Screening Prototype Features in Terms of Intuitive Use: Design Considerations and Proof of Concept

SANDRINE FISCHER, MAKOTO ITOH AND TOSHIYUKI INAGAKI

Laboratory for Cognitive Systems Science, Laboratory of Advanced Research B #0826, Department of Risk Engineering, University of Tsukuba, 1-1-1 Tennoudai, Tsukuba-shi, Ibaraki-ken 305-8573, Japan
*Corresponding author: fischer@css.risk.tsukuba.ac.jp

Graphical interface use involves schemata operations that range from transfer to induction. The former apply existing knowledge, such as prior schemata, and are effortless, preconscious and intuitive. The latter, which consist in constructing new schemata, are resource-consuming and thus detrimental to intuitive use (IU). A quantitative method is proposed to manipulate and screen schemata operations at the level of an interface's states and features. Relevance for the design cycle of innovative interfaces is critically reviewed, and integration with existing intuitive-use design frameworks is proposed. These considerations are built upon instructional design studies suggesting that assessment should precede and inform the application of design techniques geared toward IU.

RESEARCH HIGHLIGHTS

• Recent design frameworks claim that intuitive use of a device is improved by increasing the familiarity of its features.
• Instructional design studies warn that increased familiarity can be ineffective in cases where people already possess relevant knowledge schemata.
• Incidences of such risk in intuitive-use design are reviewed, along with their prevalence in existing studies of intuitive use.
• An experimental and quantitative method for addressing this issue during the design cycle of innovative interfaces is outlined and illustrated.
• This method screens schemata operations at the level of interface states and features, which allows redesign needs to be pinpointed.

Keywords: Intuitive use; Interaction design theory, concepts, and paradigms; Quantitative methods; User studies; Graphical user interfaces

Received 9 April 2014; Revised 12 December 2014; Accepted 2 January 2015

1. INTRODUCTION

A device is ‘intuitive’ when it may be used as intended, without apparent effort, training or outside help. Recent research has shown that intuitive use (IU) is based on prior familiarity (Blackler et al., 2004; see also Blackler, 2008). Design frameworks based on this finding recommend that device familiarity be increased by thoroughly inspecting users’ past experiences and habits (Blackler et al., 2007, 2014; Loeffler et al., 2013). This strategy disregards numerous instructional design studies that found no benefit in increasing familiarity when people already possess relevant knowledge schemata (Kalyuga, 2007; Mayer, 2001; Sweller et al., 2003). Such studies may explain why several IU and HCI studies have failed to record any usage improvement from increased familiarity. For example, Hurtienne et al. (2013) found that redesigning a ticket vending machine through familiarity principles improved neither its overall usage nor satisfaction. Admittedly, the functionality of a ticket vending machine is well known to most people. Such a case suggests that familiarity-oriented design should focus only on features (e.g. menu, options) incompatible with prior schemata while skipping features that are well known and already compatible with prior schemata. The motivation for
this strategy is not only to save time in development but also to ensure that redesign efforts translate into tangible benefits.

Modern (e.g. agile) development frameworks recommend that products be specified, developed and tested through short iterations (Ceret et al., 2013). Market characterization (both for purposes of inspiration and differentiation) and user profiling (habits and capabilities) permit early specification of interactive wireframes or prototypes. User tests can then be deployed for identifying ‘pain points’ and determining which features of the prototype require redesign. Such tests may serve to screen the compatibility of a prototype’s features with users’ prior schemata. A quantitative operationalization of this rationale, which consists in manipulating users’ schemata, is presented in Fischer et al. (2014). The present paper illustrates the bottom line of this approach for user testing. We focus on the function of graphical interfaces, namely the purpose of their features (e.g. print a document, geo-sync a to-do list). In Section 2.1, we outline existing IU design frameworks. The primary goal of our paper is presented Sections 2.2 and 2.3, where we review why design techniques entailed by these frameworks are not likely to be effective for features already compatible with users’ prior schemata. This argument is built upon instructional design studies that should be brought to the attention of practitioners from HCI that may promote IU (e.g. stereotypes, metaphors, application of this method, which is carried out for a multistate consistency). In Blackler et al. (2007), these design techniques employed by these frameworks are not likely to be effective for features already compatible with users’ prior schemata. This argument is built upon instructional design studies that should be brought to the attention of practitioners from HCI that may promote IU (e.g. stereotypes, metaphors, application of this method, which is carried out for a multistate consistency). In Blackler et al. (2007), designers can enter a spiral at ‘a suitable place’ to identify and improve the features of a device.

Stage 2 (familiarity inspection) aims to extract body reflectors, stereotypes, familiar appearances, functions, locations, affordances and metaphors relevant to the intended user population. This stage combines products reviews, literature reviews, user interviews and usability techniques (e.g. naturalistic observations, participatory design, etc.). Stage 3 (design) is structured so that domain stereotypes (e.g. familiar labels from the domain), metaphors and affordances (e.g. familiar things susceptible to transfer from other domains) and redundancy and consistency can be applied (Blackler et al., 2007). Although not explicitly mentioned, it is expected that user testing occurs after Stage 3. The spiral tool has been deployed for the design and redesign of feature appearance and location for a remote controller, microwave oven and MP3 player (Blackler et al., 2007, 2010, 2014).

