathan Grudin. 1992. Consistency, Standards, and Formal Approaches to Interface Development and Evaluation: A Note on Wiecha, Bennett, Boies, Gould, and Greene. *ACM Transactions on Information Systems* 10, 1 (Jan. 1992), 103–111. doi:10.1145/128756.128760

[32] Philip J. Guo. 2018. Non-Native English Speakers Learning Computer Programming: Barriers, Desires, and Design Opportunities. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, Montreal QC Canada, 1–14. doi:10.1145/3173574.3173970

[33] Stuart Hallifax, Audrey Serna, Jean-Charles Marty, Guillaume Lavoué, and Elise Lavoué. 2019. Factors to Consider for Tailored Gamification. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. ACM, Barcelona Spain, 559–572. doi:10.1145/3311350.3347167

[34] Steven Hannah. 2008. Sorting Out Card Sorting: Comparing Methods for Information Architects, Usability Specialists, and Other Practitioners. (Nov. 2008).

[35] Vikas Hassija, Vinay Chamola, Atmesh Mahapatra, Abhinandan Singal, Divyansh Goel, Kaizhu Huang, Simone Scardapane, Indro Spinelli, Mufti Mahmud, and Amir Hussain. 2024. Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence. *Cognitive Computation* 16, 1 (2024), 45–74. doi:10.1007/s12559-023-10179-8

[36] Sebastian Heil, Maxim Bakaev, and Martin Gaedke. 2016. Measuring and Ensuring Similarity of User Interfaces: The Impact of Web Layout. In *Web Information Systems Engineering – WISE 2016*, Wojciech Cellary, Mohamed F. Mokbel, Jianmin Wang, Hua Wang, Rui Zhou, and Yanchun Zhang (Eds.). Vol. 10041. Springer International Publishing, Cham, 252–260. doi:10.1007/978-3-319-48740-3_18

[37] Monique M. Hennink, Bonnie N. Kaiser, and Vincent C. Marconi. 2017. Code Saturation Versus Meaning Saturation: How Many Interviews Are Enough? *Qualitative Health Research* 27, 4 (March 2017), 591–608. doi:10.1177/1049732316665344

[38] Felix Hill, Adam Santoro, David G. T. Barrett, Ari S. Morcos, and Timothy Lillicrap. 2019. Learning to Make Analogies by Contrasting Abstract Relational Structure. doi:10.48550/arXiv.1902.00120 arXiv:1902.00120 [cs]

[39] Kasper Hornbæk, Per Ola Kristensson, and Antti Oulasvirta. 2025. Introduction to user interfaces. In *Introduction to Human-Computer Interaction*. Oxford University Press. doi:10.1093/oso/9780192864543.003.0023 arXiv:https://academic.oup.com/book/0/chapter/529001532/chapter-pdf/64021648/oso-9780192864543-chapter-23.pdf

[40] Yangyu Hu, Guosheng Xu, Bowen Zhang, Kun Lai, Guoai Xu, and Miao Zhang. 2020. Robust App Clone Detection Based on Similarity of UI Structure. *IEEE Access* 8 (2020), 77142–77155. doi:10.1109/ACCESS.2020.2988400

[41] Patrycja Kałamała, Aleksandra Sadowska, Wawrzyniec Ordziniak, and Adam Chuderski. 2017. Gestalt Effects in Visual Working Memory. *Experimental Psychology* (Feb. 2017).

[42] Solène Kalénine, Françoise Bonthoux, and Anna M. Borghi. 2009. How Action and Context Priming Influence Categorization: A Developmental Study. *British Journal of Developmental Psychology* 27, 3 (Sept. 2009), 717–730. doi:10.1348/026151008X369928

[43] 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*. ACM, Boston Massachusetts USA, 67–71. doi:10.1145/22627.22350

[44] Maurice G Kendall and B Babington Smith. 1940. On the Method of Paired Comparisons. *Biometrika* 31, 3/4 (1940), 324–345.

