

UNIVERSITY OF LILLE

DOCTORAL THESIS

to obtain the degree of:

DOCTORATE OF THE UNIVERSITY OF LILLE

in the discipline

“COMPUTER SCIENCE”

by

Alice LOIZEAU

Understanding and designing around error in interactive systems

Thesis submitted on October 13th, 2025 and defended on December 9th, 2025 in
front of a jury composed of:

<i>Reviewers</i>	CARL GUTWIN	Professor, University of Saskatchewan
	CATHERINE LETONDAL	Professor, Ecole Nationale de l’Aviation Civile (ENAC)
<i>Examiners</i>	DUNCAN BRUMBY	Professor, University College London (UCL)
	ANNE ETIEN	Professor, University of Lille, <i>Jury president</i>
	PHILIPPE PALANQUE	Professor, University of Toulouse
<i>Co-Supervisor</i>	MATHIEU NANCEL	Chargé de recherche, Inria
<i>Thesis Director</i>	STÉPHANE HUOT	Directeur de recherche, Inria

INRIA CENTER AT THE UNIVERSITY OF LILLE

40 AV. HALLEY, 59650 VILLENEUVE-D’ASCQ
FRANCE

UNIVERSITÉ DE LILLE

THÈSE

pour obtenir le grade de:

DOCTORAT DE L'UNIVERSITÉ DE LILLE

dans la spécialité

“INFORMATIQUE”

par

Alice LOIZEAU

Comprendre et concevoir avec l'erreur dans les systèmes interactifs

Thèse soumise le 13 Octobre 2025 et soutenue le 9 Décembre 2025, devant le jury
composé de:

<i>Rapporteurs</i>	CARL GUTWIN	Professeur, University of Saskatchewan
	CATHERINE LETONDAL	Professeur, Ecole Nationale de l'Aviation Civile (ENAC)
<i>Examineurs</i>	DUNCAN BRUMBY	Professeur, University College London (UCL)
	ANNE ETIEN	Professeur, Université de Lille, <i>Présidente du jury</i>
	PHILIPPE PALANQUE	Professeur, University of Toulouse
<i>Co-Encadrant</i>	MATHIEU NANCEL	Chargé de recherche, Inria
<i>Directeur de thèse</i>	STÉPHANE HUOT	Directeur de recherche, Inria

CENTRE INRIA DE L'UNIVERSITÉ DE LILLE

40 AV. HALLEY, 59650 VILLENEUVE-D'ASCQ
FRANCE

Abstract

Computer Science considers that an error occurs when the system deviates from what it was built to do. The error is defined in reference to the device's specifications, the designer's envisioned use-cases. This approach establishes a hierarchy in the problems that users encounter: incidents that conflict with the designers' intentions are generally more likely to be resolved. Meanwhile, some user problems in Human-Computer Interaction (HCI) are studied not as errors, but as disturbances, annoyance- or frustration-inducing. These types of incidents fall into an intermediate space in the usual categorizations; they denote problems without being formally called errors.

This thesis explores the concept of error in HCI and proposes a new definition centered around user experience. With this change of perspective, the user becomes the reference by which the error is evaluated.

First, I provide an analysis of this concept in HCI and in the fields from which it has inherited. The word is studied semantically, first through an analysis of its synonyms in common language, then in its observed use in the domain of research. I identify and differentiate the various approaches to the concept in critical systems, safety sciences, and HCI, both as a research topic and research method. I also discuss the existing gap between what these models capture, and what users experience.

From this, I introduce a new definition of error as a *stray from user expectations*. This perspective allows us to focus on the experience of the person going through the error: their emotions, the repercussions they undergo, and their attribution of blame in the incident. This definition aims to encompass every phenomenon that presents a problem for users.

Negative experiences encountered by individuals on their interfaces are categorized and explored in depth via user studies. I conducted workshops with groups from different backgrounds, enriching a taxonomy of errors and prompting an analysis of the related discourse. We used these results to develop an online questionnaire, enabling the collection of information from a larger number of people, revealing meaningful trends based on the demographics or experience of the users, but also on the type of error.

Finally, in a controlled experiment, we explored the design space of solutions to the specific problem of interaction interferences. This phenomenon can only be characterized as an error from the angle of user-experience: it can occur even when every component of the interaction loop behaves as expected, and it presents an interesting combination of usually high frustration and low consequences.

These different projects highlight the relevance of a new approach to error from the user's perspective, opening discussions on the evaluation of incident criticality, the methods adopted in this evaluation, and the diversity and complexity of possible inconveniences on our interfaces. This work also offers recommendations and venues of future work to address these issues, by better taking user experience into

account in the study and handling of errors. This thesis defends user experience as the starting point for discriminating and fixing interaction errors.

Résumé

L'informatique considère qu'une erreur est présente lorsque le système sort de ce qu'il est supposé faire. L'erreur est définie par rapport aux spécifications de l'outil, aux cas d'usages imaginés dans son design. Cette approche hiérarchise les problèmes que peut rencontrer l'utilisateur-ice : les incidents entrant en conflit avec les ambitions de conception auront généralement davantage de chances d'être traités. En parallèle, certains problèmes utilisateurs en Interaction Humain-Machine (IHM) sont étudiés, non comme des erreurs, mais en tant que perturbations, gênes ou déclencheurs de frustrations. Ces types d'incidents investissent un espace intermédiaire dans les catégorisations habituelles ; ils sont problématiques sans être des erreurs nommées comme telles.

Cette thèse explore la notion d'erreur en IHM et en propose une nouvelle définition centrée sur l'expérience des utilisateurs et utilisatrices. Avec ce changement de perspective, l'utilisateur devient la référence à partir de laquelle l'erreur est évaluée.

Tout d'abord, j'apporte une analyse de ce concept en IHM et dans les domaines dont elle a hérité. Le mot est étudié sémantiquement, d'abord à travers l'analyse de ses synonymes dans la langue commune, ensuite dans son usage observé dans le champ de recherche. Je présente et distingue les différentes approches du concept dans les systèmes critiques, en ingénierie de la sécurité et en IHM, à la fois en tant que sujet et méthode de recherche. Je me penche également sur l'écart existant entre ce que ces modèles recouvrent et le vécu dans l'utilisation.

J'introduis en conséquence une nouvelle définition de l'erreur comme un *écart par rapport aux attentes de l'utilisateur-ice*. Cet angle d'étude permet de mettre la focale sur le vécu de la personne qui fait face à l'erreur : ses émotions, ses répercussions subies, son attribution de blâme dans l'incident. Cette définition se veut couvrir tous les phénomènes présentant un problème aux utilisateurs et utilisatrices.

Les événements négatifs vécus par les individus sur leurs interfaces sont catégorisés et approfondis par des études sur les utilisateur-ices. Des workshops sont conduits avec des groupes de différentes populations et permettent un enrichissement d'une taxonomie de l'erreur ainsi qu'une analyse du discours associé. Ces résultats sont ensuite mobilisés pour déployer un questionnaire en ligne, permettant de recueillir des informations sur un plus large nombre de personnes, révélant des tendances significatives basées sur les données démographiques ou l'expérience des utilisateur-ices, mais aussi sur le type d'erreur.

Enfin, dans le cadre d'une expérience contrôlée, nous explorons le designspace des solutions au problème spécifique des interférences d'interaction. Ce phénomène peut être caractérisé comme une erreur, du point de vue de l'expérience utilisateur seulement : il peut se produire même lorsque tous les composants de la boucle d'interaction se comportent comme prévu, et présente une combinaison intéressante de frustration généralement élevée et de faibles conséquences.

Ces différentes études soulignent la pertinence d'une nouvelle approche de l'erreur du point de vue de l'utilisateur-ice, ouvrant des discussions sur l'évaluation

de la criticité des incidents, les méthodes mobilisées dans cette évaluation, ainsi que sur la diversité et la complexité des désagréments possibles sur nos interfaces. Ce travail propose également des recommandations et des pistes pour les futurs travaux visant à résoudre ces problèmes, en tenant compte de l'expérience utilisateur dans l'étude et la gestion des erreurs. Cette thèse défend l'expérience utilisateur comme point de départ pour identifier et corriger les erreurs d'interaction.

Acknowledgements

This thesis represents the conclusion of four, if not five, years of working in Lille and learning HCI research. Even though I created this topic, what made it exciting and interesting, and what helped me progress as a researcher throughout my PhD, came down a lot to my supervision and support system.

I want to thank most of all Mathieu Nancel for his relentless support and advice, not only during my PhD, but also during the internship that preceded. Your kindness, quality of work, disponibility and humor have consistently made my work more interesting. It also made the process easier in many ways. I value very much as well our discussions about the nature of research and what it is to do good research work. They will stick with me as I go on.

I also thank of course Stéphane Huot for directing my thesis and for giving me the opportunity to work at Inria. By assuming more and more responsibilities as my thesis went on, so far as to directing the research center, you made my work possible in countless ways throughout my time at Inria.

I want to extend my gratitude to the jury as well for their work in reviewing this document and evaluating it, along with my defense.

I was lucky enough to be surrounded with the Loki team during this time. I want to thank all the *permanents* and *ephemerals* that were there with me, sharing the lively Start-R office. Through you, I learned a lot about the research profession and the field of HCI. In the future, if I meet students planning on a PhD, I will enthusiastically recommend you guys.

In particular, I want to thank Sylvain Malacria for working with Mathieu and me on the “shitty popups” article (here in chapter 3). I started this work as an intern and it accompanied me as I learned about HCI in general and research work. It is work that I am proud of, and I am proud and happy I did it with you two. Thank you also for the deep discussions about confidence relating one’s work.

I also thank Bruno Fruchard for the initiation and co-organization of “Les Innovantes”. While I am invested in the issues of gender equality and feminism, it was not necessarily easy to be the one raising them in a mostly masculine environment, even as considerate as Loki. The fact that you were the one bringing them to the table made me very comfortable and I was very happy when you invited me to organize a mediation event with you. “Les Innovantes” was a mediation day to promote computer sciences and HCI research to high-school girls. We held it on December 5th 2023 and December 6th 2024. It was cool to do, and made me feel like I made a little difference. I want to thank the communication department of the research center as well, and Magda in particular, for helping running this event.

I am also thankful for participating in the doctoral consortium of the IHM conference in Troyes, 2023. The jury was very helpful at this moment in my thesis, and I really enjoyed talking with the other student participants.

My friends and family supported me a lot throughout this time and I want to send them all my gratitude. Thanks Thomas for your presence and your patience. Merci Maman, Papa, Justine et Gabriel pour votre soutien dans ce roller-coaster émotionnel. Thank you Sophie and Anis for going through this around the same time. To Ophélie, Gab, Agathe, Aurore, Hugo, Claire, Louis. Thank you Louise, *Patte de*

Lou [133], for your fantastic drawings that make this thing a little more fun to read. To Jean-Luc and Agathe for the summer shelter. Bruno, Suliac, Quentin, Gaëlle and Sandra for the DnD nights. Les Intrépides and all the other circus folks for sharing this passion and giving me one of my favorite breathers. Thanks Les Rageux, bubbles of the Sphere in Paris, and the Torino gang.

```

X W S E D I P E R T N I T O M G O O O U F
Y H U M B P T D M H U G O B C Q X H E M P
H V Z A R K C A I E D V H N T P H V S I P
X J U T S L M E N A P A P C S H O F B D Z
C U F H A A J S T N E N A M R E P R M Z S
L S U I N N C E L L E A G Q Z X U I H U G
E T T E R M I A G A T H E S R N J E L T D
I I Z U I A A S Y X S O A R O R E I O O R
L N R G X D G L C T T N U U B Z A C M D L
E E H N G T J E E C D C N X T C N D E I T
H O P A I S H P U R G P L M C A - A Y F A
P R M W R T H S A X S V X A A W L J L W U
O W J I I A N U O E B L E D I N U H Y S R
K Z K L N W N E J P F X A P O R C C K G O
R O Z E P J I T U U H K E R B K E W A K R
L G R H C Y A H O Q M I C G E E B B V V E
R Z S M H J V O X X R A E M M M R H N H B
J T Z I C V L M Y B G R R M A I E E S O Z
Y L S M U E Y A S U L J B G E L E H G R P
T O R I N O S S A Q K R H L L H E V P O M
T O S M E O L U S E T N A V O N N I H E W

```

FIGURE 1: Word search: 30 words from the Acknowledgement section, all directions allowed

Contents

Abstract	3
Résumé	5
Acknowledgements	7
Introduction	13
1 Error as a concept in and around HCI	17
1.1 Definitions of error	17
1.1.1 A first semantic approach	17
What does “error” mean in everyday language?	17
How is the word “error” used in HCI literature?	19
1.1.2 Definitions around error in critical systems	21
Definitions in dependable computing according to Avizienis et al.’s taxonomy	21
Other definitions: human errors, mistakes and slips	25
1.2 Critical approaches of error in Safety Sciences	27
1.2.1 History of the concept	27
1.2.2 Overview of models and frameworks in Safety Sciences	29
1.2.3 Discussions among safety practitioners	32
The <i>Old View</i> : Overview of the ATS, the SRK, the GEMS and the PCM	32
Reason’s arguments for new safety practices	36
The <i>New View</i> : definition from Dekker and Hollnagel, the CO- COM and the FRAM	38
What opposes <i>Old</i> and <i>New</i> ?	43
The common ground	45
1.3 Sparse approaches of error in HCI	47
1.3.1 Existing taxonomies in HCI	47
Classifications of “human errors”	47
Global classifications	52
Classifications for specific contexts	54
Studies focussing on specific error types	58
1.3.2 The industrial heritage of HCI in user-experience evaluation	59
1.3.3 Error as a tool for research	61
1.4 Arguments for a different definition of error	63
1.4.1 Missing link with individual experience: existing studies	64
Frustrating events	64
Preference of control over accuracy	66
Errors can create interesting user experiences	67
System interpretation errors	67
1.4.2 The edge cases of errors	68

	Slips	68
	Interruptions	70
	Interaction interferences	71
2	The lived experience of error cost	75
2.1	Categorization	75
2.1.1	Categories extracted from the literature about error character- ization	76
	FAIL-UNSAFE: a special kind of ERROR IN PLANNING	78
2.1.2	“Bordeline cases” categories	78
2.2	Study 1: Open-ended user consultation on the taxonomy	80
2.2.1	Workshops methodology	80
	Groups	80
	Structure	80
	Data and material for analysis	85
2.2.2	Results on the categorization	87
	Feedback on the categories	87
	Comparing the workshops through <i>Incidents_W</i>	91
	Changes to the categorization	95
2.2.3	Thematic Analysis	96
	Coding	96
	Global observations on codes	102
	Comparing the categories through codes	103
	Comparing the workshops populations through codes	109
	Codes and themes	110
2.2.4	Discussion	112
	Implications for the categorization	112
	Observations of users facing errors	115
2.3	Study 2: User polling on precise error situations	116
2.3.1	Online survey	116
	Structure	117
	Demographic questions	121
	Questions about each incident	122
	Questions about feelings and emotions	125
	Participants’ recruitment	127
2.3.2	Results	127
	Participants	127
	Data Pre-processing and Analysis Approach	128
	General correlations with demographics	129
	General observations	129
	Categories results	131
2.3.3	Discussing the survey	143
	Global observations on errors	143
	Remarks on the CATEGORIES	144
	Comparing workshops’ and survey’s results	145
3	Mitigating interaction interferences and their consequences: an applied study	147
3.1	Background: Perception and impact of interaction interferences	147
3.1.1	Characterization of Interaction Interferences	147
3.1.2	Causes of Interaction Interferences	147

3.1.3	The Singularity of Interferences as Errors	148
3.1.4	Preventing and Correcting Interferences	149
3.2	Solutions to Interferences	150
3.2.1	A Design Space of Solutions	150
	Types of behavior	150
	Interpretative requirements	151
	Technical requirements	152
3.2.2	Naive Behaviors Design for Pointing-triggered GUIs	153
3.3	Controlled Study	157
3.3.1	Participants	157
3.3.2	Apparatus	157
3.3.3	Tasks	158
	Primary task	158
	Secondary task	158
3.3.4	Procedure	160
	Calibration phase	160
	Task practice phase	161
	BEHAVIORS phase	162
3.3.5	Design	163
3.4	Results	164
3.4.1	Data Pre-processing and Analysis Approach	164
3.4.2	Triggering Interferences	164
	From user interactions	165
	From secondary task answers	165
	From subjective feedback	167
3.4.3	Addressing Interferences Overall	167
	From secondary task results	167
	From secondary task answers	167
	From subjective feedback	169
3.4.4	Side-effects of BEHAVIORS	169
	From user inputs	169
	From subjective feedback	170
3.5	Discussion	173
3.5.1	Overall Utility of <i>Naive</i> BEHAVIORS	173
3.5.2	Feasibility and Technical Requirements	173
3.5.3	Applicability to Other Types of Interferences	174
	DISPLACE	175
	POSTPONE	175
	BLOCK	175
	MAPPING	176
	<i>Correct</i>	176
3.5.4	Overriding the Behaviors	176
3.6	Summary and Conclusion	176
4	The stray from user expectations and its revealing possibilities	179
4.1	Redefining errors as user expectation violations	179
4.1.1	What does a user-centered concept of error entail?	179
4.1.2	Errors and bugs as negotiated denominations	181
4.1.3	The setting of wrong expectations	182
4.2	Methodological considerations around errors	183
4.2.1	Building on classical questionnaires	184

4.2.2	Quantitative and qualitative	185
4.3	The stakes of interaction put forth by errors	186
4.3.1	The benefits of a distance with representation models	186
4.3.2	Errors and art	188
4.3.3	Distancing a Dark Pattern	189
4.3.4	The discretion of interaction interferences	190
Conclusion		191
Glossary		193
Acronyms		195
List of Figures		199
List of Tables		202
A Definitions of synonyms of error		203
A.1	English definitions	203
A.2	French definitions and their traduction	205
B Workshop Reports		209
B.1	Workshop 1 (HCI researchers)	209
B.2	Workshop 2 (baseline)	210
B.3	Workshop 3 (IT professionals)	211
B.4	Workshop 4 (baseline)	212
Bibliography		213

Introduction

When my smartphone freezes, my code does not compile, or my computer is slow, I get frustrated. It catches me off guard, I had not planned on spending time solving a problem at that moment, it bothers me. Yet behind this apparent uniformity of “the error is annoying”, there often is something else at play.

Sometimes, I am worried I will not be able to accomplish what I set out to do. What if I miss my train because I cannot show my ticket on my smartphone? What if I cannot figure out this code? What if my device cannot run this app? Other times, I get annoyed with my tools. They should be faster, clearer, they should be transparent and not get in the way of what I am doing. Sometimes I find people to blame. Who decided to make my tool obsolete so quickly? Who placed an advertisement where I was going to click? At times, my smartphone runs out of battery when I do not need it, but I am still nervous about not having it at hand. I lose a feeling of safety that I grew used to, and generally take for granted. And, more often than I would like to admit, I am confronted with my own limitations. It was me who did not know, and I should have. I should have known that this update could erase some of my files. Since I learned how to code, I should understand this error and be able to fix it. I should have planned ahead and stored my photos in multiple locations, because then I would still have them when my smartphone was stolen.

When something “wrong” happens, my reaction is not only about the event itself. The device, the situation, the task matter. There is also a tendency to see the system as a deterministic machine that is only doing what it was made for. On the other hand, the human that I am is irrational and can fault. “To err is human”, as the saying goes. That saying also exists in French, but reads slightly different. We say “The error is human”¹. In the English phrase, the act of making a mistake is something all humans do, when in the French one the error is human by nature. The language hints about how human fallibility is reflected in errors.

The field of HCI studies the experience of users with digital tools, and how their design can influence it. What about errors? When conducting controlled experiments in the domain, an error from the participant is an information. We put controllers [84], pointers, screens [125], environments [157] in the hand of the user to see what works and what does not. When HCI focuses on error itself, it inherits from studies being set in critical environments. Of course, error is most worthy of investigation when a lot is at stakes: in an airplane cockpit or in a nuclear power plant. This focus on safety in extreme environments amplifies a need for predictability and control. Machines are handled well in this context, but people not so much. They are valued for adapting, making compromises and handling unknown situations. They are not predictable. As a consequence, the study of human errors is integral to the study of error. Error in HCI borrows from human information processing studies, from safety engineering and overall from Safety Sciences. The discussion revolves

¹“L’erreur est humaine”

around the same things: where the problem comes from, how to prevent it and how to fix it.

In this thesis, I choose to shift the focus perspective and investigate the user experience of errors. What is at stakes for the user when an error happens? I start by defining an error through the user's eyes:

an event, caused by or during an interaction with a digital device, that does not meet the user's intents, expectations and wants about that interaction.

With the focus on user experience, the range of what qualifies as an error is enlarged. A negative experience caused by the interface can include other phenomena that would not otherwise be considered errors. The shift in focus allows for a better understanding of what is problematic, and how to solve it.

Among the phenomena that qualify as errors with this new definition, interaction interferences stand out [162]. They are a phenomenon due to a change in the interface as the user was about to perform an input, happening too late for the user to stop their course. They are caused by a human physiological threshold and by the system limitations in considering it. They can feel human-induced, which is probably why people do not flag them as errors to be fixed, and possibly the reason why the phenomenon has gone under the radar up until recently. However, when people are asked about it directly, they notice it and report frustration [162]. Users seemingly feel that they have to adapt to the system, up to their reaction time. That is, until they are pointed out that the system could behave differently. Interaction interferences are a matter of negative experience, more than they are about material cost (in data, hardware or time) [162]. They would not be a problem of priority for Safety Sciences, if all we cared about was predictability and efficiency. They are a problem worth solving if we care about the user's emotions. How do we design with the user experience in mind? How can phenomena like interaction interferences be mitigated on the system's part?

This thesis investigates errors in everyday use of interactive systems, and how the understanding of user experience can help shaping the design of their interactions.

I start by presenting what an error is, linguistically, in its academic roots in critical systems and in HCI. I then create a categorization of errors based on user experience, that I confront to users in two ways. First, I report on workshops that I conducted with different populations to understand their take on the categorization and to note the points that organically come out when talking about errors. From here, I poll users about specific errors in order to grasp what the stakes are to them when they happen. I then explore the design space of solutions to the specific problem of interaction interferences, a phenomenon which illustrates well the flaws in current approaches: it can occur even when every component of the interaction loop behaves as expected, it does not usually induce grave consequences but can cause negative emotions. I design and test solutions in a controlled experiment, a study that was published at ToCHI in 2025 [110]. Finally, the discussion revolves around the implications of the new proposed definitions in terms of error evaluation and reflexions on the opportunities brought forth by these incidents.

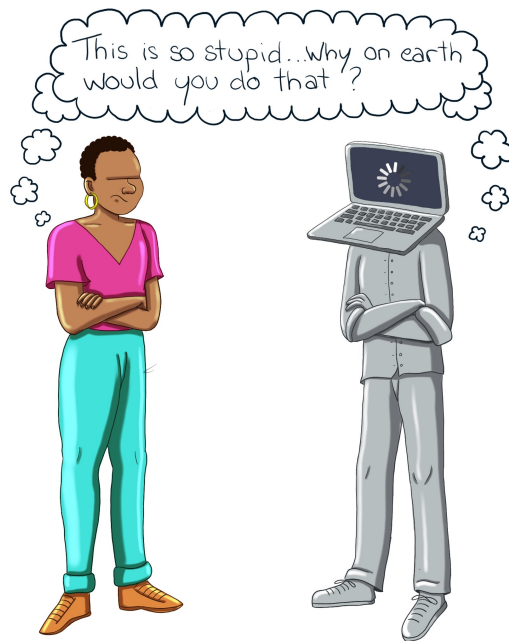


FIGURE 2: Expectation clash drawn by *Patte de Lou* [133]

Chapter 1

Error as a concept in and around HCI

1.1 Definitions of error

1.1.1 A first semantic approach

What does “error” mean in everyday language?

Before delving into the meaning of “error” in the HCI literature, I set out to grasp its general meaning. When people say “error” day-to-day, what do they mean? What is implied? In order to find out, I carried out a semantic analysis based on the observation of synonyms. I elected to conduct this study for French and English synonyms, to have two comparable references, and for they are the languages that I am most comfortable in. They also happen to have a very similar word for the concept, with the same latin root: “error” and “erreur”. In French and English, the words associated with the idea or concept of error provide information on its semantic implications, *i.e.* the different meanings associated with it. These implications may vary from one language to the other. However, I did not incorporate more languages in the frame, because the point is not to have an exhaustive linguistic analysis. In order to compare what an error implies in HCI specifically, we want here a starting point for understanding what an error is in general.

I gathered the synonyms from two leading websites in the area. *Thesaurus* [178] for English, and *Synonymo.fr* [175] for French. To get a wide view of the meaning of the word, we also needed sources that gave numerous synonyms, which was the case for both of them. By the end of 2021, *Thesaurus* gave 49 synonyms for “error” and *Synonymo* gave 53 synonyms for “erreur”. Since the number of synonyms was equivalent in the 2 languages, I elected to keep all of them in the analysis.

With the lists of synonyms established in both languages, I identified different codes that linked some of them together. I made them emerge from the similarities between synonyms and the recurring ideas they convey. In this simple Thematic Analysis, I followed a bottom-up approach. For instance, in English, “sin” and “offense” both have the underlying concept of intent, so they are tagged by the same code. I also allowed words to be in several categories at once.

The process of tagging was helped by the definitions of these synonyms, in both languages. These definitions can be found in Appendix A. I define the different themes stemming from this analysis in Figure 1.3.

We can then observe that each code highlights a different dimension of error. The most common code is *[STR]ay*, which applies to words involving deviation to what

- | | | |
|---|---|---|
| <ul style="list-style-type: none"> • blunder [SKI] • failure [STR] • fault [SKI] [FLA] • flaw [FLA] • glitch [FLA] • inaccuracy [FLA] • lapse [STR] • miscalculation [STR] • miscue [SKI] • misdeed [STR] • mismanagement [STR] • misstep [SKI] • misunderstanding [INT] • offense [STR] • omission [STR] [CON] • sin [STR] | <ul style="list-style-type: none"> • transgression [STR] • wrongdoing [STR] • X [CON] • absurdity [SKI] • boner [CON] • boo-boo [CON] [SKI] • delinquency [STR] • delusion [INT] • deviation [STR] • erratum [CON] • fall [CON] • fallacy [INT] • falsehood [DEC] • falsity [STR] [DEC] • goof [SKI] • howler [SKI] | <ul style="list-style-type: none"> • misapprehension [INT] • misconception [INT] • miss [CON] • oversight [SKI] • screamer [DEC] • slight [CON] • slip [SKI] [CON] • solecism [STR] • stumble [SKI] • trespass [STR] • untruth [STR] • bad job [STR] [SKI] • faux pas [STR] • misbelief [INT] • misjudgment [INT] • screwup [STR] [SKI] • slipup [SKI] |
|---|---|---|

TABLE 1.1: Most common synonyms in English according to The-saurus [178]

- | | | |
|--|--|---|
| <ul style="list-style-type: none"> • aberration [INT] • écart [STR] • égarement [STR] [SKI] • ânerie [SKI] • étourderie [SKI] • aveuglement [SKI] • bêtise [SKI] • bévue [SKI] • bavure [CON] • blague [SKI] [DEC] • boulette [SKI] • bourde [SKI] [DEC] • confusion [INT] [CON] • contresens [INT] • coquille [CON] • défaut [STR] [FLA] • dérèglement [STR] | <ul style="list-style-type: none"> • déviation [STR] • distraction [SKI] • doublon [CON] • duperie [DEC] • entorse [STR] • erratum [CON] • errements [STR] • faiblesse [SKI] [FLA] • faute [STR] • faux [STR] [DEC] • faux pas [STR] [SKI] • folie [SKI] • fourvoiement [STR] • gaffe [STR] • hérésie [STR] • illusion [INT] [DEC] • impair [STR] • inadvertance [SKI] | <ul style="list-style-type: none"> • inexactitude [FLA] • infidélité [STR] [DEC] • injustice [STR] • irrégularité [STR] [FLA] • mécompte [INT] • mégarde [SKI] • maladresse [STR] [SKI] • malentendu [INT] • manquement [STR] • méprise [INT] • mensonge [DEC] • non-sens [STR] [FLA] • paralogisme [SKI] • plantage [STR] • préjugé [INT] • quiproquo [INT] • sophisme [DEC] • tromperie [DEC] |
|--|--|---|

TABLE 1.2: Most common synonyms in French according to Synonymo.fr [175]

[STR] STRay from normalcy, reason, moral or rule
[SKI] action from someone lacking intelligence, attention or SKill
[INT] error in the INTerpretation of a situation or a person
[DEC] voluntary DECEPTION
[CON] material or emotional CONsequence from a bad action
[FLA] the process or the object may fill its purpose in some way, but has a FLAW

TABLE 1.3: Themes of error coming from its synonyms

is expected. One deviates from the norm (“lapse” / “déviation”), does not follow a rule (“transgression” or “faute”) or goes against morality (“offense” or “impair”).