Hurtienne (2009) showed that sensorimotor knowledge abstractions, called image schemas, support fast and correct mappings between the layouts and functions of an interface. For example, the image schema UP-DOWN is relevant to designing a volume control or attractiveness meter. Hurtienne’s Image Schema CATalog1 links dozens of these image schemas to hundreds of metaphorical extensions (e.g. UP = ‘louder’) and user interfaces. One design framework that employs these image schemas is the ‘IBIS’ method (German for ‘design of IV with image schemas’, Loeffler et al., 2013), whereby practitioners:

Stage 1: select end-users for systems similar to the one being designed,
Stage 2: conduct surveys and contextual enquiries in which the end-users use the system to be redesigned under thinking-aloud conditions; extract image schemas,
Stage 3: apply these image schemas to specifying a prototype,
Stage 4: evaluate the prototype by means of a questionnaire toolbox.

Stage 2 (contextual inquiry) aims at collecting verbal protocols from which an image schema expert extracts users’ image schemata and determines relevant design metaphors. In Loeffler et al. (2013), the IBIS method was deployed to redesign the layout of an image browser and product order manager. Both the spiral tool and IBIS method place inspection of users’ familiarities and habits ahead of any actual design and testing. This strategy has a cost. One of the student designers who tested Blackler et al.’s framework (2007) reported that he spent ‘a great deal more time investigating and analyzing the intended users than he would otherwise’ (p. 8). One partner who applied Loeffler et al.‘s framework (2013) also reported that ‘the time required to plan, conduct and transcribe the contextual inquiries was strongly underestimated’ (p. 8). In addition, neither framework informs the practitioners as to which features actually require inspection and redesign. For Blackler et al. (2007), designers can enter a spiral at ‘a suitable point and leave it when necessary’ (p. 8), while for Loeffler et al. (2013) ‘image-schematic metaphors that are suitable for


---

INTERACTING WITH COMPUTERS, Vol. 27 No. 3, 2015
the implementation are prioritized and afterwards mapped to
design techniques support
are the risks that inspection and implementation of user habits
do not translate into substantial improvements.

2.2. Risks of ineffective design

Instructional design is the practice of creating experiences that
render the acquisition of knowledge and skill more
effective and appealing. Researchers in this field
investigate the extent to which instructional techniques support
the construction of rich and versatile knowledge. In particular,
design techniques are not effective when applied to content that
is either too innovative or too familiar. When content is too
innovative, people lack the minimum knowledge required to
assimilate and elaborate upon it. In such cases, we say that
instructional design is inoperative, meaning it fails to support
knowledge construction (see Fig. 1). A sensible interpretation of
this first risk is that completely novel concepts—even when well
formulated—tend to be ignored or falsely interpreted, leading to
an incomplete or skewed understanding (Chalmer, 2003).

A second risk is that techniques that are typically beneficial
for new content (Borgman, 1999; Hsu, 2006; Mayer, 1976)
lose their efficacy when content is too simple or familiar
(Hsu, 2005, 2006; Mayer, 1999; Ozgunor and Guthrie, 2004;
Pollock et al., 2002). This phenomenon is revealed by factor
interactions between prior knowledge (e.g. experts vs. novices)
and the techniques advocated in the IU literature, which
include metaphors, stereotypes and redundancy. Metaphors are
considered essential to IU, since they represent new concepts in
terms of familiar ones (Hurtienne and Blackler, 2007; Loefller
et al., 2013; O’Brien et al., 2010). However, consider a familiar
concept F, such as a warehouse, used as a metaphor for
understanding a technical concept T, such as a database. The
metaphor conveyed by F provides little benefit to a database
programmer who already possesses prior knowledge of T.
In fact, instructional design research has shown that while
metaphors may assist novices in assimilating concepts, they
are useless for experts (Hsu, 2006; Mayer, 1999; Mayer, 2001).

Population stereotypes should support IU design since they
reflect cultural conventions (Blackler, 2008; Blackler and
Hurtienne, 2007). However, when it comes to labeling concepts,
Furnas et al. (1987) consider the existence of stereotypes a
myth. They found that the likelihood of two people labeling
a concept with identical keywords was low (7-18%). Keyword
agreement was also low (~33%) for commonly shared topics
(e.g. keywords proposed by expert cooks to novice and expert
cooks). The same study showed a satisfactory agreement
(50-100%) only when several words were combined to label a
single concept. In fact, expertise (or shared knowledge), which
is indispensable to vocabulary stereotypes, entails multiple
semantic associations that in turn override stereotypes. This
enables experts to easily resolve ambiguities in vocabulary.

Table 1. Factor interaction between prior knowledge (expertise) and
redesign.

| Novices perform better after design improvement |
| Experts perform equally well regardless of design improvement |
less well when the vocabulary was integrated. This phenomenon is particularly detrimental since aside from wasting time attempting to improve something that requires no improvement, redesign actually causes performance deterioration. These two forms of factor interaction are illustrated in Fig. 1.