[45] Anjali Khurana, Hariharan Subramonyam, and Parmit K Chilana. 2024. Why and When LLM-Based Assistants Can Go Wrong: Investigating the Effectiveness of Prompt-Based Interactions for Software Help-Seeking. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*. ACM, Greenville SC USA, 288–303. doi:10.1145/3640543.3645200

[46] Kimia Kiani, George Cui, Andrea Bunt, Joanna McGrenere, and Parmit K. Chilana. 2019. Beyond "One-Size-Fits-All": Understanding the Diversity in How Software Newcomers Discover and Make Use of Help Resources. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–14. doi:10.1145/3290605.3300570

[47] Andy J. King, Allison J. Lazard, and Shawna R. White. 2020. The Influence of Visual Complexity on Initial User Impressions: Testing the Persuasive Model of Web Design. *Behaviour & Information Technology* 39, 5 (May 2020), 497–510. doi:10.1080/0144929X.2019.1602167

[48] Ben Lafreniere and Tovi Grossman. 2018. Blocks-to-CAD: A Cross-Application Bridge from Minecraft to 3D Modeling. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. ACM, Berlin Germany, 637–648. doi:10.1145/3242587.3242602

[49] Kyungho Lee. 2025. Towards a Working Definition of Designing Generative User Interfaces. In *Companion Publication of the 2025 ACM Designing Interactive Systems Conference*. ACM, Funchal Portugal, 489–495. doi:10.1145/3715668.3736365

[50] Frédéric B Leloup, Michael R Pointer, Philip Dutré, and Peter Hanselaer. 2010. Geometry of Illumination, Luminance Contrast, and Gloss Perception. *JOSA A* 27, 9 (2010), 2046–2054.

[51] Clayton Lewis. 1982. *Using the" thinking-aloud" method in cognitive interface design*. IBM TJ Watson Research Center Yorktown Heights, NY.

[52] Toby Jia-Jun Li, Lindsay Popowski, Tom Mitchell, and Brad A Myers. 2021. Screen2Vec: Semantic Embedding of GUI Screens and GUI Components. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–15. doi:10.1145/3411764.3445049

[53] Andrew Lovett and Kenneth Forbus. 2017. Modeling Visual Problem Solving as Analogical Reasoning. *Psychological Review* 124, 1 (2017), 60–90. doi:10.1037/rev0000039

[54] Kevin Lynch. 1964. *The Image of the City*. MIT Press.

[55] Eva Mackamul, Géry Casiez, and Sylvain Malacria. 2024. Clarifying and differentiating discoverability. *Human–Computer Interaction* (2024), 1–26. doi:10.1080/07370024.2024.2364606

[56] Shareen Mahmud, Jessalyn Alvina, Parmit K. Chilana, Andrea Bunt, and Joanna McGrenere. 2020. Learning Through Exploration: How Children, Adults, and Older Adults Interact with a New Feature-Rich Application. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–14. doi:10.1145/3313831.3376414

[57] Dipu Manandhar, Dan Ruta, and John Collomosse. 2020. Learning Structural Similarity of User Interface Layouts Using Graph Networks. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 730–746. doi:10.1007/978-3-030-58542-6_44

[58] Rafal Mantiuk, Maria Perez-Ortiz, and Aliaksei Mikhailiuk. 2024. Pwcmp.

[59] Rafał K. Mantiuk, Anna Tomaszewska, and Radosław Mantiuk. 2012. Comparison of Four Subjective Methods for Image Quality Assessment. *Computer Graphics Forum* 31, 8 (2012), 2478–2491. doi:10.1111/j.1467-8659.2012.03188.x

[60] Joanna McGrenere, Ronald M. Baecker, and Kellogg S. Booth. 2002. An evaluation of a multiple interface design solution for bloated software. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Minneapolis, Minnesota, USA) *(CHI '02)*. Association for Computing Machinery, New York, NY, USA, 164–170. doi:10.1145/503376.503406

[61] Joanna McGrenere, Ronald M. Baecker, and Kellogg S. Booth. 2007. A Field Evaluation of an Adaptable Two-Interface Design for Feature-Rich Software. *ACM Transactions on Computer-Human Interaction* 14, 1 (May 2007), 3. doi:10.1145/1229855.1229858

[62] Joanna McGrenere and Gale Moore. 2000. Are We All in the Same" Bloat"?. In *Graphics Interface*, Vol. 2000. Graphics interface, Montreal QC Canada, 187–196.