In fact, the word comes from the latin *error* both in English and in French, meaning “*straying from the path*” [186].

Secondly, an error can be caused by a person lacking the necessary abilities or expertise for the proper execution of a process. These words bear the [SKI]ll code. A human may lack intelligence or judgment (“absurdity” or “bêtise”), attention (“oversight”/“distraction”) or skill (“slip” or “maladresse”). They may also simply fail by accident (“slipup” or “mégarde”). These incidents are thus not necessarily intentional.

An error can also be a situation in which communication fails to take place, or more precisely, where the interpretation of an object, situation or person is distorted. In English, we find “misunderstanding” or “delusion”. In French, we have “contresens” or “malentendu”. These words are tagged with [INT]erpretation.

An error can also be induced, caused deliberately. Examples include “falsity” in English, and “duperie” or “mensonge” in French. In the words tagged with [DEC]eption, the falsehood is intentional, not accidental.

Some words not only refer to an erroneous event at a given point in time, but also to the consequences of it happening. For example, “fall” in English or “confusion” in French can refer to an incident (when a fall occurs, when a confusion is made) but also to the state following that incident (where the fall is the consequence of an error, where the confusion is a result of miscommunication). [CON]sequence tags words that can refer to the consequences of an error.

Finally, some synonyms of the word refer to a shortcoming, a dysfunctional element. They imply that the expected result or object fulfills some of its functions, but not all. Here, [FLA]w emphasizes the element of incompleteness in error, with words like “flaw” and “inaccuracy” in English, or “défaut” and “faiblesse” in French.

Of course, the precise tagging in the tables 1.1 and 1.2 is debatable; words oftentimes have several meanings and uses. In fact, some words in this analysis bear several codes at the same time, making a representation such as Figure 1.1 possible. Synonym lists have the same limitation: they are not definitive or absolute. This complexity comes from the very nature of language, but also from the fact that English and French are spoken in different places around the world. As such, vocabulary and its use can branch. A word can have different meanings or implications in different places. This analysis is thus rudimentary, it does not claim to hold an absolute truth about the meaning of words. Instead, the themes that are put forward by this analysis are interesting in that they inform the concepts held by error.

In the general language, the word error encapsulates indeed different ideas. First, an error is an event where a person or an object strays from expectation. It can sometimes be caused by the faillibility of a person, or an intention on their part. An error can also point to the consequences of a faulty action or process, be it in emotional, material or performance costs. Everyday use of vocabulary shaped these aspects of error. However, “error” is also a very common word and concept when working with technology. Interaction with a computer and research in HCI have specific constraints that can shape the use of vocabulary. It is possible we find different themes when looking at how HCI speaks about error. How is the use of the word of “error” shaped by the research field and its practices?

How is the word “error” used in HCI literature?

“Error” is not only used in articles *about* error in the HCI literature. The word and concept is actually found in a large proportion of articles in the field, even if the

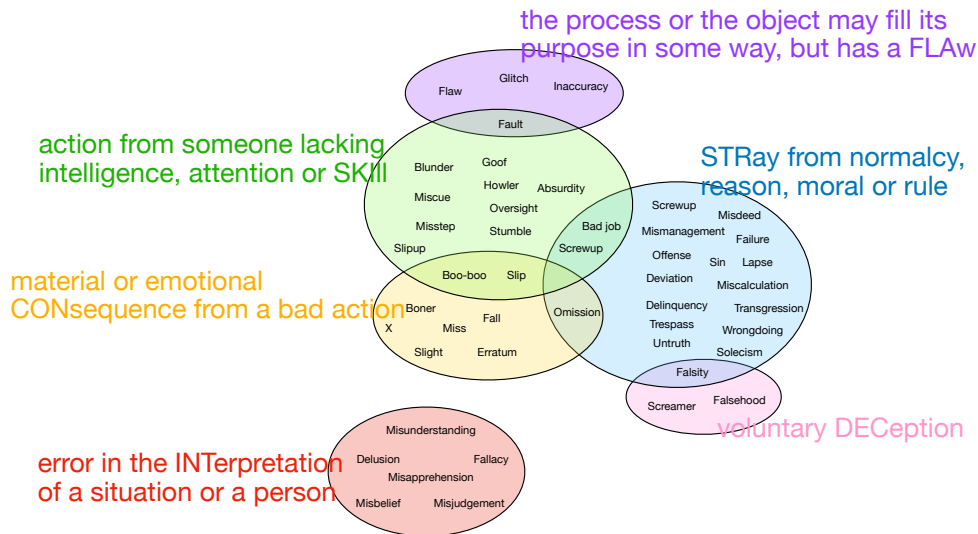


FIGURE 1.1: Venn diagram of the English synonyms codes

study is not about error specifically. I will now report on how the HCI literature uses the concept, and mirror it to the synonym analysis above.

When we search in the *ACM Library* [1] for articles with the word “error” in their title, we get 7,501 results. This number jumps to 45,542 if we look for that word in abstracts. A systematic literature review would be possible, but not easy and straight-forward. This is left for future work and is not part of this thesis. However, it is still possible to draw my impressions from my state of the art and from conversations with other researchers.

In this section, I focus on articles that do not study error as a concept, but rather the ones that reference it. The concept is also an object of study in HCI, as we will later see in section 1.3. Here though, the aim is to observe how the word is used when it is not the center of the study, *i.e.* what are the shared values and assumptions that surround the word when it is commonly used in the field.

It is indeed possible to compare the words associated with error in everyday language and in the field of Human-Computer Interaction. A number of themes can easily be found in both. Error seems in both cases to represent a move away from a desired state, placed in an opposition between correct action and erroneous action. The definition of error as a problem of communication or interpretation is very present in HCI, particularly in approaches where error is not associated with user blame [123, 132].

Looking back at our list of synonyms, we notice some new themes emerging. In HCI and in computer science in general, error is quite often associated with human error. In everyday language, too, it seems that error is sometimes directly linked to human action. Certain words cannot easily be associated with the malfunctioning of a machine or process (e.g. “goof” and “screwup” in English, “folie” and “mal-adresse” in French). It also appeared that there could be confusion about the temporality of the error. For most of the approaches observed, the error itself is situated in an erroneous action of the user. However, it can also be located further upstream, in the process that leads to this action (an erroneous cognitive schema) or in its direct

consequences (the system ends up in an undesired state). In our vocabulary, too, we can observe this kind of ambivalence. In French, is an “inadvertance” the act of forgetting something, or the action that follows? In the same way, “misunderstanding” seems to refer to the moment of the communication gone wrong, and the ensuing different views of what was communicated.

Finally, it appears that certain ideas do not appear in everyday language, but are essential to understand the concept of error in human-machine interaction and ergonomics. For example, human error in HCI is often linked to notions of control, controllability, precision, reliability and commands, particularly with regard to errors linked to input techniques [28, 54, 105, 166]. As a result, it is associated with a sense of agency, or the feeling of being at the origin of and in control of one’s action. It is also associated with attributed or felt responsibility and blame [132, 140]. It is finally impossible to approach error in our field without addressing the notion of performance. Error is frequently expressed in terms of the “error rate” in controlled experiments, which is often an important indicator of the performance and reliability of an interaction tool or technique ([28], [169] among many examples). Performance is a sort of antonym to error in the literature, mostly when it is accompanied by a lack of speed.

The concept of error may seem obvious in everyday language, and therefore in HCI as well. It is a notion that we sometimes think needs no clarification or definition. Yet error as a word and error as an object or tool of study carry different implications. In what follows, we will therefore present the different approaches to error found in the literature.

1.1.2 Definitions around error in critical systems

The concept of error is rarely defined as such in HCI literature. As previously stated, it is a common word that generally calls for no definition, although it encompasses various situations and phenomena. The scientific definitions of error that HCI inherited therefore come from other fields: fields where it is necessary to reflect on error, because its occurrences are critical.

Definitions in dependable computing according to Avizienis et al.’s taxonomy

Avizienis et al. in 2004 presented in a taxonomy the concepts of secure computing [9]. Even though the focus of the paper is safety and security, this work serves as the foundation for larger or more field-specific definitions of error, as [131] in HCI. Indeed, Avizienis et al. define the user as the receiver of a service, whereas the provider would be a system. The user in this framework need not be a human; it could be another system. While this taxonomy is the base of subsequent conceptualizations around error, it is specifically system oriented. The interaction in which the notions are defined do not have to involve a person at all.

The work defines the concept of dependable computing, which is the core of the taxonomy. In fact, the author provides us with four different definitions of *dependability*.

According to the International Organization for Standardization (ISO) in 1992 [49], dependability is “*The collective term used to describe the availability performance and its influencing factors: reliability performance, maintainability performance and*

maintenance support performance". For ISO, the main element grounding dependability is availability. According to Avizienis et al. [9], this traces back to the telephony standard, where availability was most important.

Another standard [139] defines dependability as "*The extent to which the system can be relied upon to perform exclusively and correctly the system task(s) under defined operational and environmental conditions over a defined period of time, or at a given instant of time*". This definition brings forth the capacity of the system to be relied upon. Here, the notion of availability is not even mentioned.

Two other definitions are brought by Avizienis et al. [9]. The first one generalizes over the usual ("*more classical*") understanding of the notion: dependability is "*the ability to deliver service that can justifiably be trusted*". The paper proposes on top of that an alternate definition based on the argument that a system can be dependable even in the face of failures. Being dependable then becomes "*the ability of a system to avoid service failures that are more frequent or more severe than is acceptable*". According to Avizienis et al. [9], dependability is not the complete absence of failures, but the limitation of their happening and their severity.

How dependability is defined is essential, because on it relies the notion of dependence and trust. Indeed, Avizienis et al. define the *dependence* between two systems as "*the extent to which [one system]'s dependability is (or would be) affected by that of [the other]*". In other words, a system A is dependent on a system B if B's dependability has an impact on A's. From there, *trust* is "*accepted dependence*" between two systems.

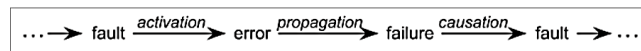


FIGURE 1.2: The chain of dependability according to [9]

The article also sheds a light on the authors's conceptual difference between a failure, an error, and a fault. When the service provided deviates from a *correct* state, it is referred to as a *failure* (or a service failure). Indeed, a correct service is "*when the service implements the system function*". So, a failure is a non-compliance to specifications. The way the external system strays is called an *error*. A *fault* is the assumed cause of an error. Not all faults however make it to the external state of the system and result in failures. When they do, they are active, otherwise they are dormant.

Taking a step back, the study not only defines the notions of fault, error and failure, but presents them as a chain of causations. As one can see in Figure 1.2, a fault causes an error that causes a failure. A failure, can in turn cause a fault. The main difference between the three concepts does not seem to be rooted in some "who came first?" chicken and egg question. Rather, the capacity to be perceived seems to draw the line between a fault and an error: an error is a fault made visible. If the fault is dormant, it means that it has not been noticed yet. In turn, an error and a failure differ in their criticality. If an error causes the system to not be able to deliver an acceptable service, it becomes a failure.

To attain dependability – as defined by Avizienis, four kinds of strategies exist to avoid faults or their consequences. *Fault prevention* is meant to stop faults from occurring or from being introduced. When a fault is active, *fault tolerance* aims to prevent service failures. Then, the goal of *fault elimination* is to lessen the frequency and intensity of defects. Estimating the current number, future incidence, and anticipated sequences of faults is *fault forecasting*.

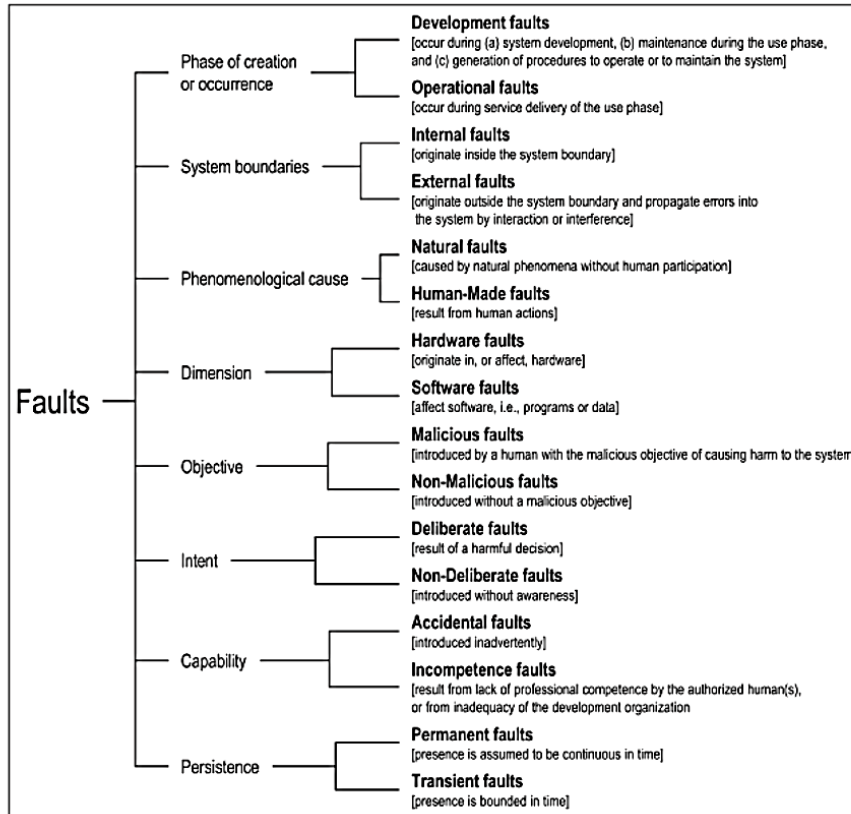


FIGURE 1.3: Fault classes as presented by Avizienis et al. [9]

The different fault classes are described in Figure 1.3. The study defines a fault as the cause of an error, but interestingly, a hardware fault “*originate[s] in, or affect[s], hardware*”. In that case, a fault seems to be also characterized according to the consequences it has, the error it becomes. In the case of hardware, it is indeed difficult to draw the line between fault and error. For Avizienis et al., an error is the manner in which the systems fails, *i.e.* the phenomenon. In the case of hardware, something broken can be seen as the cause and the phenomenon – if my battery dies, what is the failure, what is the way it fails, what is the cause?

The classes in Figure 1.4 are not mutually exclusive, and their combinations fall into three groupings that overlap each other: Development faults, Physical faults and Interaction faults. Development faults include all combinaisons that involve the eponymous class (see Figure 1.3), *i.e.* faults that occur during system development, design or maintenance. Similarly, Physical faults are all the faults that are not software faults. An Interaction fault is an external fault, *i.e.* a fault that originates within the system boundaries. Physical faults is the only grouping that overlap the other two. A Physical fault can indeed sometimes also be a Development fault or an Interaction fault, but not the three at once.

Avizienis et al. [9] also define specification faults, which originate in the specifications not mirroring correctly the system functions. From Figure 1.3, we can conjecture that a specification fault is a kind of development fault, occurring at the very first steps of conception.

Avizienis et al. [9] then bring forth a classification of failures. As we see in Figure 1.5, service failure can be seen from four viewpoints: the service content and timing (domain), whether they are signaled to the user (detectability), whether they

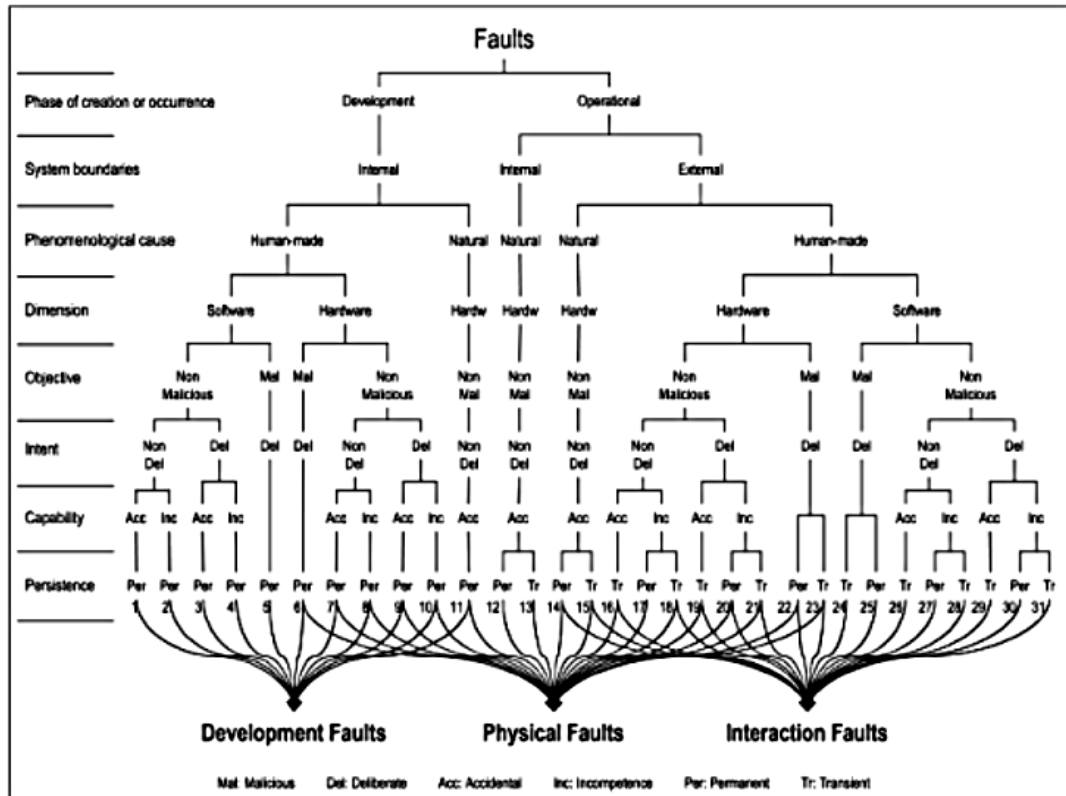


FIGURE 1.4: Fault classes tree as presented by Avizienis et al. [9]

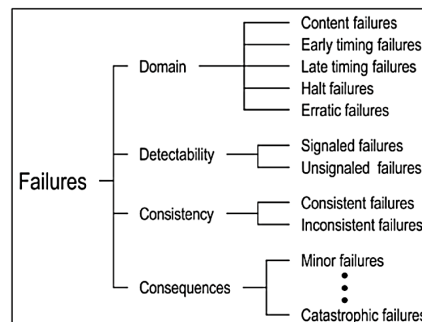


FIGURE 1.5: Failure modes according to the four viewpoints as presented by Avizienis et al. [9]

appear the same to all users (consistency) and by the grade of their consequences. A minor failure is not catastrophic: its cost does not exceed the benefits that a correct service would provide. With that, Avizienis et al. define a *fail-safe system*: “A system whose failures are, to an acceptable extent, all minor ones”.

I will lastly present here the main point of Avizienis et al. [9] on errors. An error can lead to a service failure, if it reaches a component that is part of the external state of the system. An error can be detected, or be latent if not. It can cause a failure depending on the structure of the system (is it fault tolerant?) or its behavior (is the error in a part of the system that is needed for the service?). Avizienis et al. classify errors according to the service failures that they cause, presented in Figure 1.5.

Avizienis et al. [9]’s article is an interesting starting point when it comes to characterizing errors. The definitions proposed however have a strong relationship to

the security and dependability domains. An error here is the symptom of a fault that can propagate to cause a failure *i.e.* a state of the system where it does not fulfill its function(s). An error is seen there from the conceptor's point of view, where the function of a system is well defined from the beginning. An error is straying for the system's function, a definition reminiscent of its etymology (see section 1.1.1). However, when we are not working with critical systems, straying can be interesting, or even desirable [39]. Exploring outside the defined function(s) of a system can lead to other user strategies to accomplish the same task, or even different use cases that the system can help for.

This does not mean that Avizienis' definitions do not work in the field of HCI, but rather that the *system's function* is difficult to pin down in that context. What is, today, the function of a computer, of a tablet, of a smartphone? If those tools used to have a restricted set of functions (calculus, data management, communication etc.), it would be difficult to list everything people use them for now [36, 91]. It would be then very restrictive to only define a system's function depending on its developers and conceptors' idea. What of the users' expectations?

Other definitions: human errors, mistakes and slips

Depending on the theory around error, words will not have the same definition. The definition given to "error" will impact the meaning of close concepts. For example, in Avizienis et al. [9]'s approach, the notion of error is directly related to the understanding of fault and failure. "Slips" or "mistakes" are not a part of this model, and thus not defined formally.

Definition of "human error"	Source
<i>"an unsuccessful experiment with unacceptable consequences"</i>	Rasmussen [142]
NA fault: <i>"incorrect selection of action to reach a goal, or incorrect execution of that action"</i>	Hollnagel [78]
<i>"a generic term to encompass all those occasions in which a planned sequence of mental or physical activities fails to achieve its intended outcome, and when these failures cannot be attributed to the intervention of some chance agency"</i>	Reason [149]

TABLE 1.4: Definitions of "error"

In other approaches, presented in the table 1.4, "error" is often synonym of "human error". This analogy is indeed relatively common in the literature on errors.

For Rasmussen [142], an error starts with an experiment. To him, human experiments are necessary to learn and develop skills, and an error is when one of those experiments has consequences that cannot be reversed. When those effects can be reversed, the definition of error depends on *"a reference or norm in terms of the successful outcome of the activity"*. Rasmussen accepts human error as a concept, but underlines the fact that the outcome matters more than the behavior in the judgement of human actions: *"if a short cut is successful, the person is clever, if not, it is an error and he is blamed"* [141].

Hollnagel, on the other hand, refuses to give a definition to “error” or “human error”. In a position paper for North Atlantic Treaty Organization (NATO) Conference on Human Error, Hollnagel [78] presents a taxonomy of human error specifically and defines its key terms. According to him, the word “error” is misleading, and one should instead talk about “action” or “activity”. His argument is that we could understand human behavior better by considering that the same processes produce both correct and erroneous actions. The refusal to give a definition of human error is caused by a change in focus of the theory itself, that this chapter will present later on. Based on this, his definition of “fault” separates the planification and the execution part of human action. A fault does not replace a human error in his vocabulary, to him, it is a “*non-observable construct*” [146]. His definition of fault can however be taken as a reference to compare his theory of error with that of other sources.

For Reason [149], a human error starts with a “*planned sequence of mental or physical activities*”. Similarly to the two other authors, an error is dependent of the outcome of these activities. As Hollnagel as well, planification and execution are two possibilities for things to go wrong.

Definition of “mistake”	Source
“ <i>incorrect selection of goal state; incorrect goal decision</i> ”	Hollnagel [78]
“ <i>an error in the intention</i> ”	Norman [130]
“ <i>[Errors] in the adequacy of the plan</i> ”	Reason [149]

TABLE 1.5: Definitions of “mistake”

Definitions of “mistake” for Hollnagel [78] and Reason [149] cover one type of error in their approach: a mistake is an error of planification (see table 1.5). If Norman does not give a definition of error or human error in this particular article [130], he shares the definitions of “mistake” presented here.

Definition of “slip”	Source
“ <i>the error that occurs when a person does an action that is not intended</i> ”	Norman [128]
“ <i>an error in carrying out the intention</i> ”	Norman [130]
“ <i>unintentional substitution of a correct performance segment (action) with an incorrect one</i> ”	Hollnagel [78]
“ <i>discrepancy between the intended actions and the ones that were actually executed</i> ” (also for lapse)	Reason [149]

TABLE 1.6: Definitions of “slip”

Similarly, the definition of “slips” is pretty similar among those three authors, as can be seen in table 1.6. With “mistake”, it is two sides of the same coin: when mistakes are committed in the goal formulation or the planning of the action, a slip happens at the execution of that action.

Reason [149]’s approach also includes lapses and violations as technical vocabulary, clarified in Figure 1.6.

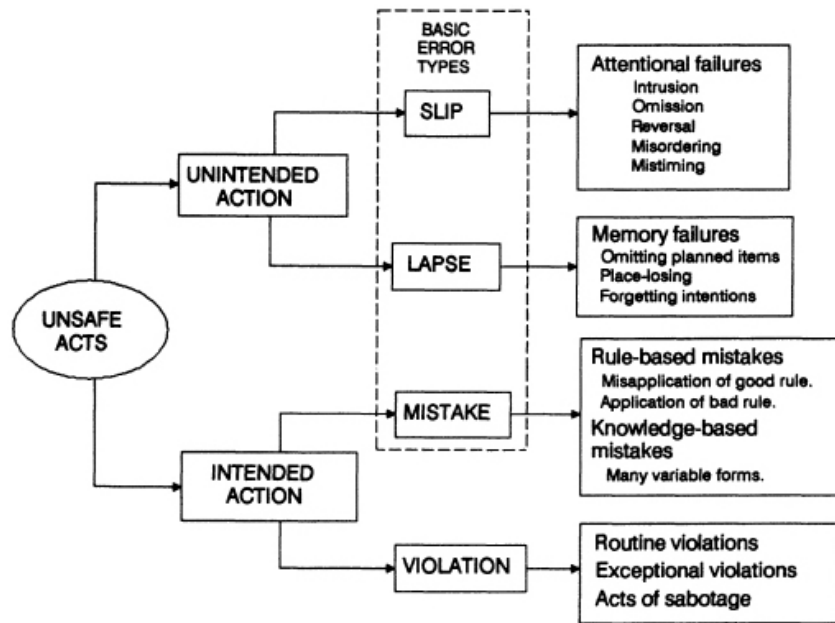


FIGURE 1.6: Taxonomy of unsafe acts in Reason [149]

At first glance, the concept of error seems to be relatively shared, given the similarity of the definitions given by different authors. The rest of this chapter explains how its treatment differs from one theory to the next; how tools, models and methods made from an idea of what an error is goes to show its different conceptualizations. This treatment of error depends indeed on epistemological positions. Does error intervene as a disruptive element, or is it part of the normal functioning of things? Is it contextual or systemic? The answers to these questions determine the models, and therefore the methods, used to deal with error.