Both forms of the interaction are conceptualized in terms of cognitive load theory and schemata. Schemata designate the rich, goal-oriented and abstract knowledge representations possessed by experts, yet not by novices (Chi et al., 1981, 1982; Sweller, 2003, 2004). Schemata ’chunk’ the processing of information in working memory, which renders processing automatic (Shiffrin and Schneider, 1977; Sweller et al., 2003; van Merriënboer and Sweller, 2005). As a schema is applied ‘as is’, or ‘transferred’, its structure overides the analysis of information and results in minimal effort for comprehension and decision to proceed. This is why experts do not need guidance to intuit and excel at a variety of tasks. Nevertheless, if guidance is added that experts cannot refrain from processing, redesign can result in greater cognitive load and decreased performance (Swellet et al., 2003).

2.3. Incidence in IU research

Instructional design studies do not imply that experts knew the tested material, but that their knowledge schemata were sufficiently abstract for subsuming the material. A schema represents the principle(s) common to several experiences at the expense of their specificity. The learning process by which experiences are abstracted into a single schema requires either time or effort. For example, the first time one discovered how to click an icon likely resulted in a specific representation. If exposed to only that one icon, other icons would require effort and deliberation to process. Only by clicking more icons, in different contexts and for other purposes, would one abstract a common schema that subsumes all such experiences past and future. Any icon would then be automatically processed through that same schema. In other words, knowledge schemata are abstract and transferable to a variety of instances (Rumelhart and Ortony, 1977). This argument, which was demonstrated by Gick and Holyoak (1983), Hintzman (1986) and Reber (1967), seems applicable to IU. Blackler et al. (2010) found more IUs from participants with a broad technological background than from those familiar with only a certain type of device. Hurtienne et al. (2013) showed that usability reflects prior experience with various device types rather than one. O’Brien et al. (2012) confirmed through naturalistic analysis that prior knowledge associated with the use of everyday devices has multiple sources. Fischer et al. (2015) found feature functions familiar from other devices and domains to yield the same transfer operation.

We argue along these lines, and the risks outlined in Section 2.2, that intuitive features are already compatible with prior schemata and should not benefit from redesign. Several findings in IU literature support this argument. Hurtienne et al. (2013) redesigned a ticket-vending machine through metaphors, familiar tabs, clear labels, etc. Contrary to expectation, usage performances and satisfaction did not improve. Perhaps schemata related to purchasing were sufficiently generic to override a variety of instantiations, including the redesign of a vending machine. However, Hurtienne et al. found that participants’ prior experience with technology had a greater effect for the original machine than the redesigned one. This outcome resembles the factor interaction in Table 1, which indicates that redesign has a greater effect on participants with low experience rather than high. Such selective effects of redesign, though, are likely to be hindered when participants with different technology experiences are pooled together.

Two other examples of usage accuracy not benefiting from redesign include Gudur et al. (2013), who applied a simpler menu structure to a pet-sitting game, and Blackler et al. (2014), who applied their spiral tool to the features of a microwave oven. Neither design strategy resulted in improvement of the device’s ‘correct uses’ by participants. The number of ‘IUs’ coded at the level of participants’ think-aloud protocols increased, though. Their coding employed Blackler’s five heuristics (2008): verbalized expectations, certainty of correctness, reference to past experience, absence of verbalized reasoning and latency. Interestingly, such a selective effect of design on verbalizations, yet not on accuracy, seems imputable to the schema construct. As discussed by Camp et al. (2001), gradual acquisition of domain schemata (e.g. by education or training) causes a shift in performance criterion from accuracy to speed to automaticity.

Novices must rely on short-term memory to interpret, form and test hypotheses about the information they encounter (for classical accounts of automaticity, see Anderson, 1993; Posner and Snyder, 1975). In such cases, accuracy is a primary criterion of performance (van Merriënboer, 1997) and verbalizations tend

![Figure 1. Effectiveness of design techniques (solid) and the expertise reversal effect (dashed), as a function of prior knowledge. The vertical line (dotted) represents the threshold of prior knowledge beyond which one is considered an expert. At lowest levels of prior knowledge, design techniques are ineffective. At intermediary levels of prior knowledge, they become beneficial. At highest levels of prior knowledge (e.g. experts), the content is intuitive and design techniques tend to be either ineffective (solid) or counter-effective (dashed).](image-url)
Table 2. Factor interaction between prior knowledge (feature familiarity) and redesign.

- Features not compatible with prior schemata are used more correctly after design improvement
- Features compatible with prior schemata are used equally correctly regardless of design improvement

to be detailed (Chi et al., 1981). As domain rules are acquired in long-term memory, performance becomes more accurate. This initially results in improvement of speed (e.g. decision time, completion time), and then development of automaticity (e.g. dual task performance), which ultimately renders verbalizations less detailed and more principle-oriented (Chi et al., 1981, Posner and Snyder, 1975). From this perspective, the design of simple functions (e.g. feeding a pet, heating food, purchasing a ticket) would seem better gauged by speed and verbal protocols rather than accuracy. Conversely, accuracy seems more sensible for assessing the design of innovative functions and cutting-edge technology. Altogether, the fact that redesign modulated the five coding heuristics for Gudur et al. (2013) and Blackler et al. (2014), yet did not increase accuracy, likely stems from the tested devices being fairly common and having basic functionality. Regardless of the criterion, and without a means to diagnose the innovativeness of a device, UI-oriented techniques run the risk of being ineffective.