[63] Jeremy Mendel and Richard Pak. 2009. The Effect of Interface Consistency and Cognitive Load on User Performance in an Information Search Task. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 53, 22 (Oct. 2009), 1684–1688. doi:10.1177/154193120905302206

[64] Eleni Michailidou, Simon Harper, and Sean Bechhofer. 2008. Visual Complexity and Aesthetic Perception of Web Pages. In *Proceedings of the 26th Annual ACM International Conference on Design of Communication*. ACM, Lisbon Portugal, 215–224. doi:10.1145/1456536.1456581

[65] Aliaksei Miniukovich, Simone Sulpizio, and Antonella De Angeli. 2018. Visual Complexity of Graphical User Interfaces. 1–9. doi:10.1145/3206505.3206549

[66] Michael Muller and Justin Weisz. 2023. Analogies-Based Design Using a Generative AI Application: A Play in Three Acts. In *ACM Conference on Designing Interactive Systems*.

[67] Jairo A. Navarrete-Ulloa and Maximo Trench. 2025. Transfer Across Episodes of Analogical Reasoning: The Role of Visuo-Spatial Schemas. *Journal of Cognition* 8, 1 (Jan. 2025). doi:10.5334/joc.408

[68] Jakob Nielsen. 2000. End of Web Design. https://www.nngroup.com/articles/end-of-web-design/.

[69] Cristina Olaverri-Monreal, Christoph Draxler, and Klaus-Josef Bengler. 2011. Variable Menus for the Local Adaptation of Graphical User Interfaces. In *6th Iberian Conference on Information Systems and Technologies (CISTI 2011)*. 1–6.

[70] Aude Oliva, Michael L Mack, Mochan Shrestha, and Angela Peeper. 2004. Identifying the Perceptual Dimensions of Visual Complexity of Scenes. (2004).

[71] François Osiurak, 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. *Cortex* 45, 6 (June 2009), 769–783. doi:10.1016/j.cortex.2008.06.013

[72] STEPHEN J. PAYNE. 1991. A descriptive study of mental models†. *Behaviour & Information Technology* 10, 1 (1991), 3–21. doi:10.1080/01449299108924268

[73] Maria Perez-Ortiz and Rafal K. Mantiuk. 2017. A Practical Guide and Software for Analysing Pairwise Comparison Experiments. doi:10.48550/arXiv.1712.03686 arXiv:1712.03686 [cs, stat]

[74] Maria Perez-Ortiz, Aliaksei Mikhailiuk, Emin Zerman, Vedad Hulusic, Giuseppe Valenzise, and Rafal K. Mantiuk. 2020. From Pairwise Comparisons and Rating to a Unified Quality Scale. *IEEE Transactions on Image Processing* 29 (2020), 1139–1151. doi:10.1109/TIP.2019.2936103

[75] Raphaël Perraud and Sylvain Malacria. 2025. Measuring Interface Similarity: Computing a More Perceptual Distance between Graphical User Interfaces. In *36e Conférence Internationale Francophone Sur l'Interaction Humain-Machine (IHM'25)*. Toulouse, France.

[76] Raphaël Perraud, Aurélien Tabard, and Sylvain Malacria. 2024. Tutorial Mismatches: Investigating the Frictions Due to Interface Differences When Following Software Video Tutorials. In *ACM Conference on Designing Interactive Systems (DIS 2024)*. Copenhagen, Denmark. doi:10.1145/3643834.3661511

[77] Dwight J. Peterson and Marian E. Berryhill. 2013. The Gestalt Principle of Similarity Benefits Visual Working Memory. *Psychonomic Bulletin & Review* 20, 6 (Dec. 2013), 1282–1289. doi:10.3758/s13423-013-0460-x

[78] Xiaofan Qian, Ying Yang, and Yong Gong. 2011. The Art of Metaphor: A Method for Interface Design Based on Mental Models. In *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry (VRCAI '11)*. Association for Computing Machinery, New York, NY, USA, 171–178. doi:10.1145/2087756.2087780