1.2 Critical approaches of error in Safety Sciences

1.2.1 History of the concept

The concept of error has evolved in time, be in Safety Sciences or in HCI. For the sake of this thesis, we will look at its evolution since World War II. The work of Read et al. [146] is very precious to this end, as it describes how the notion of human error has moved through this time.

Technology is known to thrive during war time, when all efforts and means are focused on a conflict. It is also a time where practicality wins over theoretical efficiency. Things need to be usable, and fast. In the context of World War II, the US Army Air Force Aero Medical Lab hired a psychologist for the first time to investigate the causes for their recurrent crash landings. Chapanis studied the Boeing B-17 and realized that the crashes were due to a design similarity between the commands for the flaps and landing gear, and that pilots could confuse one for the other in demanding times. By changing the shape of the switches to better discriminate them, the rate of B-17 crashes went down.

Later on, Fitts and Jones [59] generalize Chapanis' work by showing that most pilot errors were actually stemming from design errors. They provide recommendations to avoid said errors, stating that *"the incidence of human errors can be reduced substantially*

by designing controls and cockpits in relation to human requirements”.

This work initiated interest around human error as a field of study. Human error was not just a practical nuisance anymore, that sometimes prevented an otherwise well-designed tool to function. These studies around cockpits introduced the idea of adapting design to human psychology, by studying how humans act with that tool.

As the interest for human error takes off, the 60s get interested in how humans make decisions, how they take action and the overall process that takes place in their mind as they do. The first information processing-based approaches to human error are published during this time. Payne and Altman [134] for example, provide “a procedure for the early, quantitative evaluation of electronic equipment operability”. In this work, they study human behavior in order to adapt how electronic equipment presents information and how it can be controlled.

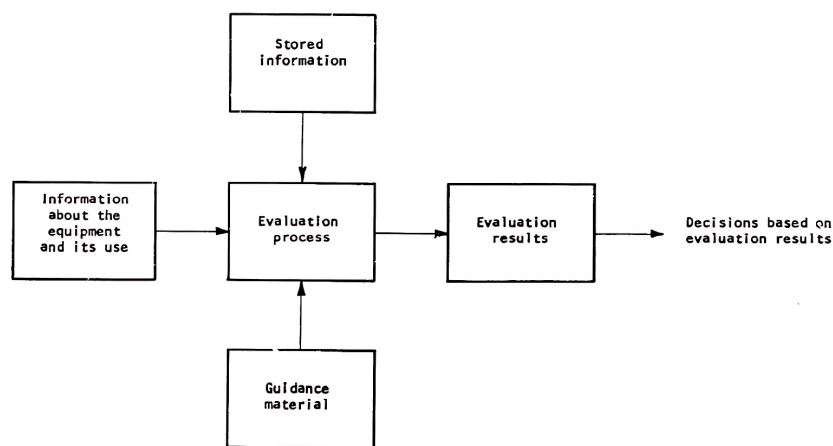


FIGURE 1.7: The user operability evaluation simplified schematic in Payne and Altman [134]

If Figure 1.7 is a simplified version of the schematic developed in the paper, it however conveys the lens through which the human operator is seen. They are an information processor; making decisions based on information stored, and on information available to them. This work is a good example of the 60s’ point of view on the human operator. They are part of the system and are studied as such, evaluated through the *inputs* they require and the *output* they provide. More precisely, it does not consider all human errors the same. *Input errors* are related to perception, *mediation errors* stem from the information processing and *output errors* are wrong physical responses [134].

The context in which errors are made by humans is considered more and more in this period. For Simon, human decisions are not optimized for rationality, but people rather act as “*satisficiers*” [171]; they aim for an acceptable threshold rather than optimizing or maximizing any performance aspect. Their strategy is dependent on the context, its criticality, and on their goal, they will not necessarily use the most reliable course of action at all times. Simon [170] argues that human decisions are local or bounded, *i.e.* that the rationality of a decision is not general but dependent on the operator’s particular position, knowledge and constraints.

This leads, in the 80s, to the point of view that human error is not an error per se. More specifically, that speaking of human errors is not helpful or relevant in lowering the overall frequency of system failures. Human error, according to Hollnagel

[78], is not an observable occurrence. It is instead the attributed cause of a discrepancy between the intended and actual states of a system. That mismatch is the thing that is observable. Hollnagel questions the relevance of using human error as an explanation for a system's poor performance. In fact, Rasmussen [142] observes that this explanation is usually employed when no alternative is available, i.e., when the phenomena appears to be outside the scope of the system's technological components. Hollnagel's position paper does not support the need for a theory of human error, but rather a need for a theory of human action. He argues that human errors are produced by the same mechanisms that produce correct actions. However, he still wants to differentiate between internal and external causes of errors for the operator. Because to him, those are often intermingled: *"Stress, for instance, is often caused by (situationally) unreasonable demands to the operator. And deficiencies in the design of the interface may often be compensated by the adaptability of the operator"*. To explain an error, a theory of human action would need to be better at encompassing those internal and external causes, because it would account for them to explain human behavior.

For Hollnagel, the environment quality plays a bigger role than the operator quality to ensure the reliability of a system. Because of this, human performance put in number is not that valuable a tool. In fact, *"quantitative measures, such as error rates, will [...] be inadequate and even misleading"*. Instead, we should lean into qualitative studies, that could then inform better more relevant quantitative measures.

Hollnagel takes part of an overall impulse to study errors within the system and not in isolation of it; to defend that the concept of human error is conceptually poor, because the human is part of the system.

These discussions around human errors and their origins leads the Safety Sciences to engage with the notion of reliability differently. In 2004 is held the first symposium on Resilience Engineering (RE) [43]. RE is a subfield of Safety Sciences, studying ways in which a complex system can react to the unexpected. I will now examine the frameworks that those domains have brought forth, and the discussions that they motivate.

1.2.2 Overview of models and frameworks in Safety Sciences

Errors and their consequences are most often studied in fields where they are critical; i.e. where they can have grave consequences. Typically, Safety Sciences focus on scenarios like operating a power plant or piloting an aircraft. Several frameworks and methods to analyze human error, and error in general, co-exist in the field. These models of error are established first and foremost to reduce the number of errors, but the way in which they do so can carry big epistemological differences.

In 1998, Kirwan [88] lists and evaluates different frameworks for Human Error Identification (HEI). Thirty-five approaches are considered, including influential ones like Skill, Rule and Knowledge [144] (SRK) by Rasmussen [144] and Generic Error Modeling System [149] (GEMS) by Reason [149]). Kirwan first reports the criteria by which he assesses those theories. The evaluation process is interesting to examine, because it says a lot about what is relevant for HEI, i.e. what is at stake around human error.

According to [88], a good HEI method can be recognized by its:

- Comprehensiveness: the number of different human behavior that the technique can encompass

- Consistency: strictly structured vs. open-ended results
- Theoretical validity:
 - based on a model of human behavior
 - assessing External Error Modes (what happened) but also Psychological Error Mechanisms (pattern recognition) and/or Performance Shaping Factors (situational factors)
- Usefulness: can reduce the number of errors
- Resources:
 - required time to apply the technique
 - required training to use the technique
 - required expertise to use the technique
- Documentability: easy or difficult to document
- Acceptability:
 - has been used and assessed
 - available or discontinued
 - earliest applicable life cycle stage
- Versatility: Task versus error-based HEI
- Recovery: can model error recovery
- Probabilistic Safety Assessment [127] (PSA) representation: format or method in which the results inform the PSA
- Output quantifiability: does the technique produce quantitative data for Human Reliability Assessment (HRA)
- Validation evidence
- Special features
- Maturity: in terms of age and usage
- PSA practicability: can PSA use this technique and its output

In other words, a good HEI method encompasses various human behaviors, produces quantitative data useful for PSA, is structured and consistent, validated by theory, efficient in terms of resources, time, and activatable results. It is also best that the method is documented, validated, and easily documentable. One can observe that the evaluation of the techniques is made with the goal to capitalize on them. Here, roughly speaking, the criteria call for a systematic method that significantly reduces the number of errors with as few resources as possible. We could argue that context-specific, qualitative or long-term approaches may be overlooked or underestimated with these specifications.

Interestingly, Kirwan notes that none of the method reviewed ticks all the evaluation boxes. Instead, most of the models are “*not highly structured*” and not “*fully prescriptive*”. A lot of the methodologies to assess human errors view the subject from an academic point of view, optimizing for understanding rather than capitalizing.

The author goes so far as to see that this is “*implying that HEI is still an art rather than a well-definable science*” [88]. This judgement here is coherent with the intent behind the evaluation criteria. If what they value is the productivity and efficiency of models before anything else, models that are build for exploration and understanding are not going to be highly regarded. However, the fact that so many models chose qualitative, adaptive approaches may be telling of the challenges that HEI faces. Is human error so context- and situation-specific that it cannot be captured by more formal modelizations?

Kirwan sees some trends in Human-Error Identification methods. Firstly, most of them seem to revolve around Error Of Commission (EOC) and cognitive errors. As opposed to errors of omission, EOC are errors made in action. The second observed trend is a preference for models, instead of error-based taxonomies for example. In Safety Sciences at the time, according to him, the focus seems to be on understanding how human act altogether rather than having a list of errors that they can make.

However, Kirwan presents a list of the error types that an ideal new system should cover. For the practicability, or PSA, of a HEI method, it should address “1. Skill and Rule-Based Error Forms (slips and lapses, and rule-following errors) 2. Cognitive errors (diagnosis and decision-making errors) 3. Errors of Commission 4. Rule Violations 5. Teamwork and Communication Errors”. This list can be seen as a rudimentary taxonomy for human errors, errors that should be identified by a good framework, model or method in order to lower their occurrence.

Other models attempt to predict the probability of human error occurrence in order to avoid them. Leadbetter et al. [104] in 2001 develops a model for assigning probabilities to human errors in the context of Air-Traffic Control (ATC). The paper provides a model of the control flow in the HCI (see the state machine in Figure 1.8).

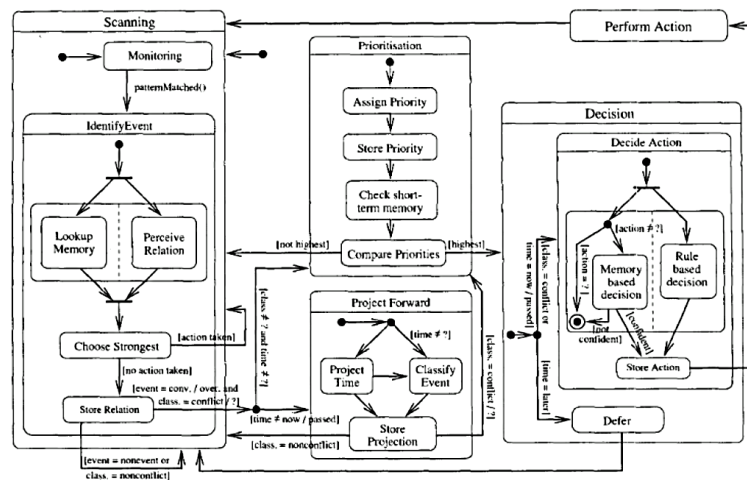


FIGURE 1.8: “Control flow model of the ATC simulation operator”

Leadbetter and al. consider three kinds of human errors: errors of perception (“Monitoring” in Figure 1.8), lack of performance in the action (“Perform Action”) and errors in the generation of actions (cognitive error, happening at any other spot in the flow). The paper suggests that a probability should be assigned to every cognitive error. The goal is to predict the error-rate of every step of the cognitive process,

as represented in the model. Contrary to the majority of the approaches in Kirwan [88], this paper gives way to quantitative methods based on the model presented. In the same spirit, Gertman et al. [61] present a HRA method to estimate the probability for human errors in response to events in a nuclear power plant. The paper uses the Standardized Plant Analysis Risk-Human Reliability Analysis [61] (SPAR-H) method, designed specifically for that context. They tested its efficacy at predicting error rates for different tasks against other HRA methods, with promising results for SPAR-H. In this approach, humans perform tasks that relate either to an action or a diagnosis. The goal is to support risk analysis based on event probabilities.

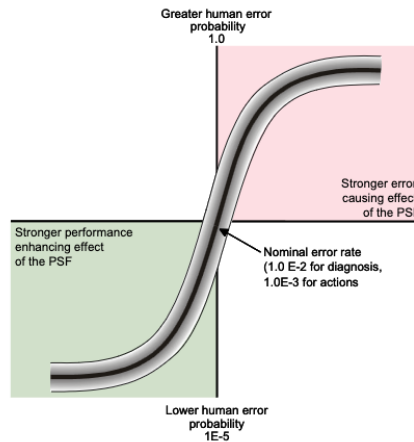


FIGURE 1.9: Influence of Performance Shaping Factors on human error probability [61]

In both Leadbetter et al. [104] and Gertman et al. [61] works, human error is considered in a critical system, be it aviation or nuclear power facilities. In both cases, the response to those contexts was to apply a quantitative approach to predict and evaluate human error. Indeed, when the costs for potential errors are high, one wants to make sure that the possibilities for an operator to disrupt the system are as limited as possible. These environments are already very constrained in terms of what actions are possible for the human at a given moment. It is a response to a very constrained environment where errors can become highly critical.

1.2.3 Discussions among safety practitioners

Among the diverse methods, frameworks, models and taxonomies to identify and characterize human error, two views stand out and oppose each other, fueling recent debates. A taxonomy by Read et al. [146] and a paper by Le Coze [103] present these perspectives and the debates that surround them.

The *Old View*: Overview of the ATS, the SRK, the GEMS and the PCM

The first view stems from work on human information management models up to around the 1980s. It is sometimes called *Old View* in the modern literature, a name it gained a posteriori from *New View* supporters criticizing it. It is associated with models that break down the steps involved in human interpretation of information, that are then precursors to models that aim to pinpoint where and when an error occurs in this process.

Norman [128] with the Activation Trigger Schema [128] (ATS) for example, establishes a theory of action based on cognitive schema. In this model, a person takes

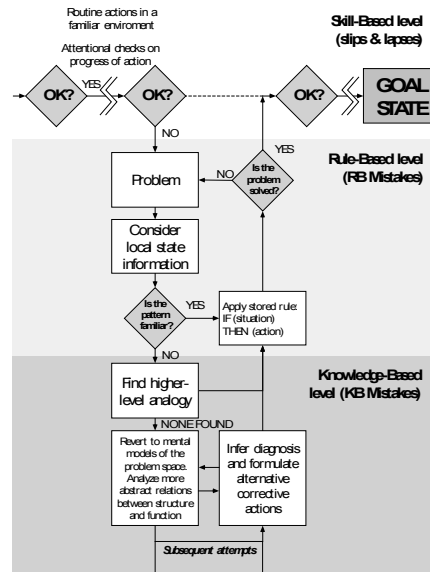


FIGURE 1.11: GEMS as presented by Embrey [51]

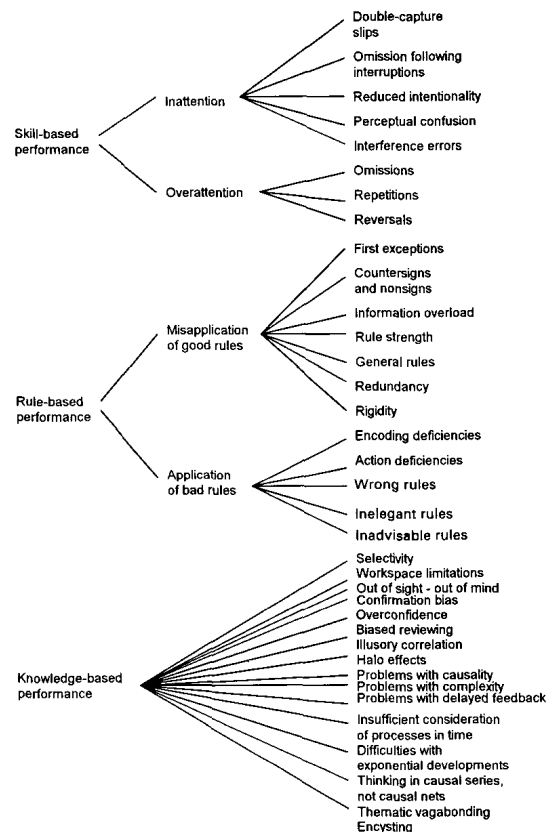


FIGURE 1.12: Categories of errors from the GEMS presented by Hollnagel [80]

One last core example of an *Old View* model is Neisser [127]'s Perceptual Cycle Model [127] (PCM). Similarly to Norman [128], the model is based on Schema Theory, and explains how schemas are triggered by the environment. However, in this approach, they can be updated by the person owning them. People applying schemas to the environment can observe their consequences and adapt them on the

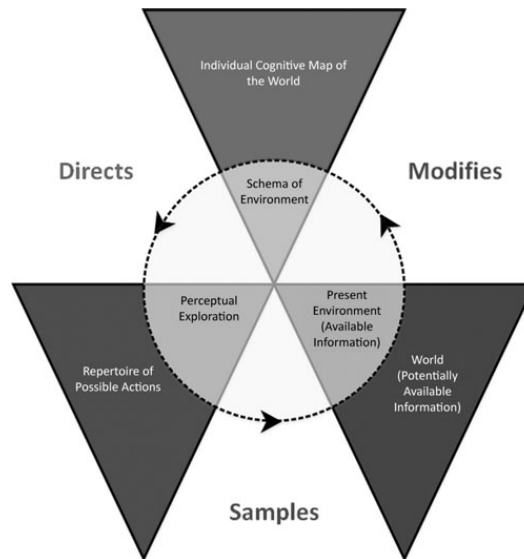


FIGURE 1.13: Neisser's Perceptual Model Cycle as presented in [138]

go. This update can in turn change how the person act on the environment. As one can see in Figure 1.13, there is a cyclical dynamic where the schema directs the perceptual exploration, that samples the present environment which then modifies the schema.

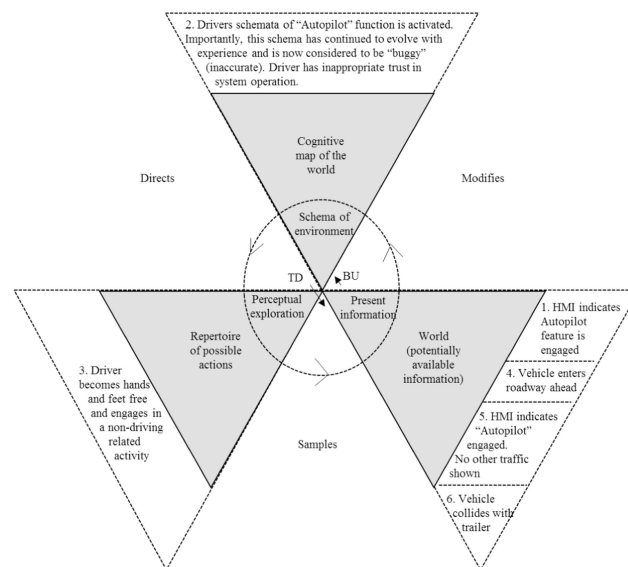


FIGURE 1.14: PCM adapted and applied to the Tesla crash incident by Banks, Plant, and Stanton [16]

The PCM has been used to analyze human errors in critical systems in the last decade. Human errors happening in semi-autonomous vehicles for example, can lead to grave consequences, as in the Tesla crash on May 7th 2016 [16]. The driver of a Tesla in Autopilot mode was killed when their car struck a tractor trailer that was crossing a highway intersection close to Williston in Florida. In their paper, Banks, Plant, and Stanton [16] investigate the causes of the accident using the PCM (see Figure 1.14). Their analysis suggests that an error of design in the Autopilot mode may have been causal to the incident in that it could have misled the driver's schema

on the system's capabilities and autonomy. Another study from Revell et al. [156] also uses the PCM in order to underline the downfalls of autonomous vehicles' interfaces. One of their conclusions is also the recommendation to design for the desired mental model that the user should have of the system.

The *Old View* got its name and characterization a posteriori. The frameworks presented here have in common that they modelize human thought process in order to better understand where precisely a human error takes place. Applied for error management in organizations, they become what Hollnagel later calls *Safety-I* thinking [146].

Reason's arguments for new safety practices

According to Reason [152], *Safety-I* in companies is characterized by the belief "*that safety is best achieved through a predetermined consistency of their processes and behavior*". From this perspective, uncertainty is best reduced to a minimum. Hence variability of performance is not desirable either. Among other reasons, this prompted critiques toward the *Old View* which motivated the need for other approaches. Reason is a key figure in this process. Though he is not always considered a *New View* advocate, he formulated a lot of the criticism that was raised against error management practice at the time.

The Swiss Cheese Model (SCM) marks a milestone in Reason's work. According to [100], that is where he moved from a psychological view of accidents, characteristic of the so-called *Old View*, to a systemic approach, as defended in the *New View*. This explains why Reason is an author representative of and cited in the two *Views*; he is not referenced for the same work for each one [146].

In the SCM Reason first brings to the table the notion that bad luck is often involved in mishaps, accidents and errors in complex systems. However, it is often overlooked for other possible causes. Reason illustrates how hazards can get through walls of safeguards in organizations by the SCM (see Figure 1.15). First published in 2000 [150], this model became very popular in safety practices.

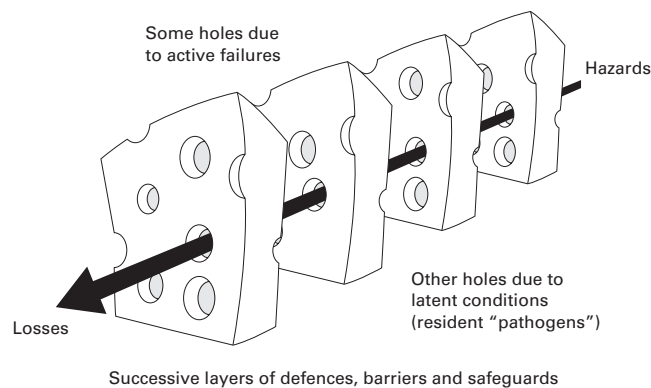


FIGURE 1.15: SCM to explain the cause of accidents from Reason, Carthey, and Leval [154]

Larouzee and Le Coze [100] explain that the model distinguishes between latent and active errors. Latent errors can be compared to pathogens, and active errors to actual illness. Reason makes the case that the more latent errors there are in a system, the more likely the system will have active errors. Also, big systems can

encompass more latent errors (or pathogens). Finally, latent errors can be detected, a process that is easier than tracing what led to an active error. In the Figure 1.15, the cheese layers are protection barriers. The holes are latent and active errors. In a confluence of circumstances, these errors, loopholes and/or unsafe acts, can align and lead to an accident. It is worth noting that there is no need for an active error, an actual unsafe act, to happen for an accident to occur in this model. The congruence, the “alignment” of those safety “holes” might be enough. Reason published several variations of his Organizational Accident Model (OAM) over the years (*i.e.* [147] or [153]).

Reason, Carthey, and Leval [154] later develop the Vulnerable System Syndrome [154] (VSS), presenting the human cognitive biases that leads an organization to not solving its actual problems. The VSS has three symptoms, that are reinforcing each other: a strong tendency to blame individuals, a denial in the face of systemic errors or problems, and an excessive reliance on quantitative indicators for productivity.

The blame bias finds roots in human psychology, in 4 tendencies.

1. The Attribution Error: we are more prone to explain a wrongdoing by a lack of performance by the person (they were inattentive, reckless etc.) than by the context leading to their decision.
2. The Illusion of Free Will: it is important for people to feel like their actions have consequences on the world, that they have the freedom to decide what their impact will be, when often they interact with situations too complex to ensure this level of determinism.
3. The Just World Hypothesis: the belief that good things happen to good people, and bad things to bad people, leading to conclude that bad things must be caused by bad persons.
4. The Hindsight Bias: the impression that past events were predictable, more so than they were without the perspective of looking back on it.

Those 4 biases are encouraged on an organizational level. It is easy to point at operators when looking for causes of errors, and convenient from an administrative point of view, preventing further investigation and alleviating responsibility. One consequence of this blaming pattern (that Reason calls “*blame cycle*” [151]) is that the cause of the error is not actually studied: “*the institution fails to learn that errors and non-compliances mark the starting point of an investigation, not its conclusion*” [154]. The blaming is made on the least manageable part of the process as well, as very little can be done to fix the fallibility of people. It is in fact “*far easier to fix situations than to change people*” [154]. It is also interesting to note that, according to Shaver [167]’s study, the more serious an incident is, the more blame will be attributed to the human.

The second symptom of a VSS is denial that the organization can be the cause of an accident. This is not necessarily intentional, often safety management just happens locally, limiting the possibility of solving the problem in-depth. In situations where managers are evaluated on the cost-efficiency of the processes they put in place, “good” management can often mean “easy and not expensive”.

This relates to the final bias: the measurement of performance. How do we know that safety goals have been reached? If we focus on efficiency and cost-effectiveness,

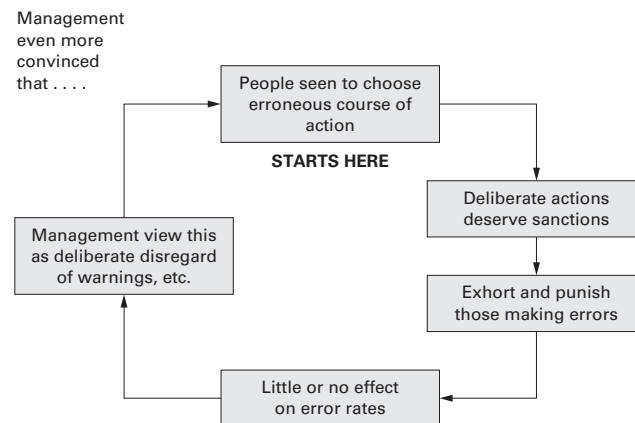


FIGURE 1.16: Blame cycle from Reason, Carthey, and Leval [154]

the data that the organization is going to assess its safety quality on is going to reflect those values. However, safety does not always agree with these constraints, and doing better might mean doing less with more: *“Managing by such objectives is what professional managers have been trained for and, not unreasonably, they feel that their performance will be judged primarily by the extent to which they achieve these goals. This opens the way to the single minded pursuit of the wrong kind of excellence.”* [154].

Through his work, Reason takes the first step in the direction of *Old View* criticism. In fact the theories or the models do not seem to be the problem, but he questions the relevance of their application in an organization for safety management.

The New View: definition from Dekker and Hollnagel, the COCOM and the FRAM

The second view I will describe has come to be known the *New View*. It is “new” in that its raison d’être derives largely from a critique and construction on the established standpoint at the time. In particular, the GEMS and the SRK raised a number of issues when applied for safety in organizations.