3. AN EXPERIMENTAL APPROACH TO SCREENING SCHEMATA OPERATIONS

3.1. Operationalization considerations

We derived Table 1 from instructional design studies that controlled prior knowledge as an inter-individual factor (i.e. novices and experts are different individuals of an experiment). Interactive products tend to mix common and innovative features, though, so that prior knowledge acts as an intra-individual and intra-material factor. Table 1 can thus be reformulated into Table 2.

In order to operationalize Table 2, one could redesign an interface for IU and conduct user tests that compare original and redesigned versions as a function of features' prior familiarity. Since the redesign of an entire interface can be overkill, though, one could instead manipulate participants’ representation of the original interface through instructions. As for design manipulations, the effectiveness of instructions tailored to the construction of new schemata depends on prior knowledge (Table 3).

As indicated by the dashed cells in Table 3, instructional manipulation allows factor interaction to be addressed with just one version of the material and two experimental groups: one group whose instructions are tailored for construction of new schemata, and one whose instructions are not (control).

3.2. Schemata operations and patterns

In instructional design, a variety of manipulations have been devised that support the construction, assimilation or learning of new schemata. Many, such as taking notes vs. listening (Peper and Mayer, 1986), are too educationally oriented for HCI settings. Some manipulations (e.g. Mayer, 1980), however, present traits that are ecological and transposable to HCI. Notably, one could capture screenshots of the states (i.e. distinct feature configurations displayed on the screen) for a graphical user interface and make several groups study these screenshots prior to its use. In this way, Fischer et al. (2014) manipulated amendment of prior schemata and induction of new schemata via three instructional groups:

- a control group, who used the interface without studying its screenshots,
- a reading group, who used the interface after studying every screenshot through ‘word matching’,
- an induction group, who used the interface after studying every screenshot through ‘function matching’.

Word and function matching followed the procedure illustrated in Fig. 2: participants were presented with a clue (word or function), then a screenshot of the interface, and then gave a yes/no judgment based on whether the clue matched the screenshot. Word clues corresponded to a single word appearing in a screenshot. Thus, word matching required that screenshots be skimmed at a lexical level, which is not likely to support induction of new schemata (for classic evidence that schema induction is not supported by reading, see Gick and Holyoak, 1980; Mayer, 1980). Function clues were sentences that described, in the form of a concrete task, a functional

![Figure 2. The clue-screenshot matching procedure used for manipulating schema operations.](interacting-with-computers-vol-27-no-3-2015)
Table 3. Examples of design manipulations, instructional manipulations and their pattern of interaction with prior knowledge.

<table>
<thead>
<tr>
<th>Prior knowledge control</th>
<th>Experimental manipulation</th>
<th>Performance pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Novices, innovative features, etc.</td>
<td>Design (an original and a redesigned version of the material)</td>
<td>Control group: study conditions such as reading, listening, summarizing</td>
</tr>
<tr>
<td>Vs. Experts (Kalyuga and colleagues; McNamara and colleagues), features familiar from other devices and domains (Fischer et al., 2015), etc.</td>
<td>Original design</td>
<td>Equally good performances, regardless of experimental manipulation (e.g., original vs. improved design, or control vs. schema-tailored groups), indicate transfer of prior knowledge.</td>
</tr>
<tr>
<td>Vs. Design improved by simplifying (Hsu, 2005; Mayer, 1976; Borgman, 1999), adding metaphors (Hsu, 2006), adding explanations (Mayer, 1999), suppressing ambiguity (McNamara and McDaniel, 2004), increasing coherence (McNamara, 2001; McNamara and Kintsch, 1996b; Orgungor, 2004), increasing redundancy (Kalyuga et al., 2000, Mayer, 1999), etc.</td>
<td>Vs. Schema-tailored group: study conditions such as note-taking (Peper and Mayer, 1986), comparison (Gick and Holyoak, 1983; Gentner et al., 2003), comparison within an elaborative context (Mayer, 1980; Fischer et al., 2009), etc.</td>
<td>Better performance with the improved design (relative to the original), or from the schema-tailored group (relative to the control), characterizes a construction of new schemata.</td>
</tr>
</tbody>
</table>

Matching, their performances would be as poor as the control group. This pattern, called inoperative induction, is likely due to participants being too naive about a domain to benefit from schema-tailored instructions (cf. Fig. 1; Chalmers and Humphreys, 2003). Differences among group performances are more pronounced when schema manipulation is more effective. Features compatible with prior schemata, yet that require some amendments, should affect the reading group relative to the control. Based on classical accounts of schema abstraction (Reber, 1989) and schema theory (Rumelhart and Norman, 1978), we posit that word matching allows for moderately resource-demanding schemata amendment operations, such as abstraction of recurring appearance attributes (e.g., layouts, labels), accretion and tuning of existing schemata. Note, however, that the construct validity and practicality of this pattern has yet to be substantiated. Features whose function does not fall under an existing schema yet still is assimilable should affect the induction group relative to the control. In such cases, the induction group is expected to outperform the control group. This pattern, called positive schema induction, occurs when features have innovative functions whose underlying schemata can be induced through function matching (Fischer et al., 2015).

The aforementioned patterns, and schemata operations they represent, are listed in Table 4. This table relates our

---

**Note:** The table and text continue with further details and analysis regarding the interaction between design and instructional manipulations and their effect on performance based on prior knowledge. The text also discusses the relationship between features in screenshots and the induction of new schemata, highlighting the importance of resource-demanding schemata amendment operations for effective learning.
manipulations to the performance and design considerations summarized in Tables 1–3, and highlights their relevance to the screening of an interface for redesign.