[79] Zhuwei Qin, Fuxun Yu, Chenchen Liu, and Xiang Chen. 2018. How Convolutional Neural Networks See the World — A Survey of Convolutional Neural Network Visualization Methods. *Mathematical Foundations of Computing* 1, 2 (May 2018), 149–180. doi:10.3934/mfc.2018008

[80] Vidya Ramesh, Charlie Hsu, Maneesh Agrawala, and Björn Hartmann. 2011. ShowMeHow: Translating User Interface Instructions between Applications. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. ACM, Santa Barbara California USA, 127–134. doi:10.1145/2047196.2047212

[81] Katharina Reinecke, Tom Yeh, Luke Miratrix, Rahmatri Mardiko, Yuechen Zhao, Jenny Liu, and Krzysztof Z. Gajos. 2013. Predicting Users' First Impressions of Website Aesthetics with a Quantification of Perceived Visual Complexity and Colorfulness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2049–2058. doi:10.1145/2470654.2481281

[82] 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*. ACM, Hamburg Germany, 1–14. doi:10.1145/3544548.3581246

[83] Cheul Rhee, Junghoon Moon, and Youngchan Choe. 2006. Web Interface Consistency in E-learning. *Online Information Review* 30, 1 (Jan. 2006), 53–69. doi:10.1108/14684520610650309

[84] Phillip Richter, Heiko Wersing, and Anna-Lisa Vollmer. 2024. Reducing Mental Model Mismatch with Intention-Based Feedback in Human-Robot Teaching. In *Proceedings of the 12th International Conference on Human-Agent Interaction*. ACM, Swansea United Kingdom, 399–401. doi:10.1145/3687272.3690896

[85] John Rieman. 1996. A Field Study of Exploratory Learning Strategies. *ACM Transactions on Computer-Human Interaction* 3, 3 (Sept. 1996), 189–218. doi:10.1145/234526.234527

[86] David E Rumelhart and Adele A Abrahamson. 1973. A Model for Analogical Reasoning. *Cognitive Psychology* 5, 1 (July 1973), 1–28. doi:10.1016/0010-0285(73)90023-6

[87] John W. Satzinger and Lorne Olfman. 1998. User Interface Consistency across End-User Applications: The Effects on Mental Models. *Journal of Management Information Systems* 14, 4 (March 1998), 167–193. doi:10.1080/07421222.1998.11518190

[88] Joey Scarr, Andy Cockburn, Carl Gutwin, and Sylvain Malacria. 2013. Testing the Robustness and Performance of Spatially Consistent Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 3139–3148. doi:10.1145/2470654.2466430

[89] Joseph Laurence Scarr. 2014. Understanding and Exploiting Spatial Memory in the Design of Efficient Command Selection Interfaces. (2014). doi:10.26021/3553

[90] Nihar B. Shah, Sivaraman Balakrishnan, Joseph Bradley, Abhay Parekh, Kannan Ramch, ran, and Martin J. Wainwright. 2016. Estimation from Pairwise Comparisons: Sharp Minimax Bounds with Topology Dependence. *Journal of Machine Learning Research* 17, 58 (2016), 1–47.

[91] Ben Shneiderman. 2010. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson Education India.

[92] Christian Sifaqui. 1999. Structuring User Interfaces with a Meta-Model of Mental Models. *Computers & Graphics* 23, 3 (June 1999), 323–330. doi:10.1016/S0097-8493(99)00041-2

[93] Donna Spencer and Jesse James Garrett. 2009. *Card Sorting: Designing Usable Categories*. Rosenfeld Media, Brooklyn, N.Y.

[94] Robert J Sternberg. 1989. Domain-Generality versus Domain-Specificity: The Life and Impending Death of a False Dichotomy. *Merrill-Palmer Quarterly (1982-)* (1989), 115–130.

[95] L. L. Thurstone. 1994. A Law of Comparative Judgment. *Psychological Review* 101, 2 (April 1994), 266–270. doi:10.1037/0033-295X.101.2.266

[96] Kashyap Todi, Jussi Jokinen, Kris Luyten, and Antti Oulasvirta. 2018. Familiarisation: Restructuring Layouts with Visual Learning Models. In *23rd International Conference on Intelligent User Interfaces*. ACM, Tokyo Japan, 547–558. doi:10.1145/3172944.3172949

[97] Máximo Trench and Ricardo A. Minervino. 2015. The Role of Surface Similarity in Analogical Retrieval: Bridging the Gap Between the Naturalistic and the Experimental Traditions. *Cognitive Science* 39, 6 (Aug. 2015), 1292–1319. doi:10.1111/cogs.12201

[98] Kristi Tsukida, Maya R Gupta, et al. 2011. How to Analyze Paired Comparison Data. *Department of Electrical Engineering University of Washington, Tech. Rep. UWEETR-2011-0004* 1 (2011).