The concepts of *Old* and *New View* were coined by Dekker [44], proposing a transition from the one to the other. Hollnagel [79] is arguing for a similar transition, from what he dubs a *Safety-I* to a *Safety-II* approach. Dekker, Hollnagel, and the other representants of the *New View* have notably presented models challenging established norms. They challenge the fact that error is frequently attributed to the operator at the end of the line, without taking into account the context in which it occurs. Indeed, as argued in [146], it is easy to see frontline work as more procedural and normed, while management task are less precisely defined. Even though, despite its apparent regularity, this kind of work still requires adaptation and negotiation with rules and standards. This means that, when attributing the cause of error on an individual, it is easier to do so on the end of the line. These authors therefore advocate for an approach in which *error is an integral part of the overall process, rather than a disruptive element*.

The *New View* coincides with the 80’s turn in the definition of “human error”. In that period, more and more researchers support the idea that the notion of human error does not address the right problem, sometimes so far as to say that the human error does not exist. For Dekker, like Reason before him, the “human error” is not a

satisfying diagnosis on an accident analysis, it is rather the beginning of an investigation about the system itself: “Rather than saying what people should have done to not suffer the mishap, the new view tries to understand why what people did made sense to them at the time” [45]. Dekker however acknowledges that an adoption of the *New View* in an organization is complex. To come out of the “*blame cycle*” [151], organizations have to change their typical blame response to an accident. Dekker, Siegenthaler, and Laursen [45] observed four organizations over the span of two years that shared in the beginning a difficulty dealing with human error. The companies (an airline, an air navigation service provider, a national airforce and a regional hospital) all faced a critical environment regarding error were all “*trying to learn how to learn from failure*” [45]. The authors present 6 stages that those companies go through (or are looking to go through) in this endeavor. Stage 1 is realizing that the *Old View* does not work. The paper observes four organizations face a sort of paralysis, finding that their reactions to human errors did not seem to lower their occurrence. Reprimanding operators, implementing stricter procedures or adding automation could not solve everything. This ideally leads to Stage 2: the realization that the problem lies not with human error, but with the organization itself. This recognition is uncomfortable, because an organizational problem calls for work and time to be solved. “*It is not about placing blame*”: here the challenge is to not look for human errors elsewhere, like higher up in the hierarchy, which would bring them back to Stage 1. The label “human error” as a cause for failure informs more about the process of labelling, the risk prevention strategy in place, than it explains the failure. Stage 3 is stopping to apply the usual fixes to human errors. For a while, these reactive Safety practices are frozen. For example, when an accident happens, the process in question is not immediately updated or the perceived responsible person immediately sanctioned. Instead, the companies waited to see what would happen in the absence of those reactive measures. In the two organizations that implemented this, they observed no negative impact on the number of incidents, and sometimes even a positive one. This freeze paves the way to Stage 4 insight: that organizations are not safe by nature, and made unsafe by people. Safety is created by people and their practice, the way they manage trade-offs, constraints and conflicting goals. The authors observed a change here in language as well. To explain accidents, people under experimentation referred to contexts, systems and relationships instead of blaming operators. Stage 5 is the implementation of *New View* solutions to the problem at hand. Some organizations for instance created tools to manage conflicting goals, by presenting the possible trade-offs and their consequences, to help people manage them better. Finally, the final more meta 6th Stage is achieved when the organization tries and sees the big picture on how they learn from accidents. What are their strategy when handling errors? What can be learned from this strategy and how can it be improved? The authors point out that this 6-stage process is a difficult one, especially where the organization finds itself in a line of work, a country or a community that has an interventionist culture towards failure. Dekker, Siegenthaler, and Laursen [45] present here a timeline of an organization (a system, a company) assimilating the *New View*.

Hollnagel was one of the other main figures backing the argument that the human error was a question put in a wrong frame (see section 1.2.1). With Woods, he suggested that quantifying errors could support the idea that people are unreliable by nature and that they are to be managed as such or replaced with automation [82]. In 1991 [81], he develops his idea of *erroneous actions*. For him, there are two ways to look at them, depending on the philosophy of HCI behind.

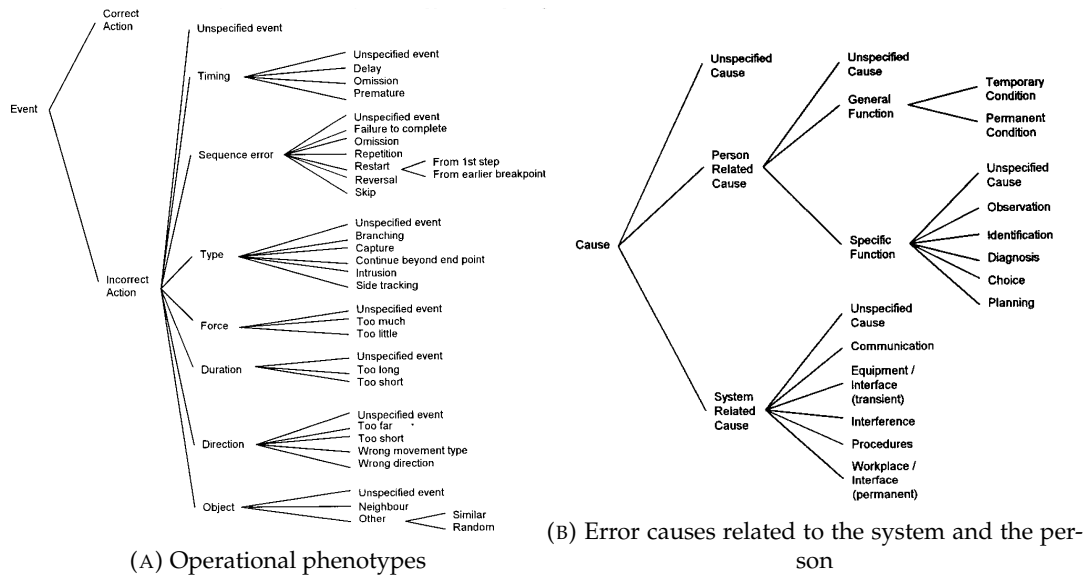


FIGURE 1.17: Phenotype and genotype of errors in Hollnagel [80]

The first standpoint, the fragmented standpoint, comes from an interpretative view of HCI. The user does not interact with the world through the machine, but with the world as portrayed in the machine. Also, they are seen as just one component out of the system's many. In this standpoint, erroneous actions are a direct consequence of the interface imperfections, and can be reduced by giving the machine more control in the most error-prone situations. This stance does not call for a detailed theory of errors, since they are solved by simply putting the human out of the equation when necessary.

The holistic standpoint comes from an amplifying view of the interface. The system, human included, is seen as a unit that cannot be taken apart. The user has a relationship with the world through the machine, that frames their possible feelings and actions. Here, erroneous actions come from an unforgiving environment, and are somewhat predictable in terms of what will go wrong and how often. The solution is to rely on the human more, and on their strengths. Hollnagel embraces that standpoint for his analysis, because it is the one that will benefit from a theory of human errors.

Hollnagel differentiates the *phenotype* and the *genotype* of an erroneous action. The first is its empirical experience, the way it can be observed. The second is related to its root causes, its functional characteristics. Hollnagel's classification [80] covers phenotypes and genotypes of errors (see Figure 1.17). The argument to distinguish genotype and phenotype is that they are easily and often times confused: "*Psychology is unique among the empirical sciences in being able to confuse manifestation and cause*" [80]. In other words, one should avoid to confuse observation and interpretation. The same article also puts forward the element of timing in the wrongdoing of an action (see "Timing" in Figure 1.17a).

Observing that phenotypes allow for a description of performance based on observation, Hollnagel later challenges information processing models, pointing out their depiction of a cognitive process isolated from the world and its triggers [53]. For that, his theory takes roots in the Joint Cognitive System [82] (JCS) he developed alongside David Woods [82]. They described humans and technology as linked in a system capable of doing cognitive tasks, and not as two entities opposing each

other. People working together with and through technology perform cognitive tasks in a JCS. For Hollnagel (and in the JCS), cognition is not computation, and should not be reduced to it. The author describes how to create a Contextual Control Model [76] (COCOM), a modelization of the interaction taking place between a system, an environment, and the user's actions and perceptions. A COCOM is a functional approach to describe performance, and because it is drawn from observation, it takes the context of the action into account. It is constructed on 3 concepts: *competence*, *control* and *constructs*. The competence is the set of all possible actions that a person can apply to a situation depending on the context and resources. This competence is brought into play in a certain control mode, that could vary between scrambled, opportunistic, tactical and strategic. The first mode refers to a situation where someone has no control and chooses their next action at random. The more one understands the context in depth, the more they can move up in control modes, up to strategic. Lastly, constructs are facts that the human knows or assumes to be true. According to Hollnagel [53], it is comparable to what a schema is in the Schema Theory. Competence, control and constructs are captured by the modeling of human action, a cyclical model: the COCOM.

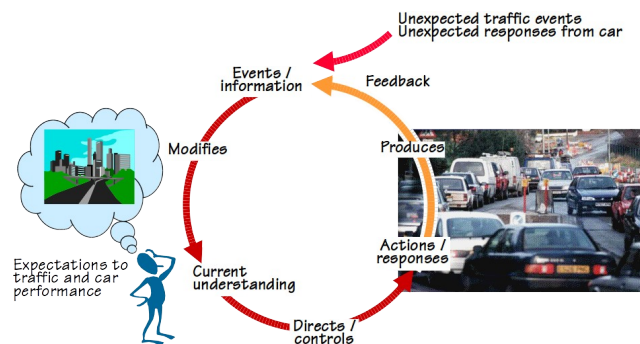


FIGURE 1.18: The cyclical model of human action by Hollnagel, applied to driving [53]

In the Figure 1.18, we see how the constructs (the current understanding of the situation) direct actions, *i.e.* the realization of competence through a control mode. The model is also designed in a cyclical manner to better account for time. The time of cognitive processes is not disconnected from the activity and context. Hollnagel argues that, in actuality, we always have limited time to make decisions, and the timing of the action is the one that matters. What's more, humans typically do not execute one process at a time, but several, each with different timings and priorities. Time is the main variant that will determine the control mode that the human can afford: the time to evaluate events, needed vs. available, the time to select an action, needed vs. available and the time to perform an action, needed vs. available. Hollnagel goes on to present ways in which control can be facilitated by reducing each of these "times needed". For example, the time needed to evaluate events could be improved by presenting the information differently, by filtering or prioritizing. The time to select actions can drop when the controls are easily available and identifiable. For Hollnagel, it often comes back to improving the HCI or Human-Machine Interaction (HMI) interface.

Later on, Hollnagel develops a method to analyze work activities, the Functional Resonance Analysis Method [77] (FRAM). The FRAM is not a method to analyze risk per se [60], however applying it to an organization can be a starting point to a risk analysis (but also other endeavors, *ie.* a restructuration). The modelization of

- **Equivalence of success and failure:** So-called “good” and “bad” actions are produced by the same processes. The fact that the outcome is different does not mean that rights actions and erroneous actions are different in nature.
- **Approximate Adjustments:** The work conditions and environment are almost never what they are prescribed to be. Socio-technical systems are complex and resistant to being spelled out. Workers are constantly adjusting to the reality of their context and resources. Their performance vary, depending on this adjustment.
- **Emergence:** An accident or an error in the system emerges from the concomitance of functional variations and their interaction. This idea differs from pointing to the malfunctioning part of the system that resulted in a failure. This attribution of blame cannot explain the phenomenon by itself. The error emerges from the system in its entirety.
- **Functional Resonance:** The variability of the outcomes of a function may cause another one to be amplified and vice-versa. The FRAM says they resonate with each other.



Rather than representing the parts of the system, the FRAM represents the interactions at play within a system, causing its complexity. The method is qualitative, one can adapt it to their context of study. Its modelization usually results in a node diagram as in Figure 1.19, meant to represent how the system is supposed to work. In 1.19, the nodes are actions related to a batch process, while the links are conditions needed to go on from one action to the next. Hollnagel clearly moves away from linear or cyclical models, allowing here for multidimensional dynamics.

The *New View* is a movement in Safety Sciences where authors take a stance against the overuse of quantitative, blame-oriented methods. The models and methods developed are more qualitative. They tend to focus on the organization as a complex system, and not on how a person makes a mistake. However, this approach is criticized for its lack of application clarity and experimental justification.

What opposes *Old* and *New*?

Read et al. [146] describe the shift from the *Old View* to the *New View* as a “*shift from human performance to systems ergonomics*”. They defend that *New* approaches are increasingly popular because they better tackle with sociotechnical systems and their issues. When an incident occurs in a system, pointing to human error should not be the conclusion.

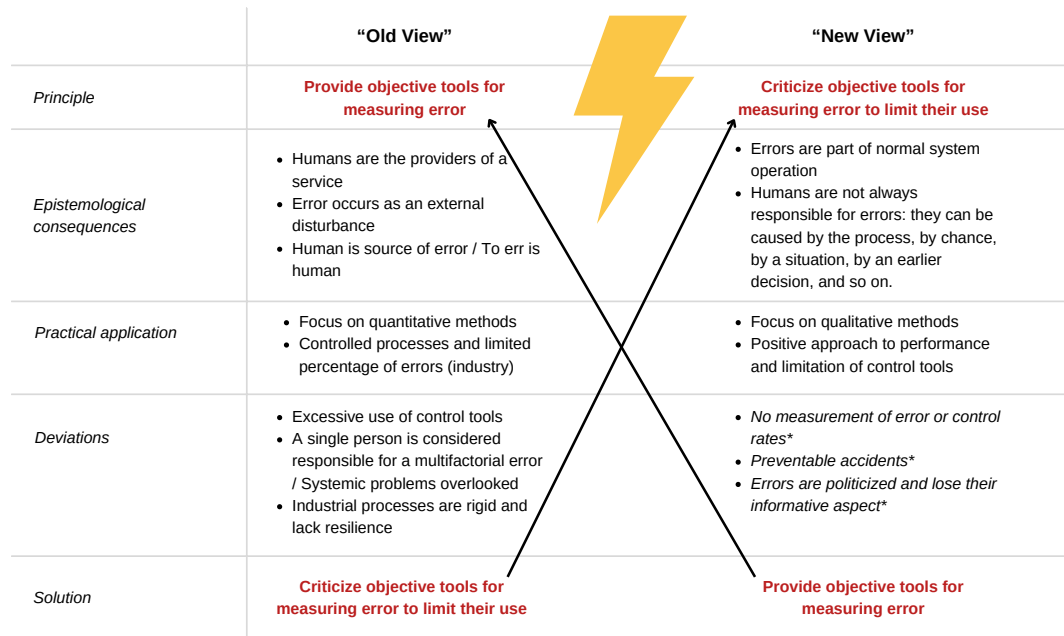
However, according to A. Cooper [37], *New View* practitioners do not propose alternative error management tools, but rather implement existing tools (derived from *Safety-I* methods) while taking a stand against its epistemological implications. M.D. Cooper (a different Cooper) is adamant on the subject, defending that *Safety-II* –the *New View*, is a case of “*ideology and emotion [...] triumph[ing] over science and practice*” [38]. According to him, “The Emperor has no clothes”²; in other words, the understanding of the “*New View*” being efficient is a shared misconception. In Cooper [38]’s view, the *New View* is a matter of “positioning” and of “movement”, in response to the improper use of *Old* methods. In other words, the starting point is not the *Old* theories and methods themselves, but their application. As a consequence, Cooper [38] states, most of the criticisms that the *New View* directed at the *Old* one were already addressed in the criticized models.

The *New View* also introduces the intention to measure performance positively, by putting good actions forward. According to Cooper [38], the desire to “focus on what goes right”, is a dead end, as in practice we always end up having to measure errors. To measure good performance, one has to define what is good and what is straying from that. Even Positive Performance Indicator (PPI), which could be a direct application of the *New View*, measures in fact Occupational Safety and Health (OSH) performances, not safety performance. Furthermore, the *New View* wants to bring the focus on adaptive capacities to create Safety. For M.D. Cooper, this leads to underestimating the presence of negative events and their consequences.

The *New View* here does not offer tools for applying Safety differently. Cooper [38] conveys that it failed to offer alternative methods to measure Safety. Instead, it brings insights for understanding it differently.

Comparing the *New* with the *Old* approaches, we can notice that the biggest and most fundamental change is a shift in scale. Where the *Old View* puts forth systems to analyze human behavior, the *New View* favors organizational approaches. In theory, the two views do not compete with two different positions on the same problem, they simply do not solve the same problem. The opposition is epistemologic before anything else.

²As an idiom, “The Emperor has no clothes” or “The Emperor’s new clothes” refers to a situation where something is generally believed to be true and praised because most people are reluctant to oppose it or to go against the flow [177]. “The Emperor’s New Clothes” is a short story by Hans Christian Andersen published in 1837, where an Emperor is fooled by two weavers selling him an imaginary cloth, supposedly only visible to the wise. The Emperor and his people all pretend to see the non-existent cloth by fear of appearing idiot.



*No proof of such deviations have been found in the literature

FIGURE 1.20: Tension between the *Old View* and the *New View*

To represent this conceptual tension as an antagonism, we use the formalism of the Figure 1.20, borrowed from Meirieu [119] and inspired in its use by Salzmann [160]. By defining a *View*, its principles and epistemological implications, we can deduce its possible deviations. We then observe that they are resolved, for each *View*, with applying the fundamental principle of the other. In a sense, this is an expected result, since the *New View* is, in a way, a reaction to the existing drifts in the application of the *Old View*. On the other hand, it is interesting to note that the possible excesses of the *New View* are theoretical, at least in part. As we saw before with [146] and [37], the *New View* does not always put aside the *Old* performance measure tools, it can incorporate them in accident investigation without making them the center of it. In our schema, the deviations of an application of the *New View* are therefore not the result of field observations, but the conjectures of its detractors.

One must be careful here not to confuse a *View*, the models and theories, for its application in error management in organizations. The opposition of the two *Views* here is in fact only possible because they are applied to the same problem: how to find causes to the happening of incidents? The *Old View* looks at how we can model a human accomplishing a task, and from there infer where in the process an error might have occurred. Because it takes roots in cognitive information processing models, the *Old View* will tend to blame individuals when it is applied to organizations. The *New View*, on the other hand, looks at how we can model an organization to infer where an error might have originated. Because modeling methods must apply to diverse organizations, it has to be qualitative to be highly adaptive. Relevant measures can then be defined from this analysis.

The models of the *Old View* focus on individual cognitive information processing. They are not problematic per se, but their use in explaining problems occurring in organizations favors certain interpretations over others. In short, the *New View*

did not prove the *Old View* to be wrong at all. However, in the context of accident investigation in complex systems, its application seems to increasingly prevail.

The common ground

Looking at the visual opposition above between *Old* and *New View* (Figure 1.20), one could imagine that organizations also “oppose” each other in practice. Companies would abide by one strategy, one *View* to build their Safety “toolbox”. This subsection presents how, on the contrary, the practice of Safety tends to be more variable and case-dependent than theory could suggest. As Carroll et al. [32] put it, “*safety is a social construct [...] rather than a unified approach*”.

One way to observe this “*social construct*” is to study the people that practice safety and their conceptions of it. In their work, Carroll et al. [32] investigate this very topic, classifying the perceived causes for safety among its practitioners. In a first study, they asked open-ended questions about what participants considered to be the causes of safety and lack thereof. The responses had some commonalities, for example more than 25% of people thought blame culture to be harmful for safety. However, responses were still surprisingly diverse. Even when the participants were given causes to rank in a second study, the most popular Top-5 was shared by less than 30% of respondents. Some values and practices for safety seemed in fact to depend on demographics. For example, people with “safety”, “risk” or “investigation” in their job description were more likely to rank fairness and trust high as a cause for Safety, and collaboration low. Males also seemed to value leader’s example more, and people with higher education ranked learning higher. In fact, other research show that even within one company, perspectives on safety and accidents will differ depending on the profession and the position in the hierarchy [62, 18]. The perceived causes of system safety do not seem to be universal. Instead, people seem to have different priorities and values relating to accident prevention. Safety practitioners negotiate their decisions depending on those differences.

It can also happen that people are used to different safety models. However the negotiation is not as heated in the real world as they can be in academic circles. Grant et al. [66] examined the 5 most cited system safety models and their literature (based on their authors), and extracted accident causation tenets. The 5 models are: Leveson’s Systems-Theoretic Accident Model and Processes [106] (STAMP), Rasmussen’s risk management framework [143], Perrow’s Normal Accident Theory [137] (NAT), Dekker’s philosophical approach [42] and Hollnagel’s FRAM³. Fifteen tenets, described in tab 1.7, came out of this study, and almost all of them were included in every examined models. These tenets are defined both positively and negatively; they are the cause for safety and the cause for accident depending on their presence and implementation in the system. Despite the apparent diversity of the most cited models, they overall share the same focuses and principles, and the same ground rules for system safety.

³Dekker and Hollnagel are usually considered belonging to the *New View* movement. For Rasmussen, while his work with the SRK is representative of the *Old View*, he later endorsed a more system-thinking standpoint, particularly with the framework in question here. In [143], he modelises a system according to functions, roles and responsibilities and their interactions. With this work, Rasmussen places himself in a *New View* approach [146]. Leveson’s and Perrow’s approaches are also qualitative, aimed to accurately describe the way a system is working in order to better predict what can fail and lead to an accident.

Tenet	Simplified definition
Vertical integration	Interaction between levels in the system hierarchy
Constraints	Influences that limit the behaviours available to components within a system
Functional dependencies	The necessary relationships between components in a system
Emergence	An outcome or property that is a result of the interactions between components in the system that cannot be fully explained by examining the components alone
Normal Performance	The way that activities are actually performed within a system, regardless of formal rules and procedures
Coupling	An interaction between components that influences their behaviour; both tight and loose interactions
Linear interactions	Direct cause effect relationships between components where the outcome is predictable
Non-linear interactions	Interactions are complex. Relationships between components where the outcome is not predictable
Modularity	The organisation of a system where sub systems and components interact but are designed and operate largely independently of each other
Feedback Loops	Communication structure and information flow to evaluate control requirements of hazardous processes
Decrementalism	Small changes in normal performance that gradually result in large changes
Sensitive dependence on initial conditions	Characteristics of the original state of the system that are amplified throughout and alters the way the system operates (interconnected webs of relationships)
Unruly technologies	Unforeseen behaviours or consequences of technologies
Performance variability	Systems and components change performance and behaviour to meet the conditions in the world and environment in which the system must operate
Contribution of the protective structure	The organised structure and system control that are intended to optimise the system, instead they do the opposite

TABLE 1.7: Tenets in system safety and their simplified definition according to Grant et al. [66]

It seems that the main point of possible discrepancy when it comes to safety application is not in defining which practices are safe and which are not, but in prioritizing these practices. The bigger and more complex a system is, the more constraints it is going to have. Some of those constraints are contradictory and yet non-negotiable, like control and adaptiveness, or productivity and risk reduction. An organization safety practice's goal is to navigate these conflicting goals, setting priorities without obliterating a necessary component of its system. Rochlin [158] suggests that "[s]afety is in some sense a story a group or organization tells about itself and its relation to its task environment."

1.3 Sparse approaches of error in HCI

1.3.1 Existing taxonomies in HCI

Classifications of “human errors”

The importance of understanding human error is not limited to critical settings. The field of HCI also tends to single human errors out as specific phenomena and objects of study. What does a human error look like on an interactive system? Researchers develop models, not only of human behavior, but of humans interacting with a system. To that end, they reinterpret existing models or theories from Safety Sciences, Cognitive Sciences and Psychology. The studies presented in this section are representative of the different angles existing in HCI when specifically talking about “human errors”.

Types and causes of human errors One of the most cited studies on human errors is Norman [130]’s categorization of slips⁴. To him, they are errors in carrying out an intention, *i.e.*, the goal is correct but its execution by the user is wrong. His study aims to develop principles to help design based on how people commit errors.

Norman [130] categorizes slips based on their causes. What causes a human to commit an error? Following schema theory, here, a human error is an error in the activation of a schema. In the correct course of action, a schema is activated and carried out if and when its trigger conditions become true. In a wrong course of action, a step is not performed correctly somewhere in this operation. Table 1.8 summarizes the different ways in which a schema activation can fail (under “Slips Resulting from Faulty Activation of Schemas”).

Mode errors Outside of the scope of schema activation, Norman [130] also accounts for errors in the formation of intention with mode errors and description errors. A *mode error* is a matter of interpretation by the person: they are acting correctly according to a certain mode of the system, but the system is not in that mode. The article proposes solutions to this kind of problem: not have modes at all, make them very distinct from each other, or make the possible commands in each mode different: if a system has several modes, their respective commands should be and look different. *Description errors* are the counterpart of mode errors. The problem is not with the interpretation of the user, but with the clarity of the interface: the mode is not visible or clear. In other words, the system is to blame. Again, the author states that such types of problems are preventable with a change of design.

This study is centered on human errors and how system design can foster or prevent their occurrence. For example, it mentions the need for a system to be consistent, because when users lack knowledge about the system they are using, they are prone to function by analogies. The system must then encourage similar designs and commands to operations that are alike, for the user analogies not to cause problems. A similar observation is made with capture errors (see 1.8). When the sequence of actions is similar between a familiar operation and a less familiar one, users will tend to fall back in the familiar course of action when performing the less

⁴As is explained in section 1.1.1, a slip error for Norman has a broader definition than the one that I use in this thesis.

A Classification of Slips Based on Their Presumed Sources
Slips in the Formation of Intention
<i>Mode errors:</i> Erroneous classification of the situation
<i>Description errors:</i> Ambiguous or incomplete specification of the intention
Slips Resulting from Faulty Activation of Schemas
<i>Unintentional activation:</i> When schemas not part of a current action sequence become activated for extraneous reasons, then become triggered and lead to slips
<i>Capture errors:</i> When a sequence being performed is similar to another more frequent or better learned sequence, the latter may capture control
<i>Data-driven activation:</i> External events cause activation of schemas
<i>Associative activation:</i> Currently active schemas activate others with which they are associated
<i>Loss of activation:</i> When schemas that have been activated lose activation, thereby losing effectiveness to control behavior
<i>Forgetting an intention</i> (but continuing with the action sequence)
<i>Misordering the components of an action sequence,</i> including skipping steps and repeating steps
Slips Resulting from Faulty Triggering of Schemas
<i>False triggering:</i> A properly activated schema is triggered at an inappropriate time
<i>Spoonerisms:</i> Reversal of event components
<i>Blends:</i> Combinations of components from two competing schemas
<i>Thoughts leading to actions:</i> Triggering of schemas meant only to be thought, not to govern action
<i>Premature triggering</i>
<i>Failure in triggering:</i> When an active schema never gets invoked because:
• Action was preempted by competing schemas. • There was insufficient activation, either as a result of forgetting or because the initial level was too low. • There was a failure of the trigger condition to match, either because the triggering conditions were badly specified or the match between occurring conditions and the required conditions were never sufficiently close.

TABLE 1.8: Classification of slips *i.e.* “error[s] in carrying out the intention” by Norman [130]

familiar operation. Again, this can be prevented with the clear discrimination, in design and commands, of different operations. For Norman [130], the solution overall is to make the system resilient to errors. Actions should be reversible and if not, actions that can lead to big consequences should be difficult to do, so that they cannot be done by accident.