The proposed screening comes down to determining the extent to which the performances of the reading and induction groups improved relative to the control group. By calculating the average values and variances of these usage performances, differences can be quantified and compared through their effect sizes (Fischer et al., 2014). The effect size is a statistic used to estimate the magnitude of a difference between two groups (generally a control and manipulated group). In other words, it constitutes a standard measure for the practical significance of a manipulation (Kelley and Preacher, 2012; McGough and Faroone, 2009). The application of effect size-based screening to a schema manipulation experiment is outlined below.

3.3. Method overview

This section presents a method applicable to multistate interfaces (or interactive wireframes, prototypes, etc.). By ‘multistate’, we mean that the interface may have hundreds of features, of which several at a time are displayed on a screen in configurations called states. The method begins with an experiment consisting of a study and usage phase, shown in the upper part of Fig. 3 and further described in Section 4.2. The study phase requires a stimulus presentation module, while the usage phase requires that participants’ actions on the interface be logged. These log files are then parsed and analyzed, as shown in the lower part of Fig. 3.

To prepare the study phase, a screenshot is taken for each state of the interface. The practitioner must generate word and function clues that match half the screenshots and mismatch the other half. Mismatching word clues correspond to a word that does not appear in the state, while mismatching function clues correspond to a function that subtly mismatches the screenshot’s features. A stimulus presentation module2 (labeled ‘Module’ in Fig. 3) is needed to display the clues, screenshots and buttons labeled ‘YES’, ‘NO’ and ‘GO’. As illustrated in Fig. 2, the module must display a clue for a predefined time (e.g. 15 s), or until participants click ‘GO’, after which the screenshot to be matched is displayed. Participants give their matching judgments by clicking the ‘YES’ or ‘NO’ buttons.

The usage phase is prepared by defining a scenario of usage tasks to be performed on the interface. Each usage task requires participants to navigate from a current state to a goal. This activity consists of recursively determining which feature in the current state best reduces the distance to the goal. Because such determination is prone to error and correction, accuracy is inversely proportional to the number of features selected or states explored during a given usage task. While one may count these actions by hand, such as from a video, a more convenient way is to automatically log them from the interface. To this effect, we employed a toolkit called Automatic Mental Model Evaluator (AMME; Rauterberg, 1993), as well as Visual Basic for Applications (VBA) macros. With an interface I/O description file (i.e. a listing of the interface’s features, commands and feature-command-feature triplets), AMME automatically formats the command actions stored in a log file into matrices and action sequences.3 These outputs are then parsed by VBA macros that count participants’ actions

---

Table 4. Correspondences between experimental groups, schema manipulations, patterns and redesign.

<table>
<thead>
<tr>
<th>Study phase</th>
<th>Screenshot encoding</th>
<th>Schemata manipulation</th>
<th>Patterns</th>
<th>Diagnosis and redesign strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control group</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>Prior schemata transfer if usage performances are as good as the reading and induction groups</td>
</tr>
<tr>
<td>Reading group</td>
<td>Word matching</td>
<td>Word-level Amendment</td>
<td>Prior schemata amendment if performances are better than the control group</td>
<td>State or feature is not intuitive and required shallow learning from participants. Redesign is desirable</td>
</tr>
<tr>
<td>Induction group</td>
<td>Function matching</td>
<td>Function-level Induction</td>
<td>Positive schemata induction if performances are better than the control group</td>
<td>State or feature is not intuitive and required deep learning. Redesign is required</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Inoperative schemata induction if performances are as poor as the control group</td>
<td>State or feature is not intuitive and required deep learning that did not occur. Redesign is required</td>
</tr>
</tbody>
</table>

---

2This module may either be handwritten code (e.g. Matlab, C#) or a software package (e.g. Psyscope, e-Prime) that presents stimuli and records participants’ responses.

3Details available on July 2014 at http://www.idemployee.id.tue.nl/g.w.r.m.rauterberg/amme.html.
by feature, state, or usage task, which then serve to screen performances at any of these granularities.

The screening process begins by calculating mean values and variances of each group’s performances. For a given granularity (e.g., data points corresponding either to features, states or usage tasks), effect sizes can be determined in terms of Cohen’s $d$, which is a standardized mean difference. With the control group as a baseline, $d_{\text{amendment}}$ is calculated between the control and reading groups, and $d_{\text{induction}}$ between the control and induction groups. The values for $d_{\text{induction}}$ vs. $d_{\text{amendment}}$ are then plotted. As a rule of thumb for interpreting effect sizes, Cohen (1988) considered a $d$ of 0.2 as small, 0.5 as medium and 0.8 as large. Based on these values, it is reasonable to consider the effect size as moderate-to-large for states having $d > 0.4$. Values that fall within the range $-0.4 < d < 0.4$ correspond either to transfer or ineptive induction, meaning the control and induction groups performed equally well or equally poorly, respectively. To distinguish between these two cases, the control group’s best and worst performances, or lowest and highest action counts, should be represented with different colors/markers in this region of the 2D scatterplot (see ‘State screening’ in Fig. 3, and also Fig. 5). Determination of these best/worst cutoff values depends on data distribution, project needs and available resources. In the following section, we showcase this screening for a concrete prototype interface. This proof of concept conveys the capacity of our screening

\[4\text{The Cohen’s }d\text{ for two distributions is the difference of their means divided by their pooled standard deviation.}\]
method to provide interface analytics that support redesign strategies at any desired level of granularity.