[99] Thomas Tullis and Larry Wood. 2004. How Many Users Are Enough for a Card-Sorting Study?. In *Usability Professionals Association (UPA)*. Minneapolis.

[100] Md. Sami Uddin. 2016. Use of Landmarks to Design Large and Efficient Command Interfaces. In *Proceedings of the 2016 ACM Companion on Interactive Surfaces and Spaces (ISS '16 Companion)*. Association for Computing Machinery, New York, NY, USA, 13–17. doi:10.1145/3009939.3009942

[101] Md. Sami Uddin, Carl Gutwin, and Andy Cockburn. 2017. The Effects of Artificial Landmarks on Learning and Performance in Spatial-Memory Interfaces. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3843–3855. doi:10.1145/3025453.3025497

[102] Md. Sami Uddin, Carl Gutwin, and Alix Goguey. 2017. Using Artificial Landmarks to Improve Revisitation Performance and Spatial Learning in Linear Control Widgets. In *Proceedings of the 5th Symposium on Spatial User Interaction (SUI '17)*. Association for Computing Machinery, New York, NY, USA, 48–57. doi:10.1145/3131277.3132184

[103] Sami Uddin and Carl Gutwin. 2021. The Image of the Interface: How People Use Landmarks to Develop Spatial Memory of Commands in Graphical Interfaces. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–17. doi:10.1145/3411764.3445050

[104] Francesco Vitale, Joanna McGrenere, Aurélien Tabard, Michel Beaudouin-Lafon, and Wendy E. Mackay. 2017. High Costs and Small Benefits: A Field Study of How Users Experience Operating System Upgrades. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, Denver Colorado USA, 4242–4253. doi:10.1145/3025453.3025509

[105] Yannick Wamain, Ewa Pluciennicka, and Solène Kalénine. 2015. A Saw Is First Identified as an Object Used on Wood: ERP Evidence for Temporal Differences between Thematic and Functional Similarity Relations. *Neuropsychologia* 71 (May 2015), 28–37. doi:10.1016/j.neuropsychologia.2015.02.034

[106] Pengcheng Wang, Zefeng Bai, Kambiz Saffarizadeh, and Chuang Wang. 2025. The Impact of App Updates on Usage Frequency and Duration. *Communications of the Association for Information Systems* 57, 1 (2025), 78. https://aisel.aisnet.org/cais/vol57/iss1/78 Accessed 2025-12-02.

[107] Jason Wu, Xiaoyi Zhang, Jeff Nichols, and Jeffrey P Bigham. 2021. Screen Parsing: Towards Reverse Engineering of UI Models from Screenshots. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. ACM, Virtual Event USA, 470–483. doi:10.1145/3472749.3474763

[108] Xin Xia, David Lo, Feng Zhu, Xinyu Wang, and Bo Zhou. 2013. Software Internationalization and Localization: An Industrial Experience. In *2013 18th International Conference on Engineering of Complex Computer Systems*. 222–231. doi:10.1109/ICECCS.2013.40

[109] Siriporn Yamnill and Gary N. McLean. 2001. Theories Supporting Transfer of Training. *Human Resource Development Quarterly* 12, 2 (2001), 195–208. doi:10.1002/hrdq.7

[110] Lingxiao Yang, Hongzhi You, Zonglei Zhen, Dahui Wang, Xiaohong Wan, Xiaohua Xie, and Ru-Yuan Zhang. 2023. Neural Prediction Errors Enable Analogical Visual Reasoning in Human Standard Intelligence Tests. In *Proceedings of the 40th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 202)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.). PMLR, 39572–39583.

[111] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, UT, 586–595. doi:10.1109/CVPR.2018.00068