Cognitive models Schema theory is far from being the only human behavior model on which we can base a theory of human error. Halbrügge, Quade, and Engelbrecht [69] present a cognitive model on sequential behavior and error. They focus of procedural error, *i.e.* errors in the execution of an operation that make the process inefficient, sub-optimal. They can either be “the addition of an unnecessary or even hindering action, which is called an intrusion. Or a necessary step can be left out, constituting an omission” [70]. It is based on the Memory For Goals [5] (MFG) cognitive model, that predicts goal formulation by users based on the type of internal memory available

to them. In the MFG modelization, the substeps in a task, towards a goal, are represented in human memory. This model has been used in HCI to explain different types of procedural errors, as in [69, 70]. In [69], Halbrügge et al. elaborate on the MFG model to not only represent errors due to memory faulting, but also errors induced by the environment. This updated model is centered on situations where the human is at fault and does not account for errors caused by bad design. The authors then designed an experiment to test it [70], in an article that studies how visual interactions can foster human-induced errors with expert users. Testing how participants behave relating to their own memory and visual cues, they observe that the weaker their memory, the more they seem to rely on the environment. Their cognitive model, featuring *knowledge in-the-head* (memory) and *knowledge in-the-world* (interface, environment) is represented in Figure 1.21.

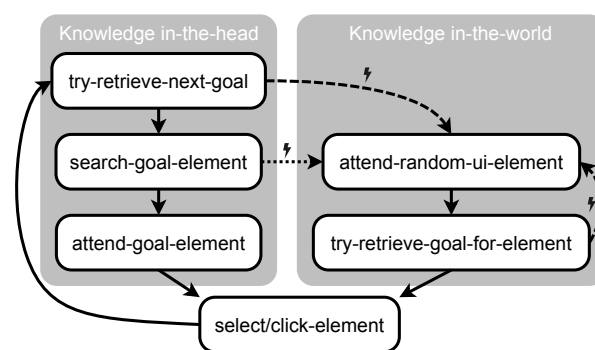


FIGURE 1.21: Simplified flow chart of Halbrügge, Quade, and Engelbrecht [70]'s cognitive model. Dashed arrows represent a retrieval error, and the dotted ones represent a failure in visual search.

Task-centered models Rather than taking roots in human behavior modelizations, some studies on human errors take another point of view: a task-centered approach. This kind of angle has precedents in HCI that go back to the 80's. Task-Action Grammar [135] (TAG) models are generative methods to come up with sets of rules to write action specifications in a task. In TAGs, the user has a library of simple, cognitively automatic tasks, and the number of possible task sequences is the number of rules required to describe the system [20]. Goals, Operators, Methods and Selection rules [31] (GOMS) is a human information management model, as the ones described in section 1.2.3, that originated in the field of HCI. It can be considered a task-centered approach as, according to Booth [20], GOMS is used to predict how the execution of a task will play out. It specifies the actions of a task, the goals, the sub-goals, the operations needed to obtain them. Both TAG and GOMS models have been used to analyze errors, but were not the most relevant methods according to Booth [20], because they were not created to that end.

Fahssi, Martinie, and Palanque [55] advocate for a task-centered approach. The authors observe that human error is usually not represented in task modeling, upstream of the design process. Indeed, errors are not part of the goals set at that stage. The article supports a representation of human error in the task model, made to better understand how things can go wrong, or at least how they went wrong.

To that end, the authors propose Human-centered Assessment and Modeling to Support Task Engineering for Resilient Systems [55] (HAMSTERS). It is a notation tool that allows for a detailed description of tasks, taking into account erroneous

behaviors. It represents in a task chart the user tasks (see symbols in table 1.9) as well as information known by the user or manipulated within the task (see codes in Figure 1.22). Lastly, the model represents the temporal links between sub-tasks: do they have to happen one before the other, simultaneously, or do they have another logical relationship? Table 1.10 presents the operators that link the sub-tasks with each other.

	Abstract	Input	Output	I/O	Processing
Abstract	 Abstract	Not Applicable	Not Applicable	Not Applicable	Not Applicable
User	 User abstract	 Perceptive	 Motor	 User	 Cognitive
Interactive	 Abstract interactive	 Input	 Output	 Input/Output	Not Applicable
System	 Abstract system	 Output	 Input	 Input/Output	 System

TABLE 1.9: HAMSTERS task types



FIGURE 1.22: HAMSTERS representation for objects, information and knowledge

Operator type	Symbol	Description
Enable	T1>>T2	T2 is executed after T1
Concurrent	T1 T2	T1 and T2 are executed at the same time
Choice	T1[]T2	T1 is executed OR T2 is executed
Disable	T1[>T2	Execution of T2 interrupts the execution of T1
Suspend-resume	T1 >T2	Execution of T2 interrupts the execution of T1, T1 execution is resumed after T2
Order Independent	T1 = T2	T1 is executed then T2 OR T2 is executed then T1

TABLE 1.10: HAMSTERS operators

The authors then carry out an example of the use of the HAMSTERS notation. They consider the task of a pilot checking the weather conditions during the flight. In Figure 1.23, we see that the notation is a tree, where each node is a sub-task. Each node has a task type and known processed information. The links in the tree are coded with the operators detailed above.

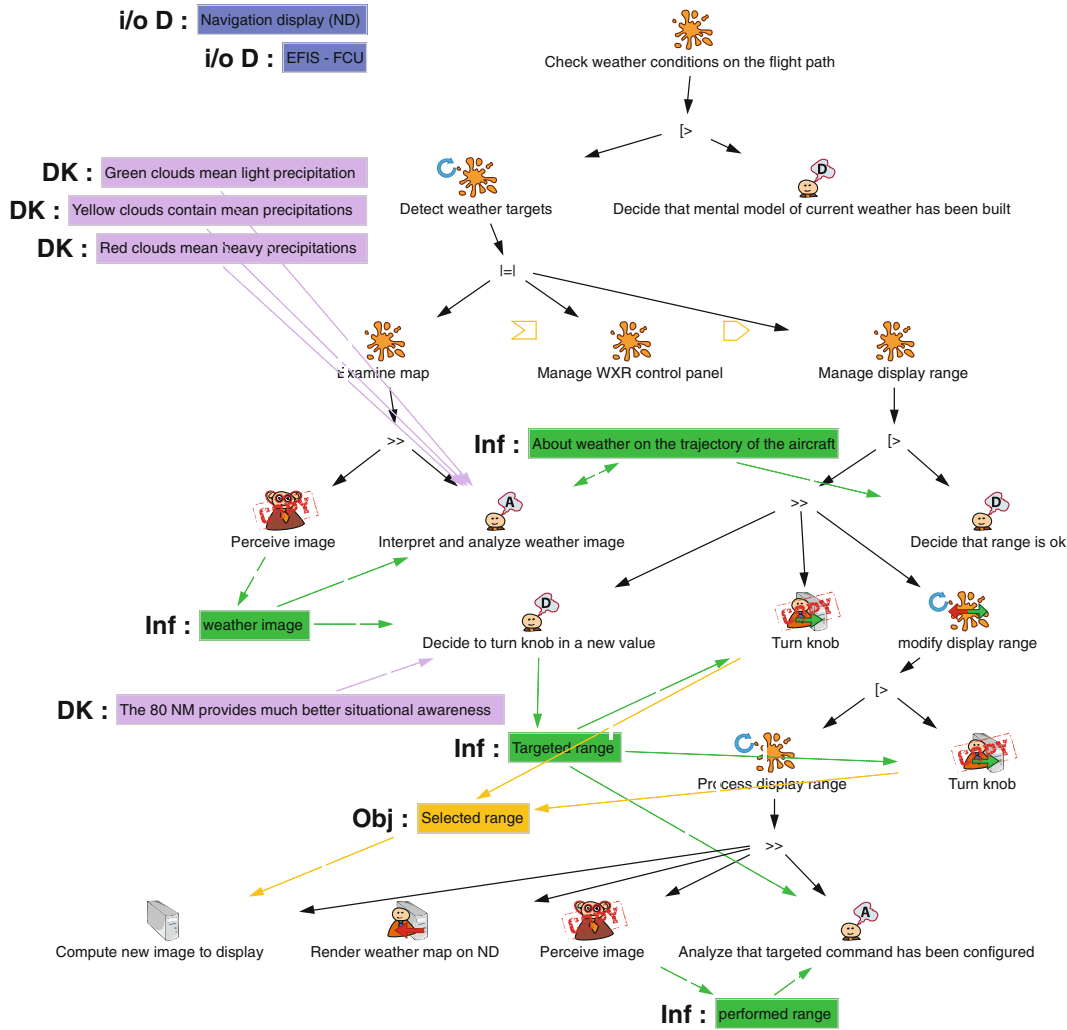


FIGURE 1.23: Checking the weather conditions HAMSTERS task representation

Fahssi, Martinie, and Palanque [55] integrate their tool within a HEI technique. They present in the article a correspondance between HAMSTERS notation elements and GEMS error types. This type of tool is made to prevent user errors by taking them into account from the design stage.

When HCI tackles “human error” as a whole, it tends to favor approaches that allows for a modelization early in the design process. To that end, they will rely on existing models, of human behavior or of the task. The theories that the domain uses are diverse for two main reasons. First, HCI can use theories from different domains like cognitive sciences, activity studies, human information processing or, in the case of human errors, safety sciences or resilience engineering. Second, even if we were to restrict ourselves to one source domain, they exists numerous ways to modelize how humans carry out a task, as supported by the many human information processing models presented in section 1.2.

Global classifications

Other than explaining human errors by modeling the user's information processing or task management, HCI can rely on interaction models to represent errors. Interaction is based on communicating information and interpreting it correctly, both for the system and the user. Incidents can emerge from this "discussion" between the user and the system. In the following are two studies of error in interaction, from different times in the history of HCI.

The Evaluated Classification Mismatch [21] (ECM) is a method of error evaluation that was developed by Booth in 1990. The work focuses on interpreting errors, that the author calls dialogue failures defined as *"breakdown[s] in communication between the system and the user; it is where either the computer or the user does not understand the other, or some information about the nature and structure of the task is not properly communicated"*. The evaluation process is made from the user's point of view. Their perspective defines what went right or wrong, depending on their view of the system and the task. The approach of error is in that way similar to the one in this thesis; an error is defined by the user and *their* interpretation.

In ECM, errors are classified in 4 steps (see Figure 1.24). The first step is to identify dialogue failures *i.e.* elements that do not match between the user and the system representations. Either the communication has stopped in some way, there was a misunderstanding, or the user needs help. The second step is to identify the object or operation associated with the dialogue failure. For Booth [21], an object is something the user can act upon in the system (a file, an application, a character string etc.). An operation is an action that the user can perform to manipulate those objects. In the third step, the type of the mismatch is classified according to its cause, either a concept or a symbol mismatch. When the user and the system do not have the same understanding of what the system does, and what sort of objects it operates, it is a concept mismatch. There, we can have operation-concept mismatch, where an operation is not understood the same by the system and the user, and surprises the person. Or an object-concept mismatch, where an object is not exactly what the user expects, in terms of form or content. On the other hand, a symbol mismatch is a misunderstanding between the user and the system based on the name or look of an element. One could say that the user is misled by the system design. Whether it concerns an object or an operation leads to an object-symbol mismatch or an operation-symbol mismatch. The last step consists in describing the mismatched elements, to understand its role in the dialogue failure. In using ECM, we seek to understand what happened, but also what was perceived by the user [20].

More recent work also analyzes error within the interaction happening between the user and the system. Palanque, Cockburn, and Gutwin [132] first present the Human-Computer Interaction Loop (HCIL) before introducing their taxonomy of errors. In Figure 1.25, the interaction loop is on the left, going from the output devices of the system, to the perceptual senses and motor actions of the user, acting in turn on the input devices. This is where interaction errors can happen. In this representation, the human and the system mirror each other with each their own "inputs" and "outputs". The human is modeled as an information processor that interacts with the system.

Their taxonomy is based on Avizienis et al. [9]'s (see section 1.1.2), modified to include incidents that can occur within the HCIL. For Avizienis, the user brings errors by their fallibility. Here, error-states that are induced in the operator, either by the environment or another person, are added. They distinguish the perceptual,

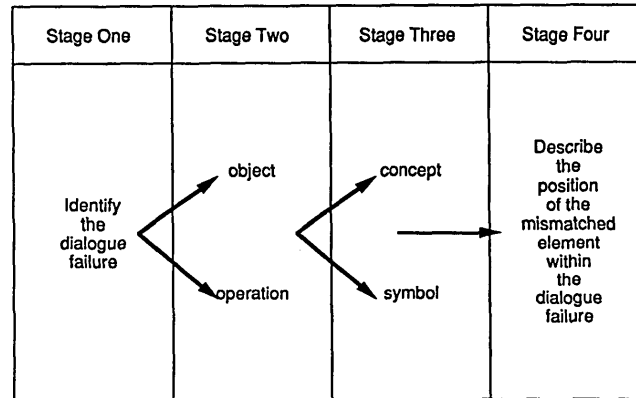


FIGURE 1.24: ECM classification process in [21]

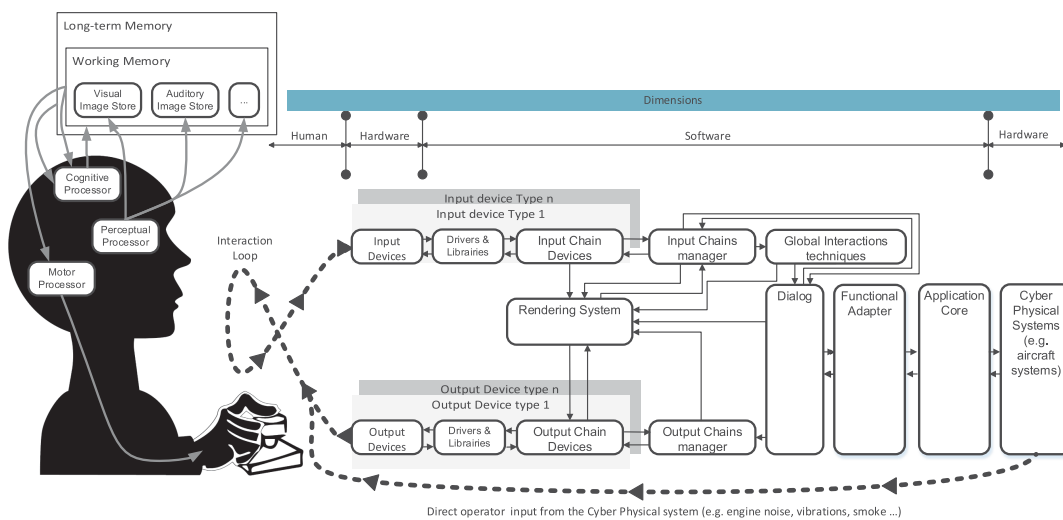


FIGURE 1.25: Multiple Input and Output Devices and Multiple Interaction Techniques [41] (MIODMIT) simplified by Palanque, Cockburn, and Gutwin [132]

motor and cognitive abilities of the human, like represented in the HCIL loop above (Figure 1.25).

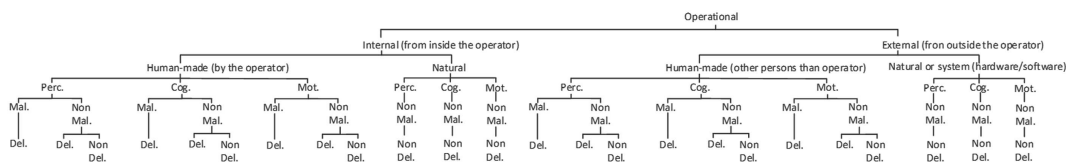


FIGURE 1.26: Palanque, Cockburn, and Gutwin [132]'s changes to Avizienis et al. [9]'s taxonomy

Avizienis et al. [9]'s fault classes (see Figure 1.3) are kept as well, except for “Capability”. Palanque et al. consider that this class focuses on the cause of the fault when all the others participate in its description. Palanque [131] later incorporates this new taxonomy of errors into the People, Organization, Interactive System, Environment [131] (POISE) framework, where people and interactive systems blend in an interactive socio-technical system.

It is interesting that theories of error, even in HCI, keep a strong lean towards

critical systems. More specifically, we recognize an industrial context, where the user is referenced as an operator and, as of Avizienis et al. [9]’s taxonomy, is considered a service provider.

The study of errors in the light of interaction design reveals types of errors linked to interaction itself. The articles in HCI explicitly tackling errors in general are however not numerous, but the field still shows a strong interest for them. Researchers tend instead to focus on specific error situations to investigate them, sometimes with special contexts or users.

Classifications for specific contexts

While some researchers tackle the subject of errors as a whole, most approaches take on a less global approach. This subsection presents taxonomies of errors, bound by specific use cases or users. This precision allows a deeper look into the specificities of certain contexts, and sometimes the highlighting of error situations uncovered by general taxonomies.

Shi et al. [168] for instance focus on typing human errors specifically in order to create a tool that simulates typing with human errors, eye and finger movements. Using other classifications on the subject, they construct a taxonomy of typographical errors, *i.e.* of errors in typed or printed text. Their taxonomy includes errors that emerged from tactile keyboards, such as mode errors (see table 1.11).

These errors are studied in context, to better understand (and potentially simulate) typing behavior. Dhakal et al. [47] report a large study analysing keystroking patterns of 168,000 volunteers, uncovering different typing behaviors. Specifically, they extract different types of typists from their results, from “slow and careful” to “fast rollovers” (pressing the next key before the previous one is released).

Typing errors are generally studied, not as a standalone phenomenon that would disturb a typing task, but as part of an overall typing behavior, that researchers aim to characterize.

Typographical errors

Insertion error: writing an extra character, *e.g.* “erroir” instead of “error”

Omission error: missing a character, *e.g.* “eror”

Substitution error, or misstroke: replacing a character by another one, usually nearby on the keyboard, *e.g.* “errpr”

Transposition error: inverting 2 adjacent characters, *e.g.* “errro”

Doubling error: doubling the wrong letter, *e.g.* “eroor”

Alternation error: doubling the wrong letter alternatively, *e.g.* “eroror”

Fat finger problem (with touchscreens): A typing error (wrong selection or multiple selection) due to a lack of precision of the finger, or keys being too close together

Mode errors (with modern keyboards): typing in the wrong keyboard mode, *e.g.* with Caps lock, “ERROR”

Autocorrection (with modern keyboards): the automatic corrector changes a word to an unintended one, *e.g.* “erratic”

TABLE 1.11: Typing errors from [168]

Pointing errors are similarly specific in that they refer to inaccuracies in target selection with an interactive device. They can happen in any setup or input device. Those errors tend as well not to be the object of specific diagnostics, as they rarely lead to critical consequences. Instead, they are generally used as performance indicators for pointing in a certain context (as in [116, 126, 179] to name a few). The study of pointing happens through models of how one “points”, rather than theories of errors in that context. Fitt’s law [58] is probably the most used model in HCI, but Müller, Oulasvirta, and Murray-Smith [123] underline that it is not the only one. Control theory, aiming to optimize a stability in the control of dynamical systems, offer different perspectives. While Fitt’s law [58] is an abstraction of pointing behaviors’ variability, control-theoretical models can foster a deeper understanding of those behaviors [123]. The interaction loop is represented in Figure 1.27.

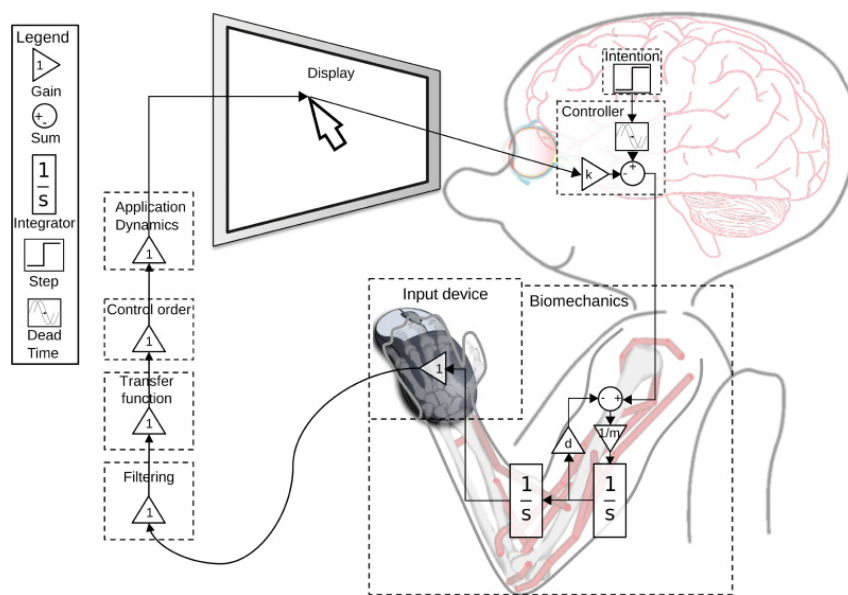


FIGURE 1.27: Müller, Oulasvirta, and Murray-Smith [123]’s control theoretic representation of the HCIL

Other contexts are specific not to the task, but to the device at play. Within the study of Augmented Reality (AR), Bahaei et al. [14] put together a taxonomy of errors. They focus specifically on faults leading to “human failures”. Departing from Avizienis et al. [9]’s taxonomy, and enriching it with the synthesis of other taxonomies of errors [142, 185, 74, 172, 61], they create a new taxonomy of faults, adding faults that are specific to the context of AR. Their approach share Avizienis et al. [9]’s view of the user: the human is a system element that is subject to failing. One can recognize in the Figure 1.28 the main fault types of Avizienis’ classification.

The specificity of an AR context adds possibilities in interaction, and in the same movement new types of error emerge. For example, because AR can be used to train for extreme environments or for mental health, one can observe errors in the rendering of these simulations: “restorative physical / mental health measure faults”. This technology can also be used in a learning setting, leading to possible “Interactive training / experience faults”. Research shows that it elicits certain responses in the brain (increased cognitive activity, memory encoding and visual attention), hence “brain elicitation faults”. Because less social presence is felt in AR contexts, there can be “social presence faults”. Faults can also happen in the very principle of the tool, in the integration of virtual elements within the perception of the real

world (“augmented environment faults”). The technology can make precise manual tasks challenging for the user, referred by “AR guided task faults”. Finally, faults may happen in the more macro adoption and legislation of AR within an organization (“organization and regulation AR adoption faults”). All these fault types are included in the classification of Figure 1.28 (left to right).

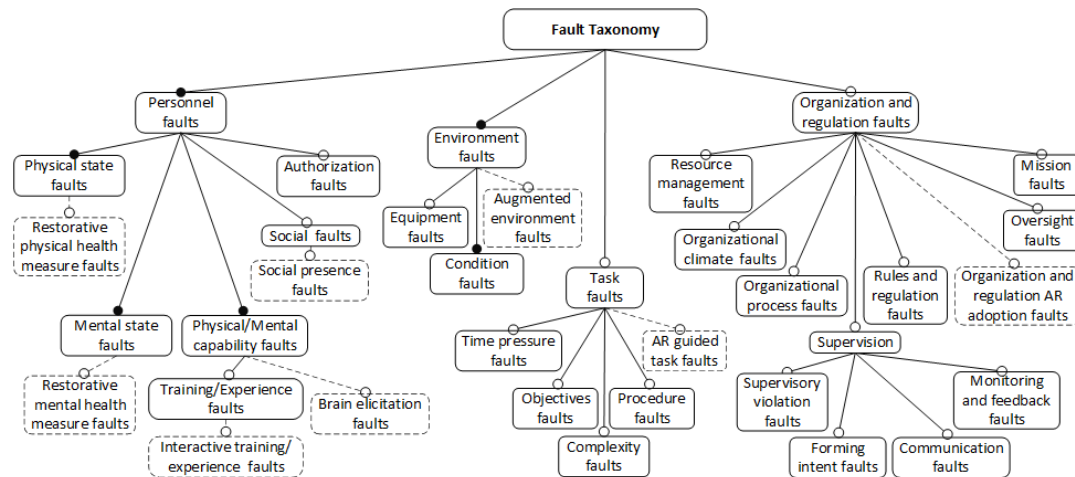


FIGURE 1.28: Bahaei et al. [14]’s taxonomy of faults. The dotted faults are AR-related.

In this taxonomy, errors are not only deduced from the explicit functionalities of a certain technology—in that if they are not delivered, it constitutes an error. Errors can also appear in the expectations that past use of the technology have created. Because we observed that AR elicits a certain mental state, we expect it to produce this effect every time. When Avizienis et al. [9]’s work defines failure (and in turn, errors and faults) according to the system’s specifications only, this approach includes cases of expectation violations as well. Their concept of error seems to cover both a system-centric approach and a designer-expectations approach.

Bugs are specific kinds of errors in that they are specific to the system. The user cannot “bug”, except in a metaphorical sense. In [120], a bug is defined as “*an unexpected defect, fault, flaw, or imperfection*”. In pointing fingers, we point to the system by definition.

It may be implicit or unconscious, but every scientific endeavor has an epistemological stance. A common one regarding bugs is *positivism*, the idea that there is an absolute truth that IT professionals refer to when classifying a bug. Widder and Goues [184], on the opposite, defends a *constructivist* approach, according to which truth is always dependent on context and on the person who validates it as such, and is always socially biased. The classification of what constitutes a bug is not a process based on an “objective” truth; it depends above all on consensus. For example, Open Source Software (OSS) projects classify bugs in 2 categories: problems that violate developers intent and everything else [35].

A bug deserves to be named as such and possibly corrected only if different people recognize it as problematic. This has direct consequences on the types of bugs that will be corrected.

Chilana, Ko, and Wobbrock [35] verify this thesis in their research on bug report descriptions. They observe that, in bug reports from users, bugs are synonymous to the system not meeting expectations. In their analysis, bugs are classified according to the source of the expectation that was violated (see table 1.12). The authors report

that bugs about “standard”, “genre conventions” and “prior behavior” were more likely to be marked *INVALID* by developers, considering that a certain behavior was expected and normal. On the other hand, violations of “specification”, “community expectations” or “runtime logic” were more likely to be fixed. Overall, their results show that more than half of bug reports are not bugs in the eyes of developers. Bugs have a greater chance of being fixed if they are expressed as a community expectation rather than a personal expectation.

In this thesis, an error is defined as an user expectation violation, as are bugs here.

Classification of different sources of expectation violations
Conventional notions of a bug as a violation of developer intent <i>Runtime logic</i> : Explicit violations of some runtime expectation, including errors, warnings, assertion violations, crashes, and hangs (e.g., “...scary deadlockassertions exiting mozilla after referencing <i>nsInstallTrigger</i> ...”). <i>Specification</i> : An agreed upon functional requirement among the developers (e.g., “There’s an incorrectly placed <i>PR_MAX</i> in the code for pref width distribution of colspanning cells.”). <i>Standards</i> : Specifications shared by the industry in which the application is deployed (e.g., “ <i>codebase</i> attribute of the <i>HTML4.0OBJECT</i> element is not supported...”).
Other sources of expectations, outside the scope of the implementation, developer community, or industry <i>Reporter expectations</i> : A reporter’s personal perspective about what the system should do (e.g., “Every time I Sort By Name by Bookmarks Firefox sorts and closes my Bookmark menu... Why does it do this??”). <i>Community expectations</i> : A reporter’s belief about a “typical” user’s expectations, including specific references to user, users, user interface, or usability. (e.g., “The preference to not show the tab bar when only onetab is open could be set to false by default. This would at least alert a new user to the possibility that tabs exist) The old tabbed browsing preferences could be returned.”). <i>Genre conventions</i> : References to applications with similar functionality; allusions to how a specific feature behaves for the same action. (e.g., “Firefox does not limit the slideshow horizontal size to the window width. The same source works correctly in IE.”). <i>Prior behavior</i> : References to the prior desirable behavior of the system (e.g., “The latest version of Firefoxonly imports one certificate from each file. I used to import all certificates previously.”).