4. PROOF OF CONCEPT

4.1. Tested interface

The method developed above is applied to an automobile onboard computer prototype. Onboard computers aggregate multiple services into a multistate interface whose I/O structure is hidden, complex and likely to puzzle novice users. They place enormous emphasis on function attributes by integrating vehicle, infrastructure and satellite technologies with entertainment and telecommunication services. Different brands and models of onboard computers may vary not only in their technological services, but also in the way these services are presented, labeled and controlled. As a result, even with a simple rental car one may encounter functions too innovative to be understood on the spot.

Our experiment used an onboard computer prototype called DoIT#. The features of this prototype were determined by review of existing onboard computers, along with analysis of R&D trends (for details about the prototype specification, see Fischer, 2010), and they range from common (e.g. dial a call, defrost the windshield) to speculative (e.g. wireless retrieval of advertisements). DoIT# was interfaced with a state window and command panel, and comprised 75 states similar to the one illustrated in the upper part of Fig. 4.

The interaction principle underlying DoIT# was that of a hierarchical navigation, commonly found not only in onboard computers, but also in web sites, smart-phone apps, etc. In Fig. 4, the labels ‘Vehicle Security Functions’, ‘Car checking’, etc., correspond to distinct features that, when selected (viz. ENTER), change the current state into another state displaying its own features. Features are selected by clicking on the command panel with a mouse. These clicks were recorded in a log file, counted using AMME and VBA macros, and then averaged to compute effect sizes. Since participants clicked on features, which were embedded in states, and states were explored during tasks, their performances could be analyzed at any of one of these granularity levels.

4.2. Participants and procedure

This analysis extends a previous study conducted on 30 Japanese students at the University of Tsukuba, Japan (Fischer et al., 2014). Prior research has shown that IU is moderated by age and technological familiarity, yet not by gender (Blackler, 2008; Blackler et al., 2010). Thus, while it is essential to recruit participants from the same age bracket (e.g. 18–39) and technological background, gender can be considered irrelevant. In our case, a preliminary survey was used to select participants who possessed a driver’s license, did not study Computer Science and had never used an onboard computer or (global positioning system) GPS navigation system. Age of participants ranged from 18 to 27 years, and gender did not play a role in our selection.

The experiment was administered by slideshow, which participants browsed at their own pace. Participants’ cognitive style was assessed by means of the Rational Experiential Inventory (Epstein and Pacini, 1999; Naitou et al., 2004; for details see Fischer et al., 2015) and evenly matched across the experimental groups. Participants in the control group proceeded directly to the usage phase. Participants in the reading and induction groups viewed an animated slide describing word or function matching, after which they matched the screenshots of DoIT#. For all the three groups, the usage phase was introduced by a brief explanatory slide. An experimenter then intervened to demonstrate two basic usage tasks and provide each participant a handout listing 10 usage tasks to perform on their own. The tasks included (i) displaying the number of kilometers traveled, (ii) requesting the route to the house of a friend whose contact information is in the address book, (iii) requesting a route that avoids tolls, (iv) displaying the navigation directions in the rear-view mirror, (v) setting the temperature to 18°C, (vi) activating a sleeping alert, (vii) activating assistance for passing cars, (viii) activating the internal air filter, (ix) setting the ventilation to silent mode and (x) calculating rest time during the trip. In order to simulate realistic usage conditions, no performance requirements were specified, and no success feedback was provided. Participants who could not finish a task notified the experimenter, who directed them to start the next task.

4.3. State and feature screening

Effect sizes can be calculated for any level of granularity (e.g. per state, feature or usage task), any subset of data (e.g. all usage
States in the low-effect size region (red dotted box) yield either an inoperative induction pattern (red squares) or transfer pattern (green diamonds).

Figure 5. State screening in terms of $d_{\text{induction}}$ (greater than $-0.4$) and $d_{\text{amendment}}$. States in the low-effect size region (red dotted box) yield either an inoperative induction pattern (red squares) or transfer pattern (green diamonds).

Figure 6. Detailed state view and feature screening for the 'Driving assistance' state (right). Red (dark) is the value of $d_{\text{induction}}$ and blue (light) is the value of $d_{\text{amendment}}$ for each corresponding feature on the left.

Figure 7. Detailed state view and feature screening for the 'Guidance' state (right). Red (dark) is the value of $d_{\text{induction}}$ and blue (light) is the value of $d_{\text{amendment}}$ for each corresponding feature on the left.

States with large values of $d_{\text{induction}}$ typically require some redesign: they yielded errors for the control group that were overcome only by the induction group, thanks to the schemata induced by the latter. Since each state consisted of several features, it might be useful to screen an interface at the level of its features. By plotting DoIT#'s features in the same manner as its states, one may prioritize the redesign of features yielding problematic patterns, such as inoperative and positive induction.