TABLE 1.12: Classification of expectation violation in bug reports by Chilana, Ko, and Wobbrock [35]

Rather than specific contexts, some research focus on specific users. Kitajima and Polson [89] presents a model for analyzing errors of expert users of an interactive system. They list out three tasks in which these users may commit errors: display design, object selection, and selection of the next action. Expert users may lack attention and are confronted to a trade-off between speed and accuracy. Because these users have specific capabilities, they elicit specific types of errors that other users may not encounter.

The taxonomies presented here reveal that errors are bound to their contexts, more specifically, that the context conditions the possibilities of errors. At the same time, the situation will impact the very notion of what an error is: it is negotiated by the people involved in the interaction, be it users or designers. A phenomenon can be an error to be solved in a certain setting, just an annoying circumstance in another, or even not problematic at all.

Studies focussing on specific error types

In HCI, most of the studies about error do not tackle the problem in general, but rather a specific situation (see above). Here, I will present works that, without theorizing or classifying errors, participate in their study by diving on a very specific issue. These articles aim to bring attention to a certain type of phenomenon, or to investigate solutions to them happening.

Procedural errors are another specific type of errors, in that they happen during the execution of a task, not inducing critical consequences but slowing down a process or making it sub-optimal [70]. Among them are postcompletion errors that, as their name suggests, occur after a task is completed. Examples include forgetting the originals in the copy machine (after making copies): this oversight does not impact the main task results— the fact that the copies were made correctly— but can impact future tasks. Studies show that this kind of errors can be reduced with a better UX design [29]. Byrne and Davis [29] also report that training the user proved ineffective: informational callbacks and even blame did not improve their performance. This kind of work informs on the practical handling of errors, and how some methods might be effective to mitigate some problems, but not others.

Other research work shed a light on settings that can be more error-prone for users. For example, according to Ament et al. [6], users tend to do more mistakes when doing a device-oriented action rather than a task-oriented action. A possible explanation is that actions that are perceived less important receive less attention on the part of the user. This hypothesis prevails, *inter alia*, in MFG. The model assumes indeed that tasks that are not directly relevant to an actual goal are not strongly represented in memory.

HCI also concerns itself with correcting the system. While in some cases, the system can be seen as having the authority on what is right or wrong, other cases of error see the user as the reference on that front. Voice interface errors are such cases, researchers generally put the blame on the system and not the user [71, 94]. Even though, pointing to the user would be possible; one could argue that they need to adapt their articulation, speech volume, accent etc. The common stance on voice interfaces though is that they should be the ones adapting to how users speak, and not the other way around.

This stance can apply to other use cases, such as Large Language Models (LLMs). For Epperson et al. [52], an LLM system giving wrong factual information is an error. This is not however a universal position on these technologies, other people consider that they were developed to mime language, not to give true informations. Vitali-Rosati [181] considers for instance that LLMs were not developed to be factually reliable, and that users should not expect this from them.

Lastly, some studies investigate the various impacts that an error type can have on users, depending on its context. In a controlled experiment, Lafreniere et al. [96] test cognitive load, frustration and user preference with false positives and false

negative errors in different contexts. Both those phenomena are input errors: false positives happen when the system recognizes an intentional input from the user when there wasn't one, and false negatives when the system does not recognize an intentional input as such. These kind of errors are usually handled by tuning a threshold in the way the system interprets inputs. This study is interesting in that it does not angle for a better accuracy, but for a better understanding of what suits users the most. It is in turn an approach on errors that is close to the angle taken in this thesis.

If the theory of error in HCI is very influenced by other fields of study (Safety Sciences, Resilience Engineering, Psychology etc.), researchers practices point to a larger definition of what is problematic in the use of interactive systems. More specifically, there seems to be a shared understanding that *what poses a problem to the user is worth solving*. Reducing error occurrences seems to be the main focus, but some studies aim to understand why and how an error is problematic. Even when focusing on specific situations, these investigations deeper our understanding of user behavior.

1.3.2 The industrial heritage of HCI in user-experience evaluation

The previous sections highlight the influence of engineering and critical systems on the notion of error in HCI. However, this heritage does not only concern the theories and methods around error, but the evaluation of user experience as a whole. Evidence of this can be found in HCI's widespread use of assessment methods from these fields. Here, I present the 2 standardized questionnaires most widely spread in HCI.

The System Usability Scale [26] (SUS) was developed by Brooke in 1984, as part of a usability engineering program [108]. It is an extremely popular tool to assess the perceived usability of a system. As a point of reference, 22958 citations can be seen on Google Scholar at the time of writing.

No	Questions	1	2	3	4	5
1	I think that I would like to use this system frequently.					
2	I found the system unnecessarily complex.					
3	I thought the system was easy to use.					
4	I think that I would need the support of a technical person to be able to use this system.					
5	I found the various functions in the system were well integrated.					
6	I thought there was too much inconsistency in this system.					
7	I imagine that most people would learn to use this system very quickly.					
8	I found the system very cumbersome to use.					
9	I felt very confident using the system.					
10	I needed to learn a lot of things before I could get going with this system.					

TABLE 1.13: The standard SUS questions. Scores go from 1 ("Strongly disagree") to 5 ("Strongly agree")

The SUS evaluates the *perceived* usability of a system, as opposed to the objective usability assessing effectiveness and efficiency [108]. This questionnaire is centered on user perception, and how they evaluate their experience using a system. According to Lewis [108] (2018), the questionnaire is universally considered reliable, valid and sensitive.

The same article presents researchs demonstrating the flexibility of the method. It is for instance possible to use the questionnaire while minorly changing the wording. It is also possible to remove one of the 10 questions without changing the resulting scores significantly.

Another very common tool to assess part of the user experience is the National Aeronautics and Space Association Task Load index (NASA TLX), developed by the NASA in 1988 [73]. This method was originally specifically dedicated to the evaluation of pilots' experiences in automated aircrafts [10]. McKendrick and Cherry [118] among many others state that this questionnaire is the most used tool for measuring mental workload, in any context. For them, the questionnaire is so popular that it almost now serves as a definition of the very concept of mental workload. As per Babaei et al. [10], the NASA TLX is the "*de facto gold-standard mental workload scale*". In HCI, the method is largely used to assess the mental workload involved in using a specific system, or one of its functionalities.

Mental demand
How much mental and perceptual activity was required - thinking, deciding, calculating, searching? Was the task easy or demanding, simple or complex, exacting or forgiving?

Very Low | Very High

Physical demand
How much physical activity was required - pushing, pulling, turning? Was the task easy or demanding, slow or brisk, restful or laborious?

Very Low | Very High

Temporal demand
How much time pressure did you feel? Was the pace slow and leisurely or rapid and frantic?

Very Low | Very High

Effort
How hard did you work mentally and physically to accomplish your level of performance?

Very Low | Very High

Performance
How successful did you feel in accomplishing the goals of the task? How satisfied were you with your performance?

Very Low | Very High

Frustration level
How discouraged, stressed, irritated, and annoyed versus gratified, content, and relaxed did you feel during the task?

Very Low | Very High

TABLE 1.14: The NASA TLX, table copied from [86]

Practitioners do not seem to hesitate to adapt the NASA TLX depending on the task at hand. For example, Adamczyk and Bailey [2] and Böhmer et al. [19] modify the test for their needs by replacing the Physical Demand by Annoyance, and by adding a question.

Many articles in HCI recommend caution when using this tool [118, 10]. The criticisms revolve, among other things, around the bias that a prior task can have on the results (the reported workload is higher after an easy task and vice-versa) [118]. For Babaei et al. [10] however, the biggest concern about NASA TLX directly comes from its success. People using the tool might do so uncritically, misunderstanding what it measures. This might be aggravated by the fact that mental workload does not have a clear shared understanding in HCI [10]. For example, practitioners might use the method to evaluate other concepts, such as cognitive load and cognitive workload [187]. For instance, Kosch et al. [90] employ cognitive workload, cognitive load and mental workload as one and the same.

NASA TLX and SUS are very widely used in HCI, to evaluate user experience. Consequently, they are often referenced when it comes to study the impact of errors on users.

Beyond the limitations that these questionnaires have as scientific methodologies, they were designed for specific contexts. They convey the constraints and priorities of engineering, industrial, and even critical environments. Usability and mental workload are important factors in user experience, but they are far from covering all the stakes of human-system interaction.

1.3.3 Error as a tool for research

In HCI, errors are informations for researchers to draw conclusions on systems. Most obvious use of errors for research purposes is to evaluate performance or accuracy. The error-rate of participants hints on their precision capabilities within the setup of the experiment (the system, the task and possible subtasks).

Pointing For pointing tasks, an error-rate below 4% in pointing tasks is generally considered good within the field. This norm in the domain can be attributed to the definition of Fitt's law, a mathematical method universally used to predict pointing behavior, where the time required to select a target is a function of the distance to that target and its width [115]. Within Fitt's law, a 96% accuracy is fitting because there, Fitt's law matches Shannon's law—modelizing entropy and noise in information theory. MacKenzie [115] recommend to aim for this percentage for the law to be applied correctly. While very popular, let us note that the 4% error-rate has been criticized in its very mathematical principle [65].

Because pointing tasks are everywhere in interaction, the assessment of user accuracy in performing them is essential. Some research also address the prediction of those performances, by attempting to predict the error-rates of those actions [189].

In pointing, errors are not only a phenomenon to avoid, they are engraved in the norms of research practices. Those conventions do not only impact scientific methods, but also the tools and designs created by them.

Informing design as a whole “*People will make errors, so make the system insensitive to them*” [130]. The ideal way one can mitigate errors is to prevent them from the design stage. Instead of blaming the user for making mistakes, or trying to educate them, a better design practice is to think about errors while creating interactions. To that end, Norman [130] presented a set of qualities to develop in a system to minimize errors and their effects. They are presented in table 1.15.

Those principles are the direct reflexion of an error analysis. One might assume that they would consequently be associated with bad designs to stay out of. They

are however formulated positively, not with things to avoid, but things to look for. Errors are a strong starting point to design directives, because they are very direct consequences of their application.

Design Principles

Feedback: The state of the system should be clearly available to the user, ideally in a form that is unambiguous and that makes the set of options readily available so as to avoid mode errors.

Similarity of response sequences: Different classes of actions should have quite dissimilar command sequences (or menu patterns) so as to avoid capture and description errors.

Actions should be reversible (as much as possible) and where both irreversible and of relatively high consequence, they should be difficult to do, thereby preventing unintentional performance.

Consistency of the system: The system should be consistent in its structure and design of command so as to minimize memory problems in retrieving the operations.

TABLE 1.15: Properties to look for for an error-insensitive system

Detecting errors There is no one solution for handling errors that will work for every context. This design work must be done again from the beginning every time [166]. Because of this, some articles investigate error detection on the part of the system. If the system was able to tell when an error happens and why, it would indeed be possible for it react in an appropriate way. Sendhilnathan et al. [166] for example train a Deep Learning model on a Virtual Reality (VR) task using gaze dynamic to spot human selection errors. In their approach, a human error is either a selection that is not relevant to the task (and possibly bears negative consequences), or a change of mind at the last moment about the selection.

An error-aware system could adapt over time to include new designs and recovery techniques. Other work on error detection focus specifically on error-prone settings to allow such adaptations to contexts in need [165].

The theory of error that is employed while detecting it has a great impact. Some phenomena might be overlooked in a specific model of errors. Also, a phenomenon might not be categorized the same in different approaches: what is a slip somewhere can be a misinterpretation somewhere else. The model that is used to detect errors conditions what can be seen.

Error simulation To help with error-aware system design, some researchers focus on simulating errors. Shi et al. [168] for instance simulate typing to predict performances and design better text entry systems. They argue for a modeling of errors based of their causes instead of enforcing statistically probable errors. Begosso and Filgueiras [17] take on a broader approach. Their human behavior simulator, S. PERERE, is an engineering tool meant to help with design. Here, the designer chooses what errors kinds in GEMS, are possible within the simulation. The goal is to test the system in precise ways: *“the erroneous situations are not obtained by simulation of human variability. If done this way, human erroneous performance would be closer to reality and exactly because of that, would be unpredictable and inappropriate to design purposes.”*.

It is interesting to see that errors need even more be precisely defined in practice. To simulate errors, the designer needs to have a full and refined picture of what errors are possible on the system, and which are potentially critical.

Feedback Another way to mitigate errors is to adapt the design so that they happen less, especially in critical settings. Borghouts et al. [22] improve the feedback on the number entry of an infusion pump dose. In their experiment, nurses see a magnitude representation of the dose they just entered, allowing them to detect large number entry errors before infusing a patient. In the same vein, Wiseman et al. [188] use a checksum to check such entry errors.

Adding constraints While design usually gears towards facilitating interactions, some research shows that adding constraints to the user can be beneficial. For errors, imposing pauses can reduce certain types of errors. Back, Brumby, and Cox [12] study the impact of enforced lockout periods in routine tasks after an interruption. Interruptions tend to increase the number of human errors: they can provoke a loss of focus on the task that can lead to errors. The authors explain that they are related to sequence errors, the completing of steps in the wrong order. Their experiment shows that an imposed break after an interruption, where the user can look at the device but not act on the interface, reduces those errors significantly.

Other studies show that making an interaction more difficult can sometimes mitigate errors. In [13], participants of an experiment had to check information on a prescription form before programming an infusion pump. When the form is close to the pump, people tend to adopt a perceptual-motor strategy: they check the pieces of information one at a time and program the pump accordingly. Doing so, they can multitask more easily on several pumps. When the form is further away, they rely more on a memory intensive strategy: they memorize the form information and then do all the programming of the pump at once. This strategy is slower and allows for less multitasking. However, doing several things at once can lead to omission errors, more present in the second setting. While less efficient, Back, Cox, and Brumby [13] find it to be less error-prone.

Those studies stand out in a domain that often takes few errors as a synonym for performance and efficiency. It begs the question of what we call performance.

Errors prompt diverse usages in HCI research, not only as a study subject, but also norms, design inspirations and design methods. In those practices, the theoretical framework is primordial; it states what is an error and how. In the next section, I will go into detail about the framework defended in this thesis.

1.4 Arguments for a different definition of error

Classical approaches of errors have defined the concept as a situation that leads the system to not be able to fulfill its function in the way intended at its conception. This definition is valuable in Safety Sciences or engineering, because they are fields in which the functionality of the system is the end goal, and where the opposite can even be dangerous. HCI inherits from those theories and models; an error, while using a digital device, is usually defined as something that prevents the systems from fulfilling its function. However, is that functionality the end-goal in this field? Or is it rather, that the user has an overall positive experience with the tool? In some cases, those two objectives are interlinked, because the tool fulfilling its function is what the user ultimately wants. However, some situations and phenomena highlight that

the traditional definition of error does not cover all user problems. In turn, this section presents arguments towards a definition of error centered on user-experience rather than system specifications: *an event, caused by or during an interaction with a digital device, that does not meet the user's intents, expectations and wants about that interaction.*

1.4.1 Missing link with individual experience: existing studies

The common definition of error brings forward material costs. Did the error cause a loss in performance, in terms of time or quality? Did the error cost money? Did it break something? On the other hand, other components of user experience are not – or less so – called into attention with that definition. Negative emotions, expectations and preferences are studied in HCI, just not as part of discussions about error.

This subsection presents some of those research perspectives, underlying the parameters of an unwanted experience that a different definition of error might bring to the table.

Frustrating events

One of the main emotions coming to mind when talking about error is frustration. It can be defined as *“a deep chronic sense or state of insecurity and dissatisfaction arising from unresolved problems or unfulfilled needs”* [120]. In the 2010's, studies of the subject report that close to half the time spent on computers is spent dealing with frustrating events [34, 102].

Hertzum and Hornbæk [75] recently ran a large study on frustration as a consequence of the use of devices. They report a decrease in the rate of frustrating events: in their 1 hour study, participants spent 11 to 20% of their time occupied with a frustrating event, which is still a significant portion. These events took between one second and an hour of their time. Also, participants reported that 84% of those events had happened before, and that 87% could happen again. Frustration may not be as omnipresent as before with changing technology but it appears unavoidable and repetitive.

From their study, Hertzum and Hornbæk [75] distinguish 25 categories of frustrating incidents, listed in table 1.16. One might notice that the level of frustration does not seem to correlate the frequency of incidents. The system freezing temporarily for example is a rather frequent incident but does not have a very high frustration median overall. Things being “just hard” is less frequent but significantly more frustrating. Of course, one should note that the less occurrences a category has, the less balanced their median will be. Although future work is needed to back that observation, those results are interesting in that they can be counter-intuitive. Repetition is indeed often related to frustration, but it does not seem to show over a 1 hour period.

A second observation over that list of frustrations is that some of them are commonly understood as errors, but not all of them. The system being slow or not working goes against the system functionality. Other categories seem to reference studied phenomena in HCI: “Accidentally make a wrong selection” could be related to a slip, “Popups” and “Cross-application tasks” being a source of frustration could hint at interaction interferences, while “Disruption of the flow of activities” clearly implies interruptions (see subsection 1.4.2). Dark patterns can also be the cause of “Privacy concerns” (see subsection 1.4.2).

Category	Percentage of incidents	Median of levels of frustration
System is slow	12	7
System froze temporarily	10	6
System crashed	10	7
Functionality not working	7	6
Disruption of the flow of activities	5	5.5
Input-device glitch	5	6.5
Connectivity issue	5	7
It is just hard	5	8
Difficulty finding things	5	4
Popups	5	6
Corrupted contents	4	7
Accidentally make a wrong selection	3	6
Unsatisfactory contents	3	4
Desired/absent functionality	3	4
Poor information presentation	3	6
Repeated login	3	7
Computer just does not work any-more	2	7.5
Uncertainty about payments	2	8
Noisy hardware	2	3
Cross-application tasks	2	5.5
Gameplay is (too) hard	2	6
Privacy concerns	2	7
Not having the right version of documents	2	8.5
Having forgotten how to do something	1	3
Running out of charge	1	9

TABLE 1.16: Categories of frustrating events in [75], their percentage among all incidents reported and their frustration level median. Frustration levels ranked from 1 (not very frustrating) to 9 (very frustrating).

Interestingly, it has been shown that computer experience might not influence frustration either [102]. Instead, frustration seems closely related to usability problems, being highly dependent on the context and the task [75].

Hertzum and Hornbæk [75] distinguish two kinds of frustrations, illustrated in Figure 1.29: the immediate frustration and the escalation of emotional outcomes. Immediate frustration is the feeling that the user experiences right after a disappointing response of the interface. It is meant to focus the user attention on the current problem to ensure that the task is completed in the end. In other words, it is adaptive and useful to the user in that it doubles down their concentration and better their chances of reaching satisfaction. On the other end, the escalation of negative emotions induced by a negative event is not beneficial to the user in the end. Hertzum and Hornbæk [75] report that frustration is correlated with feelings of fear, aggression, helplessness, anger or rage for example. Contrary to the first kind of frustration, these emotions can be discouraging and in turn, counter-productive.

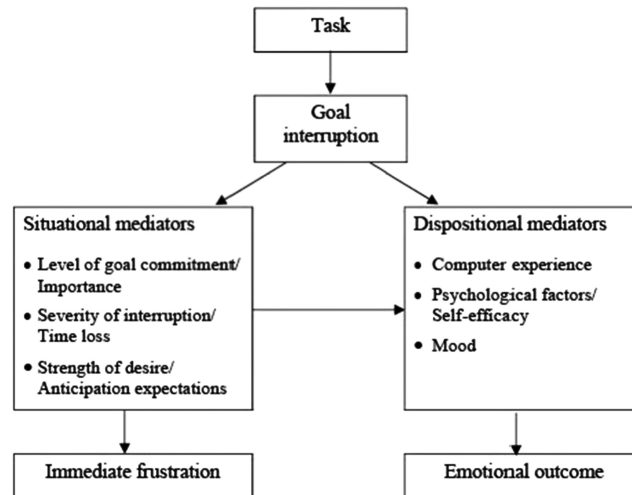


FIGURE 1.29: Computer frustration model from [75]

Frustration is not related to the functionality of a system. As Golden [64] puts it, *“it is perfectly possible to have a program which is structured, modular, readable, flexible, self-documenting, maintainable, which performs its specified function, and which is a source of constant frustration and irritation to its users”*. In other words, frustrating events are not errors in the specification violation sense. In that regard, a theoretical perfectly error-free system can still be problematic in usage.

Preference of control over accuracy

The current understanding of errors revolves in part around inaccuracy: imprecisions lead to errors. However, when it comes to user preferences, accuracy may not be the most important value.

Roy, Zhang, and Vogel [159] studied user behavior facing variable accuracy and controllability of an interface with Artificial Intelligence (AI) automation. An automated system or software is accurate when it gives out precise and exact results. Controllability on the other hand measures how effective the interface is for the user to manipulate and modify elements. The authors note that in automation in general, involving the user is usually frowned upon, it is an admission of weakness. However, when using AI to automate a task, errors and imprecision happen. In most cases, the user will either restart the software or fix the problem manually. In their experiment, the participants are using an automated AI, and are given the choice to correct it either automatically or manually. These correction tools vary in terms of accuracy (for the automation) and controllability (for manual). The authors observed that users leaned a lot on manual control, even when the available controllability was relatively low. Participants showed a strong resistance to using automatic correction to AI inaccuracies, even if it required them to spend more time on the task. The results go so far as to suggest that there is no *“natural threshold below which controllability is so low that participants stop using the manual method”*.

Another study centered around LLMs in-application assistants seem to suggest the same preference for control on the part of users. Comparing fully automated and semi-automated assistants, Khurana et al. [87] observe a preference for the latter one from participants. The semi-automated version performed better than the fully automated in various criteria like user control, software utility, and software

learnability. As the authors state, “users need the ability to intervene, make decisions, understand the process, and retain control”.

If accuracy is very valued in the current definition of errors, it could be outperformed by other system qualities in another, more user-centered definition. Namely, the ability to maintain manual control over a task appears to take precedent for users. If we were to define an error as something that does not work *for the user*, a lack of control would then maybe be more serious than an imprecise system. Errors can arise from giving control to users, but it can also encourage learning and experimenting with the system. Blaming and pointing fingers can be discouraging in that regard.

Errors can create interesting user experiences

The notions of good and bad design can be challenged as well, and with that the concept of error. Bad incidents can happen on purpose on the part of the conceptualizer, serving the user experience.

There exists normative notions of design for what makes a good interface. Affordance, clearness, immediacy are usually looked after. However, in a recent article, Cormier et al. [39] present the concept of Shitty User Experience (SUX) games; video games that violate normative design rules on purpose. For example, the player is driven to fail, has trouble interpreting or controlling the interface, as in Figure 1.30. This difficulty serves the game narrative: it is about the experience of failure. For Cormier et al. [39], these games are not poorly designed. On the contrary, they foster interesting results in players. They can be more satisfied by a hard-earned result, bond easier among themselves, collaborate better, experience joy from the absurdity of a situation or from its unpredictiveness, or feel spite – a first-hand frustration that makes them try harder (see immediate frustration in Figure 1.29). “Bad” designs can lead to valuable experience for users.



FIGURE 1.30: In the game *Stray* [173], the controls for directions are inverted when the character has a paper bag on their head (example and screenshot taken from [39])

“Good” and “bad” design is not a black and white situation, or even a scale of grey. Some defaults in a system can be valuable depending on the context. Errors, in the accepted sense of the term, can lead to positive experiences.

System interpretation errors

Interpretation errors are generally understood as coming from the user, not understanding the interface or making a mode error. However, they can also be made by

the system. These errors have received a renewed interest in recent years with the generalization of voice interfaces and gesture-based interactions.

Lam et al. [98] describe interpretation errors from the system as a failure to recognize the correct input. For examples, the detection of gesture on camera or of speech by the microphone is inaccurate. Their study show that these kind of errors are not only costly in terms of time, but affect user learning. In their experiment, participants were less accurate in a memory task when these errors would happen.

When looking to solve interpretation errors, researchers can turn to the source of the inaccuracy, developing better filters for input recognition, as Casiez, Roussel, and Vogel [33] have done with the “1€ filter”. Another solution is to strengthen the signal sent to the input device. Lam et al. [97] tested 2 ways to that end: repeating the signal, and making it more expressive through longer, more discernable commands. They have shown that participants seemed to gain time overall with these methods, even if a learning curve was present.

If we turn to user interpretation errors, they can be subject to debate in terms of blame. Was the system unclear or is it the user’s fault for not understanding or not paying enough attention? System interpretation errors seem to differ on that front. The system either takes the input the correct way, or it does not. The user is generally not blamed for not modulating their input the right way. They will however adapt to the sensitivity of the system: speaking loud and clear to a voice interface and exaggerating gestures for gesture recognition, to avoid interpretation errors [97].

If designers do not blame users in system interpretation errors, do users blame themselves? How do they experience this adaption to the system’s capabilities?

1.4.2 The edge cases of errors

In order to understand what error is in interactive systems, we’ve looked at its theoretical foundations, with the different views of error, as well as at existing models. In this thesis, we define errors from the user’s point of view, as something that does not meet their expectations. What does that entail in terms of phenomena covered by the term error? While many phenomena are universally considered to be errors – task planning errors, scoring errors or system crashes, for example – others point to the existence of grey areas. These areas are precisely what distinguishes this new definition of error from the others; it encompasses new situations. In other words, this new definition of error allows us to shed a light on other problematic phenomena.

Slips

“One afternoon in the early 1970s, I was boiling a kettle for tea. The teapot (those were the days when tea leaves went into the pot rather than teabags) was waiting open-topped on the kitchen surface. At that moment, the cat—a very noisy Burmese—turned up at the nearby kitchen door, howling to be fed. I have to confess I was slightly nervous of this cat and his needs tended to get priority. I opened a tin of cat food, dug in a spoon and dolloped a large spoonful of cat food into the teapot. I did not put tea leaves in the cat’s bowl. It was an asymmetrical behavioural spoonerism.”—Reason [148]

Sellen and Norman [164] define a slip as an action performed by automatism in an inadequate context. They also describe them as actions carried out “either in the absence of a conscious governing intention or even contrary to intentions held in the mind”.

In Booth [20], slips are referenced as “automatic errors”, as opposed to “conceptual errors”. For him, slips are the worst kind of errors because they cannot be learned from, whereas conceptual errors (mistakes, misunderstandings, misinterpretations etc.) are most common in learning phases. As Reason [148] and Sellen and Norman [164] point out, they are not specific to the use of digital tools.

They are however particularly documented and studied on digital interfaces. Examples in this case include a user that performs a keyboard shortcut in an application that does not recognize it. Or they swipe to delete a notification, out of habit, when they would have liked to read it. They can also reflexively close a dialog box, even though they knew beforehand that it was useful.

In certain modelizations, it can be difficult to pinpoint if slips are included, and if they are, how they are defined. Coming back to Payne and Altman [134] modelization from the 60’s (detailed in section 1.2.1), a human error can come from the perception, the information processing or the carrying out of the action. In the case of a reflex, it is difficult to say in what stage the error happened. Did the person recognize a wrong trigger at the perception level, or did they see everything correctly, but “activated” the wrong reflex?