Alternatively, one could use state scatterplots such as Fig. 5 to instead select the states exhibiting these patterns, and then screen the corresponding features. Screenshots for two of these states, indicated in Fig. 5, are displayed in Figs. 6 and 7 alongside the Cohen's $d$ values for their features. In these figures, light blue corresponds to prior schemata amendment and dark red to new schemata induction.

Figure 6 displays the state titled 'Driving assistance'. The third feature from the top, ‘speeding alert’, yielded a negative amendment pattern, suggesting its function was fairly

---

6 One may limit the analysis only to usage tasks that participants successfully completed. Indeed, participants may not have completed certain tasks because they did not understand the assignment. In experimental psychology, where the purpose is to understand the mechanism of cognition, focus is often on tasks completed successfully. In usability settings that intend to evaluate a prototype, however, one would consider any case where the prototype caused difficulty for participants, and therefore analyze both successful and unsuccessful tasks.

7 Patterns of negative induction and/or amendment may reflect an inconsistency between the prototype's information architecture and users’
compatible with the control group’s prior schemata. The first, second and fifth features yielded induction patterns, suggesting their functions challenged participants’ prior schemata. These features, in particular lane change alert, required our participants to induce new schemata, and may thus prove challenging for actual users to assimilate.

Figure 7 displays the state titled ‘Guidance’. The two first features from the top, itinerary details and trip statistics, and the last feature, pause, yielded positive induction patterns, which suggests that their functions challenged participants’ prior schemata. However pause, being a common-sounding feature, would normally not be expected to yield an induction pattern. By scrutinizing our data at the level of usage tasks, we identified that participants in the control and reading groups mistook the GPS navigation’s pause feature for a calculation of the time at rest, which is a separate feature of the onboard computer. Participants in the induction group did not make this mistake, though, which suggests that function matching helped them to resolve the functional difference between pausing the GPS and displaying the time spent at rest during a trip. This example illustrates how schemata manipulations can complement classical user tests. A test of one control group would only have shown that features with apparent similarities, such as pause and time-at-rest, can be mistaken. Only by adding a reading and induction group can the severity of this mistake be captured, namely whether the confusion is resolved through amendment of prior schemata or induction of new schemata.

Both the state scatterplot (Fig. 5) and detailed state view (Figs. 6 and 7) correspond to steps for screening an interface in terms of amendment and induction operations. Thorough assessment of an interface may require a successive layering of such screenings in a diagnostic fashion. This rationale is not unlike the way web analytics go beyond simple averages of website visits or traffic. Critical business decisions can be made when traffic is analyzed at finer granularities since it becomes possible to determine the impact of specific marketing campaigns and other events. Of course, averages of each group performance are needed for assessing our experimental manipulation. However, by considering the magnitude of these manipulations (viz. effect sizes) at finer granularities, patterns of schemata operations can be diagnosed for individual states and features of the interface. This can considerably facilitate the formative evaluation of large interfaces, as one is provided with quantitative and cognition-centric guidance regarding the states/features that should be redesigned. As the instructional design studies presented in Section 2.2 suggest, priority should be given to redesign of features with larger induction patterns (e.g. lane change alert, drowsiness alert, trip statistics). Depending on project resources, this redesign may involve current IU design frameworks, as well as numerous techniques devised in instructional design (e.g. illustrations, explanatory pop-ups, interactive tutorials).

5. PERSPECTIVES

5.1. Evaluation

A schema induction experiment can determine the extent to which states or features of an interface challenge the prior schemata of a user population. Such an assessment depends on the sample of participants, which must be carefully selected to meet the demographics and characteristics of expected end users. We selected young drivers who were familiar with common technologies, yet not with GPS systems and onboard computers. One expects that a different population would exhibit a different state scatterplot than the one presented in Fig. 5. For example, users naive with computers and smart phones would be expected to exhibit more inoperative induction patterns, and fewer positive induction and amendment patterns. Or, users familiar with onboard computers would be expected to exhibit more transfer patterns and fewer positive and inoperative induction patterns. The degree to which our screening method captures the prior schemata and learning needs of a user population of interest is an issue of sensitivity to examine in the future.

Methodological improvements are pending, such as the construct validity of each pattern. This work has already begun for patterns of transfer and positive induction, and is expected to extend to the patterns of inoperative induction, negative induction and negative amendment. For now, the latter are based on assumptions from literature that have yet to be addressed in a hypothetico-deductive fashion. Additional studies are needed to extend the scope of our method’s application, in particular regarding its sensitivity to individual factors, interface type, etc. Most important is verification of our method in terms of formative assessment, namely its ability to inform redesign and, if so, its cost-effectiveness compared to other lightweight techniques (e.g. cognitive walkthroughs, questionnaires).

5.2. Design cycle integration

The present paper discussed the risk of design techniques being ineffective—even counter-effective—for features that fit users’ prior schemata. Such risk may be attenuated by prototype testing before heavy deployment of IU design techniques, namely:

Stage 1: designing a prototype,
Stage 2: testing the prototype and determining which novel, innovative or challenging features should undergo Stage 3,
Stage 3: deploying IU design techniques (e.g. prior familiarity, habit inquiry), then iterating the process.

Stage I should incorporate well-established practices within the product domain (e.g. mobile vs. desktop), such as stakeholder inquiry, user-requirement and competitor analysis, information...
architecture design, interaction design patterns, usability principles, etc.