Let’s consider this phenomenon in the light of Avizienis et al. [9]’s definition of error, detailed in section 1.1.2. In their system-centric view, an error is the result of a fault being activated *i.e.* made visible, and a failure is an error that prevents the system from working. In this regard, a slip is a fault, more specifically an *operational, human-made, non-malicious, non-deliberate* and *accidental* fault (see Figure 1.3). The origin (*system boundaries*), *dimension* and *persistence* of a slip will depend on the context. With these specifications, a slip is a textbook idea of a human error, putting a finger on human limitations. Humans, even when competent, well-intentioned and well-informed, are still making errors.

The rate of slips can yet be prevented with a design change, as Fitts and Jones [59] showed in 1947, studying how command design lowered the number of B-17 crashes. The problem they describe indeed fits the description of recurring slips on the part of pilots, confusing commands that look alike. What could appear as a pure human limitation is clearly contextual, situated.

According to Sellen and Norman [164], slip errors remain faithful to the context in which the action is planned and the environment in which it is played. More particularly, they take place in an *automatic control mode* of humans, fast and context-dependent, as opposed to a *conscious control mode*, slow and flexible. They can be understood in the light of Schema Theory; a slip being the activation of an inadapted schema [164]. They underline that their causes can be diverse: an slip can occur during action planning, an external element can trigger an automatic response, or the fault can lie with the counter-intuitive affordance of an object or interface. Planning actions, for them, is not something that the user does before acting but something they do while acting.

Slips are a difficult type of error to pin down, since they can be caused by interface design (causing automatism, design too close to a different context), but not always. Back, Blandford, and Curzon [11] question the ability of users to avoid slips. They investigate the avoidance of error as a cognitive skill, but their findings still point out the importance of context in the application of that skill.

Interruptions

An interruption is a cognitive disturbance of the user prompted by an event on the interface [2]. It can be triggered by an interface event of any level of importance, from “the device is about to shut down because of a low battery” to “a system notification that the user does not care for”. An interruption can also come from the communication with someone else, an email or a message in a chat application appearing on the screen. The problem of an interruption lies in the consequence of those disturbances for the user: they lose their focus, are momentarily disconnected from their current task and can experience trouble diving back in, or even forget what they were doing. Addas and Pinsonneault [3] define interruptions in the context of interactive devices as “*perceived, IT-based external events with a range of content that captures cognitive attention and breaks the continuity of an individual’s primary task activities*”. The authors also point out 3 possible particularities of interruptions that can amplify their effect:

- They are multimodal: An interruption can be visual, aural, haptic or a combination across these three.
- They can happen in parallel, because people are using several devices at the same time.
- They are reviewable: after the alert, users can repeatedly access a notification which can maintain the division of attention.

Several studies show that an interrupted task tends to have a higher completion time [93, 63, 15]. They also raise levels of annoyance and frustration in a lot of cases [2, 15].

However, depending on when the user is interrupted in their task, they experience these effects differently. Adamczyk and Bailey [2] interrupted their participants at different moments in the completion of their primary task. They observed that levels of annoyance and frustration were reported higher in moments where the user needed more cognitive resources and engagement in the task. They advocate for *attention manager systems*, systems that can evaluate of the user attention level and manage interruptions timings depending on when the estimated disturbance cost is lower.

Depending of the importance of the task, interruptions can also impact the user differently. Bailey and Konstan [15] results tend to show that the more difficult the primary task is, the more annoying the interruption is perceived. The same study shows that a task of lesser importance interrupting a more important task to the user is perceived as more stressful and frustrating.

Some studies around interruptions also account for positive effects in some cases. Addas and Pinsonneault [3] underline that it is not always relevant to see attention capability as a finite resource that can only be divided when an interruption happens. In some cases, it is best to consider it elastic. When an event is relevant to the main task at hand, it can recenter users on what they are doing. The authors call these types of interruptions *interventions*. In their taxonomy, they overall distinguish 5 types of interruptions:

- *Informational intrusions*: interruptions that bear information relevant to a user’s secondary task.

- *Actionable intrusions*: interruptions that call for an action on the part of the user (*i.e.* a response to a message) relevant to a user's secondary task.
- *System intrusions*: interruptions coming from the system itself encountering issues or limitations.
- *Informational interventions*: interruptions that bear information relevant to a user's current primary task.
- *Actionable interventions*: interruptions that call for an action on the part of the user (*i.e.* a response to a message) relevant to a user's primary task.

The article describes an experiment in which users report cases of all these kinds of interruptions. Overall, intrusions tend to lower user performance and interventions often have a positive impact.

In the studies mentioned above, timing can appear as the main factor by which an interruption is going to be more or less impactful, on the user efficiency and on their negative emotions. However, it is also possible to affect the consequences of interruptions by changing how they appear on the screen. In their study, Böhmer et al. [19] show for example that users prefer when a phone call is not displayed full screen, but allow them to shortly continue their current task while being made aware that they have an incoming call.

Phenomena that are problematic but not typically described as errors are interesting to look at because they are a hint, in practice, of where the line is, between things that are errors and things that are not. Interruptions are such phenomena. For example, "slip errors" is a common expression when talking about slips, but a quick search in the *ACM Library* [1] for articles titled with "interruption error" shows no relevant results⁵. They seem to be defined and described with different words: a problem [2], a nuisance [15], a disruption [19], a distracting event [3]...

A definition of error based on user-experience would include interruptions. This phenomenon is caused or mediated (in the case of emails and messages) by the system, and can impact users negatively in their performance and cause negative emotions.

Interaction interferences

Another example of a set of phenomena that can motivate a different definition of error are interaction interferences. They are changes in the user interface occurring immediately before the user carries out a meaningful atomic input action, but too late for them to inhibit that action—even if they perceived the change [162]. As a consequence, the action is interpreted differently than the user intended. When the interface changes, the motor action is already planned, decided and non-reversible on the user's side. Moreover, the action was meaningful at the time it was decided and planned. Even if the change in the interface is perceived, the limitations of the user's perceptual-motor system make it so that he or she cannot curb the action in what seems to him or her to be immediacy. The study by Schmid et al. [162] was the only one on the subject to my knowledge, before the 2024 article that I wrote with Sylvain Malacria and Mathieu Nancel [110].

⁵on July 22th 2025

Schmid et al. assessed the robustness of their initial description of interferences in an online survey [162] in which they enquired whether participants understood the described phenomenon, at which frequency (if ever) it happens to them, and about the frustration and consequences of its occurrences. Participants were also asked to provide their own examples of interaction interferences. Schmid et al. found both generic and example-based interferences to be somewhat uniformly spread in frequency, with more than 66 % of participants reporting it more than once per month and 17 % up to more than once per day, but consistently annoying (87 % of answers between “Moderately” and “Extremely”). Interestingly, the perception of annoyance seemed to be independent of the device or task, suggesting that interaction interferences are a general problem regardless of the context. While the consequences of interaction interferences were judged as minor in general, some were reported to be severe. The frequency, annoyance and gravity of consequences of interferences reported in the survey are represented in Figure 1.31.

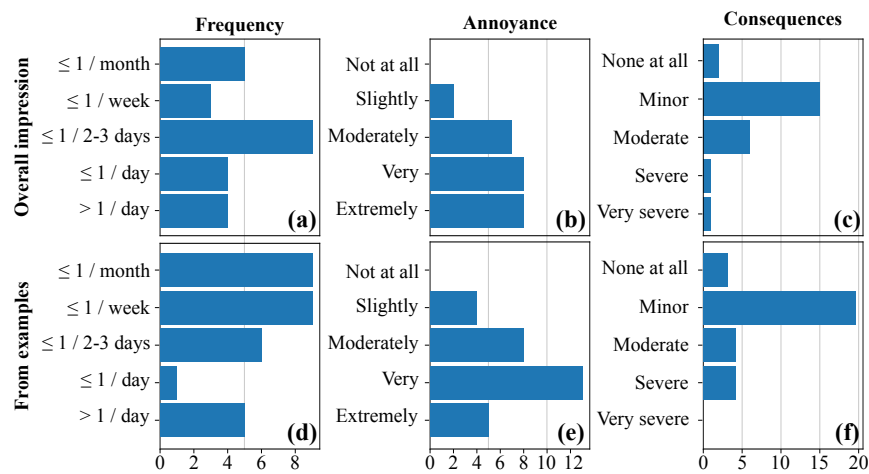


FIGURE 1.31: Interaction interferences characteristics reported by survey participants in [162]

In that same survey, Schmid et al. found that a majority of the interference examples provided by their participants occurred when selecting a target, either with a cursor or with the finger—the bulk of the rest having to do with text input. In particular, they identified examples wherein the intended target of an action either:

- moved (*e.g.* trying to click a link when the webpage unexpectedly updates),
- changed (*e.g.* the word suggestion above a gesture keyboard changes just before it is tapped),
- became occluded (*e.g.* a notification appeared above the target right before the click),
- or disappeared (*e.g.* a notification disappears right before the user clicks to close it).

Consequences can range from a harmless click, to triggering a critical operation that cannot be cancelled nor undone, *e.g.* sending an unfinished email (not all email clients have an “undo” feature), triggering a half-hour-long update right before giving a keynote, mistakenly clicking “No” (or “Yes”) when asked to save before closing an app, answering an identified scam call, etc.

Interestingly, interaction interferences do not fit in a lot of the error models presented in this section. For all of them, as we point out in [110], input errors involve a wrongly performed motor action or a wrong interpretation of the system's state by the user. Interaction interferences, while arguably a type of error, cannot be modeled by these frameworks: they are entirely unavoidable by the user in a normal course of action; occurring solely at a physiological level, they do not reflect the quality of their decisions, interpretations, or motor accuracy. They are a set of phenomena that are:

- human-induced: they happen in part because of human physiological limitations.
- not system-made: the system did not deviate from what it was designed to do, or what the designer expected.
- not human-made either: the user did no mistake in interpreting the interface, planning or carrying out their action.

In that sense, they do not fit the accepted understanding of what a human error is, because the user did nothing wrong. They can be considered as an error in the interpretation of the user action by the system, but in a way that the system is mostly unequipped for.

Chapter 2 explores interferences among a wider taxonomy of errors, for their interesting status at the crossroads between human and system error, and chapter 3 explores the design space of the solutions that can be conceived and test those solutions in a controlled experiment.

Chapter 2

The lived experience of error cost

2.1 Categorization

Before inquiring users about their lived experience with error, I wanted to make sure that they would bring up diverse situations that would cover a whole spectrum of phenomena and experiences. To that end, I elected to face them with a categorization of errors, so that they could talk about different types of negative events.

This taxonomy of error experiences originally had 10 categories (see table 2.1), that I will present in the following. Some of them are extracted from the literature, while others came up from reflexions and discussions with my advisor Mathieu Nancel and other researchers, mostly from the Loki team.

Source	Categories
Literature about error characterization	YOUR INTERPRETATION WRONG SYSTEM INTERPRETATION OMISSION ERROR IN PLANNING PROBLEM WITH THE EQUIPMENT ACTION
Discussions and practical inspiration	FAIL-UNSAFE
"Bordeline cases"	INTERRUPTION SLIP INTERACTION INTERFERENCE

TABLE 2.1: The 10 categories of errors before the workshops and their origins

This categorization covers phenomena and experiences of errors. When an event occurs, the first step is to identify it as a type of error, to determine what kind of problem has occurred. Then, this event will be characterized by the user's experience: are they experiencing significant emotional or material consequences? Are these consequences critical for them? Do they attribute responsibility for the incident to the system, themselves, or the designer?

Rather than being part of the definition of error, the assignment of blame is simply a characteristic of it. As presented in Chapter 1, assigning cause or blame is a negotiated process, for which there are different schools of thought. One of the objectives of our approach is to distance ourselves from the absolute determination of a guilty party and to characterize error as an experience, an occurrence, a phenomena happening at a certain time and place.

In the following sections I present these categories and explain their relevance. For each category, I also came up with a description of what they cover, and examples of possible occurrences (tables 2.2, 2.4 and 2.3). These elements were presented to the participants of the workshop (section 2.2). They were meant to be understood by people not necessarily familiar with design or computing vocabulary.

2.1.1 Categories extracted from the literature about error characterization

Most of our categories come from existing taxonomies and modelizations of errors. We selected them with a focus on HCI and user-experience.

YOUR INTERPRETATION and **WRONG SYSTEM INTERPRETATION** are most obviously taken from HCI literature. **YOUR INTERPRETATION** is an error in the user's interpretation of what is going on in the system, and **WRONG SYSTEM INTERPRETATION** in the system's interpretation of the user's intentions.

When representing interaction, a lot of authors will picture it as an exchange, a conversation, a loop going back and forth between the user and the system. Section 1.3.1 introduces two examples of those approaches [21, 132]. The most influential model is probably Norman's execution-evaluation cycle [129]. In all these models, an error can happen when the user interprets what is going on in the system, or when the system interprets user actions. To discriminate one from the other is not that obvious: was the command unclear or should the user have known what it was for?

We elected to name **YOUR INTERPRETATION** with a possessive to encourage participants to talk about their personal experience. We did not want them to prioritize errors that they only had seen in others because of the use of a third person.

Because error in this thesis is defined by the user perspective, the user gets to decide what events fall under **YOUR INTERPRETATION** or a **WRONG SYSTEM INTERPRETATION**. We want to investigate how participants will use these categories.

The **ERROR IN PLANNING** exists in several theories of errors, particularly Hollnagel [81]'s. To him, there are two ways user action can fail to achieve its goal. Either the plan was right, but its execution went wrong, either the plan was flawed from the beginning. This idea of having a "planification phase" in the user course of action is interesting to explore. An error in planning fits within **YOUR INTERPRETATION** and **WRONG SYSTEM INTERPRETATION**. The user can get wrong in the appropriate course of action to engage with the system. Or, one might consider that the system should allow a plan that makes sense for the user.

Outside of the phase of planning, we find errors in execution, "*erroneous actions*" as Hollnagel [81] calls them.

OMISSION errors happen when, in an action sequence, a planned action is skipped. They are also described in [81]. This kind of error is not a matter of interpretation; the user intended to do an action and did not.

One of Avizienis et al. [9]'s fault class is the System Boundary: is the fault internal (from within the system) or external (from outside the system)? The **PROBLEM WITH THE EQUIPMENT** category contains errors that originated in internal faults. The user's point of view on where the error originated is intriguing. In the case of a computer overheating in a CPU-demanding task for example, where does the error come from? Does the system fail at handling a task, or does the user misjudge the system's capacity?

Errors in the **ACTION** category happen in the physical part of an action execution. In HCI in particular, pointing errors happen in selecting a target. The user can for example overshoot and do their selection a bit too far. The system might also shift the position of the visual selector from the actual selecting point. Some **ACTIONS** are indeed **WRONG SYSTEM INTERPRETATION**: an input signal is wrongly interpreted by the system, as defined by [97, 98].

YOUR INTERPRETATION

Your interpretation of the effect of a command or action is incorrect.

- Autocorrect
- The AI prompt does not give the expected answer
- Translation software does not give you a good translation
- You click on a window in the background to give it focus, but the software interprets this as a click on a specific widget in that window.

WRONG SYSTEM INTERPRETATION

The system interprets your actions, commands, or gestures the wrong way.

- Cutting the image instead of selecting part of it because of the button icon
- The name of a command misleads you

OMISSION

Forgetting a step in the execution of a task or sequence of actions.

- Forgetting to attach a file
- Forgetting to change layers while drawing

ERROR IN PLANNING

The action plan you had in mind beforehand did not lead to the desired results; the plan was flawed.

- Actions planned in the wrong order: Unifying the color of an image before removing the background
- Plan impossible to carry out: takes too long to execute, or requires a command that does not exist on the interface
- Extra action in a plan: wasted time due to an unnecessary action (copying and pasting elements individually instead of globally, for example)

PROBLEM WITH THE EQUIPMENT

The system slows down to the point where it becomes difficult to use, or stops responding completely.

- The application currently in use crashes without warning.
- Internet failure
- Charger that doesn't actually charge

ACTION

Clicking or tapping next to the intended object: a button, a link, an email, a target, a keyboard key, etc. The click can either occur "in the void" (without consequence) or activate another object or window that is not the intended target.

- Typing error on the keyboard
- Typing next to the object you are aiming for on a touchscreen
- Overshoot with mouse

TABLE 2.2: Categories taken from the literature about error, their description (in italic) and examples (in small text)

FAIL-UNSAFE: a special kind of ERROR IN PLANNING

The **FAIL-UNSAFE** category originated in discussions with colleagues in HCI, namely Mathieu Nancel. **FAIL-UNSAFE** gathers incidents happening because a fail-safe measure was not taken. The category was inspired from a common situation: one has files on a device that they do not save somewhere else, and when the device breaks down or becomes unusable, they lose those data.

The point of **FAIL-UNSAFE** is that it is preventable. When the loss of data happens, the user can experience regret for not having planned ahead.

This category could be seen close to an **ERROR IN PLANNING**, but it does not happen in the same circumstances. An **ERROR IN PLANNING** refers to the planning of a task, where the steps to accomplish are not right. **FAIL-UNSAFE** is not on a task-scale. A fail-safe measure is not part of a broader task, it is a task in itself that the user carries out, or not.

Another category that might seem overlapping **FAIL-UNSAFE** is **OMISSION**. However, an **OMISSION** happens in the carrying out of a task. The user had to go through different steps one by one and happens to skip one. If one were to see the omission happening, they would be able to tell *when* the error happened. **FAIL-UNSAFE** differs from **OMISSION**, because there is no precise moment that one could tell that the action was missed. The problem in **FAIL-UNSAFE** is that the action was not taken before an incident happened. The user does not know when or if an incident can happen, they fail to save their data “just-in-case”.

Cases of **FAIL-UNSAFE** can be seen in HCI literature. They are not studied as a whole to my knowledge, but some articles tackle precise incidents. Zou et al. [191] for example study the reasons why people change breached passwords, or not. Other work also investigate strategies to nudge people to change a compromised password [136]. Cases of **FAIL-UNSAFE** are not seen as errors per se, but as desirable actions on the part of the user.

FAIL-UNSAFE

Forgetting or postponing a preventive action, a “just in case” measure

- Forgetting to save your document regularly
- Not making a backup
- Not preparing answers to a web form in a notepad

TABLE 2.3: **FAIL-UNSAFE** category, its description (in italic) and examples (in small text)

2.1.2 “Bordeline cases” categories

Some phenomena are not typically described as errors but are nevertheless problematic for the user. We call these “borderline cases”. Our definition of error includes phenomena that would not be typically considered errors by the system or an external observer, but that are an unwanted turn of events for the user. **INTERRUPTIONS**, **SLIPS** and **INTERACTION INTERFERENCES** are such phenomena.

An **INTERRUPTION** is a cognitive distraction caused by or through the interface that impacts the user's focus or train of thoughts. The phenomenon is presented more at length in section 1.4.2. It is an error in our definition because it is a disturbance to the user, an annoyance they would prefer not to happen.

A **SLIP**, as presented in section 1.4.2, is a reflex that is applied by automatism by the user, but in the wrong environment or context. One could argue that the phenomenon fits in the **YOUR INTERPRETATION** category, seeing it as a case of a user wrongly interpreting their environment. However, slips also happen in situations where the user knows perfectly well what is happening on the screen and what they are supposed to do. They are problematic in that unique way, that they can happen reflexively even in a context that is fully controlled and understood by the user. Slips are a matter of automatism, not interpretation. Let's note that they can also overlap, at times, with **OMISSION** errors.

A literal translation for "slip" does not exist in French, so the name of the category became "Bad reflex" ("Mauvais réflexe").

INTERACTION INTERFERENCES on the other hand are arguably **WRONG SYSTEM INTERPRETATIONS**. They are situations in which the system interprets an input on a just-changed element of the interface as any other voluntary input, as if it was possible for the user to react with a physiologically-defying immediacy (see section 1.4.2). Even if they are a sub-category within **WRONG SYSTEM INTERPRETATION**, they stand out and are worth a focus, for the simple reason that people tend not to be aware of them until they are pointed at [162]. Also, it would be interesting to see if people see them as a **WRONG SYSTEM INTERPRETATION** problem, or maybe as something different.

INTERRUPTION

Cognitive disruption during a task. You are distracted by something on the interface and forget what you were doing.

- An important email arrives while you are copying and pasting text, and you forget what you copied.
- A notification pops up on your phone while you are searching for music, and you forget the title.

SLIP

Using a familiar action in a context where it does not apply, and may even produce undesirable effects.

- Using a keyboard shortcut specific to one application in another application
- Clicking on an [OK] button reflexively without being sure what you have just confirmed
- The phone is ringing and I'm going to open the door

INTERACTION INTERFERENCE

The interface changes immediately before a click or keystroke: a pop-up appears, a list is updated, another application takes focus, etc. As a result, the click occurs "in the void" or on an unwanted object.

- Notification at the top of the phone just as you are about to click
- Mail list update

TABLE 2.4: "Borderline cases" categories, their description (in italic) and examples (in small text)

2.2 Study 1: Open-ended user consultation on the taxonomy

2.2.1 Workshops methodology

The workshops were conducted in French, from September to December 2023.

Groups

I conducted the same workshop with 4 different groups, each of 4 people. The goal was to confront the categorization to different user populations. I had a group of HCI researchers (Workshop 1), a group of computer scientists, not design-oriented (Workshop 3) and 2 groups of users whose professions were unrelated to design and technologies (Workshop 2 and Workshop 4).

The Workshop 1 was composed of 4 men, researchers in HCI, some of them also University teachers. They all had a PhD, or up. They were aged 27, 31, 44 and 45.

The Workshop 2 had 3 women and 1 non-binary person. They worked in different fields (Construction engineer, University teacher and researcher, physiotherapist and sales consultant). One of them had a bachelor degree or equivalent, two had a master or equivalent and one had a PhD or equivalent. They were aged 25, 25, 29 and 32.

The Workshop 3 featured 3 men and 1 woman, all computer scientists (one researcher, two PhD students and one developer). Three of them had a master or equivalent, and one of them had a PhD (or equivalent). They were aged 24, 25, 27 and 28.

The Workshop 4 had 4 women, working in the scientific field or in Finance and Management (administrative assistant, research contract manager, scientific mediation project manager and scientific communication and mediation manager). One of them had a 12th grade degree equivalent, and the three other had a bachelor or equivalent. They were aged 39, 39, 40 and 44.

All participants had French as a mother-tongue except for one person in the Workshop 2 who nevertheless spoke French fluently. Overall, they used laptops and smartphones several times a day, and tablets and VR sets rarely (up to once every two or three months).

Structure

Before the workshop After collecting the participants' consent to take part of the workshop, they were sent an email in preparation. The message asked them to remember a "negative event" that happened to them on a device, that they thought to be rare or original. If none came to mind, they were asked to remember the last "negative event" ("*incident négatif*") that had happened to them on a device. They were asked to write that event down somewhere in order to have it on hand at the end of the workshop.

Materials, layout and structure Each workshop lasted around 2 hours. The 4 participants sat around a table with pens, post-its, stickers, and a dozen sheets of paper size A3 at their disposal. The experimenter sat in front of them with a slideshow. They took notes while recording the sound of the discussion with a microphone. A workshop was divided in 2 phases: CATEGORIES PHASE and DISCUSSION PHASE.

Email sent

Before you come, I have a little first task for you. Can you think of a negative event that has happened to you on a computer/smartphone/tablet/... that you think is rare or original? If you can write it down somewhere, or remember it on [date], it will be useful at the end of the workshop. If no example comes to mind, simply take the last negative event you experienced with a digital tool.

TABLE 2.5: Instructions sent by email to the participants before the workshops (translated from French)

Instructions

Hello everyone, thank you for agreeing to take part in this workshop. Today we're going to work together on the negative experiences you can have with digital objects, whether it's your computer, your tablet or your phone.

Our aim is to validate and refine a categorization of these incidents, and above all to see if these categories speak to you.

In this workshop, I am going to present you with different types of situations, one for each of the large sheets we have here.

For each one, you'll be able, as a group, to associate examples of incidents that have happened to you, which you'll write on the post-its here. So for each situation I am going to present to you, the aim is for you to find all the examples you can think of, write them on the post-its and arrange them on a large sheet. The only condition is that you understand the examples proposed by the others, and that you agree amongst yourselves that an incident does indeed belong to the current category. For each situation, you'll have 5 minutes to find all the examples you can think of. I'll have a timer that goes off after 5 minutes, which normally means that the discussion stops. Of course, if at that point you're in the middle of a sentence or writing on a post-it note, you can finish what you're doing. The idea is to take about 5 minutes each time, otherwise it'll take more than 2 hours.

We'll discuss 10 situations, and then I'll ask you a few questions.

You can move or duplicate post-its as you like in the second part of the workshop. If you think of an example that does not fit into the situation we're discussing, you can write it down on a post-it and put it aside for later use. You'll have a chance to come back to all the sheets at the end of the workshop.

This task should take about 2 hours.

Is that clear to everyone? Normally, by doing this should clear up the exercise.

TABLE 2.6: Instructions given to the participants at the beginning of each workshop (translated from French)

CATEGORIES PHASE During the CATEGORIES PHASE, the participants worked on the CATEGORIES one by one in the order given in table 2.7. All groups followed the same order of CATEGORIES, because it would not have been possible to balance the order of all 10 over 4 workshops. Also, some CATEGORIES were presented before others strategically, to ease the understanding of the participants:

- **WRONG SYSTEM INTERPRETATION** was before **INTERACTION INTERFERENCE**. While an interaction interference is by definition a problem of interpretation from the system – the system interprets an out-of-human perceptuo-motor range input as voluntary, people tend to not be aware of this problem before it is explained to them. Only then do they generally look back and recognize its occurrence. We worried that, if we gave the definition for **INTERACTION INTERFERENCE** before facing participants with **WRONG SYSTEM INTERPRETATION**, that interaction interferences would take over other kinds of incidents in the latter.
- **INTERRUPTION** was right before **INTERACTION INTERFERENCE**. Since both phenomena usually feature popup windows and notifications, we did not want participants to confuse them. By presenting them one after the other, we made sure that their memory of the first definition was recent. We chose to introduce first the **CATEGORY** relating to a cognitive distraction, and then the perceptuo-motor problem.
- **FAIL-UNSAFE** was before **OMISSION**. These two **CATEGORIES** overlap each other, in the sense that some cases of omission can also be seen as a lack of fail-safe measures. To make sure participants would give examples of **OMISSION** that exclude **FAIL-UNSAFE**, we put the broader **CATEGORY** after.

Order of the CATEGORIES

1. **PROBLEM WITH THE EQUIPMENT**
2. **ERROR IN PLANNING**
3. **YOUR INTERPRETATION**
4. **WRONG SYSTEM INTERPRETATION**
5. **INTERRUPTION**
6. **INTERACTION INTERFERENCE**
7. **ACTION**
8. **SLIP**
9. **FAIL-UNSAFE**
10. **OMISSION**

TABLE 2.7: Order in which the workshop participants went through the CATEGORIES

The participants were first presented with a **CATEGORY** with its name, definition and examples. These elements are presented in tables 2.2, 2.4 and 2.3. The name and definition were written down on a slide and available to them throughout all the steps for this **CATEGORY**. The examples were just given orally, as to help them without adding too much bias. The participants wrote down the name of the **CATEGORY** on a big sheet of paper to give it a title. That sheet of paper became a *Category Sheet*.