Stage 2 requires assessment techniques that are tailored to IU, familiarity or performance. To this effect, practitioners would conduct a schema manipulation experiment and pattern screening, such as the one presented here, or consider other approaches for IU assessment, such as coding of verbal protocols (Blackler, 2008; Blackler et al., 2010; Gudur et al., 2013; Lawry et al., 2010), self-rating questionnaires (Mohs et al., 2006; Naumann and Hurtienne, 2010) or performance metrics (e.g., click behaviors and task path, time, correct uses, Blackler, 2008; O’Brien et al., 2010). As discussed in Section 2.3, the choice of method carried out at Stage 2 should take into account the relative novelty of the functions being designed. Van Merrienboer (1997) and Camp et al. (2001) have argued that when introducing complex and innovative domain skills to novices, the most important performance criteria is accuracy. In this vein, design of innovative functions, or cutting-edge technology, should first be assessed in terms of accuracy. The reason why initial iterations of prototype testing should employ a schema induction experiment, rather than mere measurements of performance, is because causes of error cannot be objectified by performance alone (Fischer et al., 2014). To circumvent this issue, researchers in HCI often ask participants to introspect about their behavior (verbal protocol analysis, surveys; for a review of this trend in IU research, see Blackler et al., 2011). These approaches suffer the drawback of being prone to subjectivity from both the experimenter and participants. A screening method that is experimental and strictly quantitative provides more objective insight into user cognition. As a prototype’s flaws have been pinpointed and consistently corrected through redesign iterations, other approaches may suffice for monitoring and consolidating enhancements. If improvements are successful, speed and automation would become more sensible assessment criteria.

Stage 3 concerns existing IU frameworks, usability recommendations and more. Either the spiral tool or IBIS method, discussed in Section 2.1, would be deployed at this stage. A third and as yet untested framework has been formulated by O’Brien et al. (2010) and approaches IU design in a balanced and encompassing fashion. This framework recommends that designers consider which features of a device are best designed for analysis, and which are best designed for intuition. This viewpoint is one that we adhere to: technology, connectivity and automation inherent to smart phones, homes and cars require a realistic determination of which features can and should be intuitive, and which should be considered from a different angle such as learning or training. O’Brien et al. (2010) related this reflection to Hammond’s (1996) analysis-intuition continuum, and proposed to address it heuristically. For instance, practitioners can ask themselves questions (p. 98) such as ‘what kind of cues will individuals be examining?’, ‘Can a task be cleanly decomposed into discrete steps?’ and ‘Should users be aware of their cognitive activity?’.

We propose a means for simultaneously determining which features are intuitive (transfer pattern) and which require an analysis effort, with the latter ranging from supporting (viz. positive/negative induction patterns) to impeding learning (viz. inoperative induction pattern). As such, our method outlines a data-driven and cognition-centric alternative to heuristic approaches to the intuition-analysis continuum.

6. CONCLUSION

IU is commonly imputed to prior knowledge factors such as past experience, habits or familiarity. One straightforward application of this idea is that designers may render devices and interfaces more intuitive by making their features closer to what users have experienced in their past, making them more familiar, etc. Yet instructional design studies have raised a flag about this strategy. Notably, effective techniques oriented toward familiarity require neither that people possess relevant abstract schemata nor be too naïve about the domain. When these conditions are not met, the design techniques lose their effectiveness and, in certain cases, prove to be counter-effective (viz. expertise-reversal effect).

The cognitive and behavioral outcomes associated with different levels of prior knowledge thus draw a more complex picture than ‘the more familiar, the more intuitive’. In light of this issue, it is important to differentiate the features of an interface compatible with users’ prior schemata from those that are not. Only the latter, whose usage and assimilation are detrimental to IU, are worth redesigning. On the one hand, this paper illustrates how to pinpoint this compatibility in terms of schemata operations involved during first-time use of prototypes and at design levels that can be acted upon by practitioners (viz. features, states). Such screening may optimize prototype redesign by ensuring that IU design techniques are only deployed for the most deserving features. Future research is still needed, though, to fully establish the application space of the manipulation and screening of schemata operations.

On the other hand, this paper advocates a diligent incorporation of user tests into current IU frameworks, and IU design in general. Not only are user tests indispensable, they should be conducted prior to application of IU design techniques. Continuing in this vein, practitioners should scale assessment criteria and redesign expectations according to the relative innovativeness of their design projects. Arguably, performance speed and think-aloud protocols that Blackler et al. (2011) praised for researching IU seem sensible for devices that are common and do not pose major first-use difficulty. However, research has found that no improvements in performance (Blackler et al., 2014; Hurtienne et al., 2013) or even satisfaction (Hurtienne et al., 2013) ought to be expected from redesigning such devices for IU. Conversely, accuracy and schemata operations would better assess interfaces that contain many features.
and/or innovative technology, and are thus prone to first-usage errors. By rendering such features more intuitive, users would experience less failure and disorientation, and thus be in better position to appraise the new technology. Thus far, though, IU design techniques have been tested on devices having basic functionality (remote control, microwave oven, ticket-vending machine, etc.), meaning the goal of rendering innovative technology intuitive remains a challenge in this field.

FUNDING

The present research was supported by the G-COE Cybernics Program in the University of Tsukuba.

REFERENCES