Then, the experimenter put 5 minutes on a timer. The participants were asked to come up with as many real situations and examples fitting to that **CATEGORY** as possible. They had to agree about the *Incidents_W*, write them down on post-its and stick them onto the *Category Sheet*. If they happened to disagree on an *Incident_W*

belonging to a CATEGORY, the experimenter would then give them the possibility to add a colored sticker to a given post-it to flag it as controversial or “unsure”. When the timer went off, the participants had time to finish their current post-it before moving on.

Finally, with the *Category Sheet* completed with their post-its of *Incidents_W*, the participants were asked: “With the examples before you, would you change the name of the category? Its description? Why?”. They could take notes on the *Category Sheet* itself if they wanted to. This step could sometimes be very short (if the participants agreed with the CATEGORY’s name), and sometimes fostered deeper discussions. For the workshop not to go over 2 hours, the experimenter would sometimes invite the participants to shorten their exchange. They then moved on to the next CATEGORY.

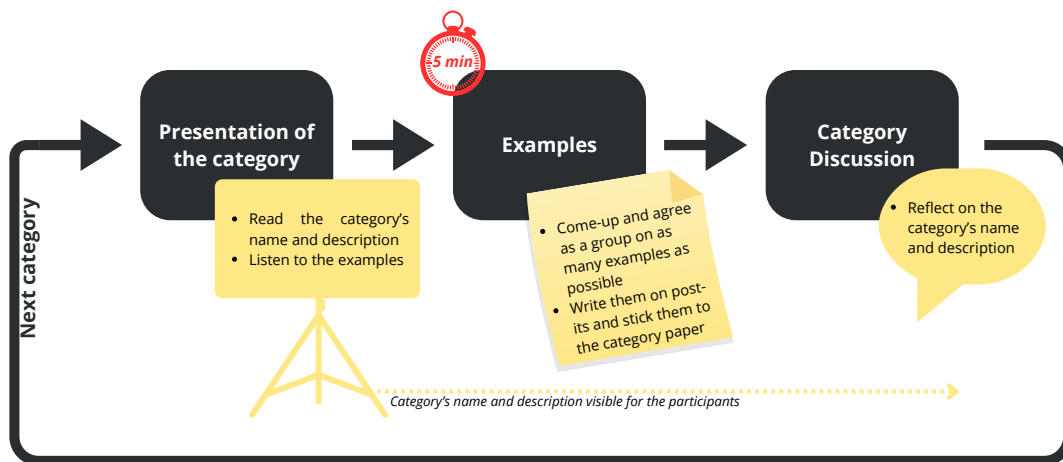


FIGURE 2.1: The steps of the CATEGORIES PHASE of the workshop. On the yellow canvas are written the tasks given to the participants at each step.

When the participants had reviewed all 10 CATEGORIES in this manner, the DISCUSSION PHASE began.

DISCUSSION PHASE For the second phase of the workshop, the participants put all the *Category Sheets* on the table. They answered together questions from the experimenter, given in table 2.8. All questions were asked in the same order in each workshop, but the experimenter allowed for some flexibility if participants organically took them in a different order.

In this phase, they were encouraged to add and move post-its around and to modify the *Category Sheets* by adding notes. These changes were documented: the participants added different stickers on the post-its to signify their status. One color for moved post-its, one color for added ones, and one color for the ones they were unsure about (that we will refer to as marked “unsure”). They also stated out loud their actions so that we could document it from the audio recording.

After the workshop After the workshop, each participant was given a unique id character string. In the following hours, they received an email from the experimenter asking them to fill out a survey, in which they entered this character string

Questions of the DISCUSSION PHASE

- Are there any post-its you'd like to transfer from one sheet to another? Why?
- Are there any sheets you'd like to put together? Why?
- Is there a sheet you'd like to split into 2 (or more)? Why?
- Would you like to add another category? Delete one?
- Can you think of any mistakes that do not fit into any of the categories discussed today?
- Take the incident you were asked to come in with today. Does it fit into one of the categories we have?
- Can you think of incidents or types of errors that we did not talk about today?

TABLE 2.8: Questions given to the workshop participants during the DISCUSSION PHASE (translated from French)

(see table 2.9). This id allowed us to link the informations given in the survey with the other data about each participant, without identifying them in our database.

Survey questions	Response type
Enter here the code given to you by the experimenter by e-mail	Free text
What is your gender identity?	Single choice: Man, Woman, Non-Binary, Does not want to answer, Prefer to self-define (Free text)
For what do you use digital tools most?	Multiple choice: Communicating (emails, messages), Using a browser, Using specialized software, Programming, Reading, Watching video content, Watching photo/image content, Social networks, Video games, Other (Free text)
What is your age?	Free number entry
What is/are your mother tongue(s)? (German, Korean, Spanish, French, Japanese, Mandarin, Italian...)	Free text
What field do you work in?	Single choice: Agronomy and food, Audiovisual and entertainment, Business, Handicrafts, Construction, Culture, Education, Finance and management, Law, Politics, Scientific, Health, Safety, Sport, Technology, Transport and logistics, Other (Free text)
What's your main job/activity?	Free text
What is your level of education?	Level 1 (compulsory schooling up to age 16), Level 2 (maximum 1-year training after secondary school), Level 3 (CAP, BEP, BTEC National Diploma), Level 4 (Baccalauréat, High-School Diploma), Level 5 (DEUG, BTS, DUT, DEUST, 12th grade, Two-year university degree in technology), Level 6 (Licence, Licence LMD, licence professionnelle, Maîtrise, Bachelor's degree), Level 7 (Master, DEA, DESS, diplôme d'ingénieur, Master's degree), Level 8 (Doctorat, habilitation à diriger des recherches, PhD and up), Other (Free text)

How often do you use these digital devices? PC, Laptop, Tablet, Smartphone and Virtual reality system	For each device, single choice: Up to once a month, Up to once a week, Up to once every two/three days, Up to once a day, Several times a day
---	---

TABLE 2.9: Questions for each participant after the workshop. The American equivalents of the French diplomas are given in italic

Data and material for analysis

The workshops produced different materials that were rich in information. This section presents these materials and how they were formatted to allow for their analysis.

Post-its and paper sheets

First, we had the *Category Sheets* created by each group. On some of them, the participants had added notes about the CATEGORY's name. Furthermore, during the CATEGORIES PHASE, a lot of remarks were done orally by the participants. Thanks to the experimenter's notes and the audio recording, we were able to list these written and oral comments on the CATEGORIES. The remarks are of 6 types, regarding:

- A CATEGORY's name,
- A CATEGORY's description,
- Splitting a CATEGORY in 2 or more,
- Uniting 2 or more CATEGORIES,
- Removing a CATEGORY and
- Creating a CATEGORY.

On each *Category Sheet* were stuck post-its with *Incidents_W*. Certain post-its also beared stickers with different meanings. Based on the experimenter's notes, backed by the audio recordings, we were able to add further specifications relating to the *Incidents_W*.

Notes taken by the experimenter

The experimenter took notes for the entire duration of each workshop. Their comprehensiveness in this matter was limited, as they were directing participants at the same time and running the workshops.

However, having these notes allowed for an early broad view of each workshop, along with some of their salient points. The experimenter was able to write a report immediately after each workshop (see appendix B). It later inspired the thematic analysis.

Audio recording

The workshops were taped for their entire duration with a microphone placed in the middle of the participants' table. These recordings were turned into a transcript with the help of Whisper.

Informations on each *Incident_W*

- Workshop number
- CATEGORY
- Written on a post-it / given orally only
- Usure about or controversial (noted “unsure” in the following)
- Taken from the given examples of the CATEGORY description
- During the CATEGORIES PHASE, placed on a previous *Category Sheet*
- During the CATEGORIES PHASE, put aside and placed on a *Category Sheet*
- During the CATEGORIES PHASE, displaced from one *Category Sheet* to another
- During the DISCUSSION PHASE, displaced from one *Category Sheet* to another
- If the post-it was displaced from one sheet to another, initial and final CATEGORIES
- During the DISCUSSION PHASE, placed on a *Category Sheet*

TABLE 2.10: Data drawn from the *Incidents_W* given by participants during the workshops.

The software was very successful at timing the vast majority of the speech and differentiating the different speakers. However, the transcript itself was mostly not reliable. The AI failed to interpret some words correctly, especially anglicisms and English words incorporated in French sentences. French people tend to use anglicisms, especially when talking about technology. Two of the groups are also trained in computer science, and therefore have a higher tendency to use English words as part of their professional vocabulary. It was also not able to detect and transcribe the speech of participants talking over each other, which could happen a lot during discussions. Speech in low voice was oftentimes missed, maybe partly because the microphone was placed in the middle of the speakers, a bit away from each of them. It seemed finally to miss a lot of negative part of sentences, which reversed the meaning of statements.

For these reasons, the experimenter read and corrected the entirety of Whisper’s output by hand, to obtain the final transcript in a spreadsheet.

The spreadsheet’s lines followed the change of speakers. Other informations were added throughout the transcription process, such as the phase (CATEGORIES PHASE or DISCUSSION PHASE) in which the speech took place, the *Category Sheet* put out, the *Incident_W* described and the CATEGORIES commented. In the following, citations from the transcript are reported with the participant and workshop numbers (*i.e.* P11_{W3}).

This first experience using Whisper allowed me to see its strengths and weaknesses. It seemed to lack precision when based on a recording with overlapping speech and anglicisms or mixed languages. I would however use it again in another setting, for example an interview, where speakers do not typically talk over each other. I would also suggest it for situations where people speak English, or where the conversation topic does not encourage the use of anglicisms.

2.2.2 Results on the categorization

The first goal of the workshops was to provide feedbacks to the categorization of errors. Did participants understand each category? Did it represent their experience? What changes did they propose?

Feedback on the categories

Participants had remarks regarding the categories, that they mostly expressed in the DISCUSSION PHASES.

As an indication of how much the participants related to the categories, this section includes the number of *Incidents_W*⁶ that were given for each of them, over the 4 workshops. We only counted the *Incidents_W* that we agreed with in terms of category, and report the other ones, when any. The number of *Incidents_W* that were initially assigned to a category, and moved to another one later on (either later in the CATEGORIES PHASE or in the DISCUSSION PHASE) are also given each time as an expression of the clarity of the category for participants.

PROBLEM WITH THE EQUIPMENT Most comments about this category aimed to distinguish different concept of “equipment” *e.g.* hardware, software, network etc. Interestingly, this line was not placed the same for every group. Workshop 1 drew the line between hardware and software. Workshop 2, on the other hand, delimited the equipment to the tool they have in their hands; the internet connection for instance is not a part of it, even if a problem stemmed from hardware. Workshop 4 differentiated “system” from “equipment” in the same way. Workshop 3 and Workshop 4 underlined the problem of latency, wanting to, respectively, include it to the title of the category, or single it out as a different category.

The *Incidents_W* given show that the participants understood what **PROBLEM WITH THE EQUIPMENT** entails. In fact, this category gathered the most examples out of all of them (49 *Incidents_W*). One of them was created for **ERROR IN PLANNING**, another one for **FAIL-UNSAFE**, and both were later reassigned as **PROBLEM WITH THE EQUIPMENT**. One post-it was marked as “unsure” over the 4 workshops and is not included in that count: “*Poor calibration of the touchscreen (the one in the subway)*”^{*} [W2]. Looking at the categorization, this example fits instead better into the **ACTION** category, and was counted as such.

Since **PROBLEM WITH THE EQUIPMENT** was the first category presented to participants, it may in part be due to an order effect but it shows nonetheless that the category mirrored some of their past experiences. Examples revolved around clear hardware crashes, but also software slowness or freezes: “*there was a CPU failure*”^{*} [P4_{W1}], “*laptops whose batteries burn out and become basically desktop computers*”^{*} [P5_{W2}] or “*the page is charging*”^{*} [P14_{W4}].

ERROR IN PLANNING This category was less popular among participants. Workshop 1 defined it as the difference between theory and practice. Workshop 2 wanted to add the fact that the user is at fault in its description. The remarks of Workshop 3 summarized well the overall feeling: that planning is not always, if not ever, a part of everyday interacting. They mentioned that they act without a plan, figuring the next steps as they go.

⁶The ^W indicates that these examples were produced in the workshops. This marks the distinction with *Sub-categories_Q*, presented to the participants of the questionnaire we report in section 2.3 below.

Participants gave only 6 correct *Incidents_W* for this category, and 18 incorrect ones. The *Incidents_W* given here were mostly about uncertainty in front of a tool that the user is not familiar with, not about planification itself (typically, “*Change the color of an image, but the filter affects the background because I don’t use/know the software*”^{*} [W2]). Examples about programming were an exception; they are in fact instances of **ERRORS IN PLANNING**, where the user sets out a algorithm that can fail in the follow-through, with P1_{W1} saying “*You write some code. You expect a certain effect. But then you don’t get the desired effect. [...] you’ve planned a sequence of actions*”^{*} or P10_{W3} “*in terms of code, we’re going to code something a little brutally, whereas if we had taken more time and had a slightly more general approach, we would have saved time later*”^{*}.

YOUR INTERPRETATION/WRONG USER INTERPRETATION Most of the comments on that category pointed out the underlying problem of communication in a **WRONG USER INTERPRETATION**. Workshop 1 and Workshop 3 took issue with the “your” in the name because they felt it was blaming the user. Because of this, **YOUR INTERPRETATION** became **WRONG USER INTERPRETATION**.

Participants gave 26 correct *Incidents_W* in total for this category. One of those examples was originally created for **ACTION**, and then moved in this category. One example was incorrect, as it described an **OMISSION**. Examples revolved around difficulties interpreting the interface, be it an application layout, a command symbol (“*icon unclear*”^{*} [P16_{W4}]) or text displayed (“*the message is not sufficiently simplified*”^{*} [P6_{W2}]).

Workshop 1 and Workshop 3 proposed to single out “bad designs” as a separate category, phrasing it as “clarity of the interface” and “bad design choices”. Two out of 4 workshops brought this point up, we elected however not to create categories based on the blame direction. Workshop 3 proposed 2 other new categories: “Unfamiliarity with the tool” and “Refusing to interpret”. The first one is a possible cause of a **WRONG USER INTERPRETATION**, while the second one is an interesting observation. Sometimes, as the participants reported, they choose to put their interpretation on hold: “*moral inability to take the time to determine which one does what exactly*”^{*} [P9_{W3}]⁷. This was a minority opinion, however, so we elected not to implement that change in our categories.

WRONG SYSTEM INTERPRETATION Again with this category, participants comments revolved around understanding problems between the system and the user. Modifications to the name of the category were also made, that only apply in French⁸, clarifying it as “wrong interpretation by the system”.

This category had the second highest number of *Incidents_W* in total (44), one of which participants marked as “unsure”. An example was originally created for **PROBLEM WITH THE EQUIPMENT**, another one for **INTERACTION INTERFERENCE** and both were later reassigned as **WRONG SYSTEM INTERPRETATION**. Examples involved mostly surprises at the device’s output after an action: “*you do things, but the focus isn’t actually on the right window*”^{*} [P3_{W1}] or “*automatic spell checker*”^{*} [P14_{W4}]. Some of them mentioned LLMs not giving out satisfactory outputs, like P9_{W3}: “*I asked ChatGPT to [...] give me the answer to a partial integration, and it did give me an answer to a partial integration, but the answer was completely wrong. [...] I asked it to do*

⁷Quotations with an asterisk were translated from French

⁸In French, the category was named “Interprétation du système” and was renamed “Interprétation par le système” to remove any ambiguity about the source of the interpretation (in this case, the system and not the user).

something it doesn't know how to do, but instead of telling me it doesn't know how to do it, it just did something stupid."^{*}.

INTERRUPTION While the name of the category comes from the literature, participants thought it lacked precision (including the group of HCI researchers). All workshops proposed different options, including "disruption", "distraction", "loss of focus" and "diverted attention".

Among all the workshops, participants gave 29 *Incidents_W* of **INTERRUPTIONS**. One *Incident_W* was not included in this count, as it described an **OMISSION**, that happened to disrupt the user as a consequence (*"a demo in a conference, and I just forgot to type cd in the terminal, so some commands didn't work anymore"* [P3_{W1}]). No examples were given in this category to be corrected later as belonging to another one. Participants gave examples of textbook interruptions, with popups or notifications disturbing their focus, as in *"you're typing [on your laptop], you have your cell phone next to you, and you're receiving messages on your cell phone"* [P6_{W2}] or *"You get a little pop-up, you get a message. [...] it says new message, I look and then that's it, it's over"* [P12_{W3}]. All workshops gave examples of "external" interruptions, i.e. interruptions coming from another device than the one they are currently using. Some examples were surprising and interesting. One participant mentioned a problem coming from the weakness of an interruption: they would have liked to be disrupted but the interface event failed to do so (*"You saw the thing, you made a mental note to go back, but then you forgot, so it's more like the distraction didn't work, it failed to distract you"* [P3_{W1}]). Two other people talked about self-interrupting; 1- switching task while waiting for something to load, and then coming back to realize they forgot part of what they were doing and 2- getting off-track during a web search.

INTERACTION INTERFERENCE In the same way as **INTERRUPTION**, participants proposed alternatives for the name of **INTERACTION INTERFERENCE**. "Asynchrony", "Latency" (of the user), "irruption", "motor" (error), "Untimeliness"⁹ and "interruption" were proposed as a category name among the workshops.

This category produced the lowest number of *Incidents_W* from participants (18). One of those examples was given for **WRONG SYSTEM INTERPRETATION** originally and then moved in this category. One post-it from Workshop 4 was marked as "unsure", and appears to describe a **SLIP** (hence, did not make the count): *"clicking on a chat popup instinctively"*. Another example from the same workshop seemed to misinterpret the phenomenon of interaction interferences. It described a spreadsheet software interpreting a click on a specific cell as a change of focus, going *"to the first field and not the one you clicked on"* [P16_{W4}], arguably a **WRONG SYSTEM INTERPRETATION**. All workshops however understood the category and gave examples of updates- (*"you start typing [...] a drop-down menu appears and you click on [it], but in fact it hadn't finished loading"* [P16_{W4}]) and popup-type interferences (*"the pop-up appears above where the time and other shortcuts are, and sometimes I want to click on it, and sure enough, the pop-up appears, and I click on it, nothing happens, and I get annoyed"* [P12_{W3}]). Workshop 1 and Workshop 4 described interaction interferences in the **WRONG SYSTEM INTERPRETATION** category, backing up the assignment of **INTERACTION INTERFERENCE** as a sub-category of **WRONG SYSTEM INTERPRETATION** (see section 2.4). Two examples referred to dark patterns most likely, where other categories feature at most one dark pattern example. They described for instance the move at the last second of the cross to close an ad. Another example

⁹"Intempestivité" in French.

brought up interaction interferences in games, where a player can shoot their ally on accident.

ACTION/AIMED ACTION The name of this category did not satisfy participants. They either completed it (“user”, “missed”, “bad”, “unaimed”, “unexpected”, “erroneous”) or proposed another name altogether: “pointing error”, “precision error”, “user input” or “distracted user”. Because we did not want to include a blame in the title, we elected to rename that category **AIMED ACTION**. They mostly blamed the user in this category, which also shows in the examples: *“you pressed a little too long, so you’re selecting instead of scrolling”** [P5_{W2}] or *“when you’re playing a video game and you want to click on something, but you click on the wrong enemy”** [P10_{W3}].

They indeed mostly concerned pointing or calibration errors. Some examples, according to us, ultimately fitted in the **WRONG USER INTERPRETATION** or **SLIP** categories. 18 correct *Incidents_W* were brought up among all the workshops. One of them was created for **WRONG SYSTEM INTERPRETATION** and later moved to **ACTION**. The participants marked 1 example as “unsure”. 5 examples were given that fitted the definition of **SLIPS**.

SLIP Participants mostly agreed with this category’s name. Because the name in French could be translated to “bad reflex”, some participants raised issues at blaming the user. They proposed “bad habit” or “unefficient contextual action” as alternatives.

23 *Incidents_W* were brought up among all the workshops. One of them was created for **WRONG USER INTERPRETATION** and later reassigned as **SLIP** and 2 of them came from **AIMED ACTION** in the same way. P4_{W1} reported for instance the case of *“Control S in the terminal, especially when you’re just starting out in your first year of college, you don’t know that it freezes your terminal, and everyone does it”**, and P5_{W2} explained being caught in a flow while using social media: *“I scroll, scroll, scroll. And suddenly, I see something flash by. Oh no! Wait, no, can I see it, please?”** No example was later moved to another category, and none of them was marked “unsure”. Examples given by the participants fitted the definition of the phenomenon. A lot of them mentioned keyboard shortcuts, controllers and keyboard habits.

FAIL-UNSAFE Since we came up with this category, it was particularly interesting to have participants’ feedback on its name and description. They proposed “Prevention”, “Laziness¹⁰”, “lack of anticipation”.

Among all the workshops, participants offered 28 *Incidents_W* of **FAIL-UNSAFE** acts. No example was marked “unsure”. The examples given showed that the category was relevant to them. They talked about postponing backups, system updates or having those fail-safe measures not work or not be enough, such as *“I think I back up [my pictures] on... On nothing at all, which is not good because I would lose a lot of stuff.”** [P14_{W4}].

OMISSION Most workshops completed the name of the category, either with the actor (“Omission from the user”) or the object (“Omission of a step”). The Workshop 2 in particular made comments about **OMISSION** overlapping other categories, namely **FAIL-UNSAFE** (omission to take a fail-safe measure), **ERROR IN PLANNING** (omission of a step in the planning of an action) and **INTERRUPTION** (omission caused by a distraction).

¹⁰more specifically “flemme”

All workshops recognized however situations that only fit in **OMISSION**: forgetting to save a document, forgetting to put a link in an email or to do a “git pull” before committing changes on git. P13_{W4} for instance reported forgetting to “*add a courtesy phrase sometimes. You’re in such a hurry to send your stuff that... Well, it depends on who you’re writing to, you know. If you write to me, I won’t take it personally, but if you write to a service provider...*”^{*} This category produced 24 correct *Incidents_W* among the 4 workshops. None were marked “unsure”, and no other example was later reassigned as another category. One post-it was first created as an **INTERACTION INTERFERENCE** and then moved to the **OMISSION** *Category Sheet* at the **DISCUSSION PHASE** during Workshop 3: “*Scroll within a scroll, scroll the wrong window*”^{*}. This would appear to be a **WRONG USER INTERPRETATION** or a **WRONG SYSTEM INTERPRETATION**, as the person does not foresee what the consequences of their action will be. It was hence not included in the category count. We note however, that participants possibly could have meant that they *forgot* that the system behaved this way, and failed to adapt.

Comparing the workshops through *Incidents_W*

Another objective of the workshops was to collect examples from different users. The aim was twofold: first check the participants’ understanding of the categories through the examples they gave, and second to provide examples for the second study (described later in section 2.3). However, these *Incidents_W* also gave informations on the different groups, in their number, and in the type of situations that were shared during the workshops.

CATEGORY	Workshop	Proportion of correct examples in category (nb)	Nature of examples
PROBLEM WITH THE EQUIPMENT	1 (HCI experts)	39% (19)	hardware failures, general “things not working” because of a crash or a freeze. One post-it: functionality that, according to the participant, should be present but is not (bluetooth that is not reconnecting automatically).
PROBLEM WITH THE EQUIPMENT	2 (baseline)	24% (12)	latency, internet connection difficulties, hardware failures. design problems, for instance a website that does not have a smartphone mode.
PROBLEM WITH THE EQUIPMENT	3 (IT professionals)	12% (6)	hardware failures or surprising behaviors, for instance a laptop turning off and on while charging.
PROBLEM WITH THE EQUIPMENT	4 (baseline)	24% (12)	latency, hardware failure, unexplained behaviors (e.g. a smartphone activating a security lock “on its own”). One post-it: not an error, but a reaction to an error: the user spamming clicks when the interface is not responding.
ERROR IN PLANNING	1 (HCI experts)	67% (4)	programming, incorrect sequencing of actions.

ERROR IN PLANNING	2 (baseline)	0% (0)	difficulties in knowing how to perform a task, not how to sequence actions. 1 post-it did not describe an event, but the emotional consequences of these difficulties (guilt before and during the interaction).
ERROR IN PLANNING	3 (IT professionals)	33% (2)	codes not enough thought through, not knowing the procedure to perform a task. One post-it: an action made difficult by the non-existence of a command (going back after following a link to a different part of a pdf).
ERROR IN PLANNING	4 (baseline)	0% (0)	not having a plan to get to a given result, not being familiar enough with a software to perform a task.
WRONG USER INTERPRETATION	1 (HCI experts)	35% (9)	unclear interfaces, misinterpretation of the system written language, interface standards.
WRONG USER INTERPRETATION	2 (baseline)	27% (7)	user misinterpreting symbols, system messages being inadequate (too specialized or counterintuitive).
WRONG USER INTERPRETATION	3 (IT professionals)	19% (5)	confusing similar-looking commands, finding a precise command in a software, interfaces straying from standards.
WRONG USER INTERPRETATION	4 (baseline)	19% (5)	commands being not clear, in terms of what they do and to what extent (e.g. ctrl+A not always selecting the whole page).
WRONG SYSTEM INTERPRETATION	1 (HCI experts)	32% (14)	interaction conflicts <i>i.e.</i> , the same input activates different commands depending on the system's state, creating mode errors. interface layout, formatting, false positives in vocal command. interaction interferences (see 2.1.2).
WRONG SYSTEM INTERPRETATION	2 (baseline)	30% (13)	autocorrect performing erroneous corrections, translation software giving out wrong outputs, LLMs not giving a satisfactory response to their prompt. vocal recognition, GPS location and steps recognition.
WRONG SYSTEM INTERPRETATION	3 (IT professionals)	20% (9)	several cases of pointing not detected by the system or not recognizing the right intention. voice recognition errors, shortcomings of LLMs ("forgetting" previous prompts and not being proficient to do mathematics).
WRONG SYSTEM INTERPRETATION	4 (baseline)	18% (8)	autocorrect, translation softwares, intention recognition in pointing, LLMs, mode error. One post-it: interaction interference (see 2.1.2).
INTERRUPTION	1 (HCI experts)	28% (8)	instances of varied importances: phone calls, emails, notifications, popups prompting system updates.
INTERRUPTION	2 (baseline)	31% (9)	social media, popups, hardware warnings (about the battery, the RAM or system update), "external interruptions" coming from another activity (driving).

INTERRUPTION	3 (IT professionals)	28% (8)	instances of varied importances: emails, hardware notifications (low battery), or useless messages that they tried removing without success.
INTERRUPTION	4 (baseline)	14% (4)	advertisement, emails, and critical popups (with countdown).
INTERACTION INTERFERENCE	1 (HCI experts)	17% (3) ¹¹	update- and popup-type interaction interferences.
INTERACTION INTERFERENCE	2 (baseline)	28% (5)	popup- and update-type events. One potentially critical example: as a medical professional, selecting the wrong patient on a list, which design does not prevent the phenomenon.
INTERACTION INTERFERENCE	3 (IT professionals)	33% (6)	mostly about latency, as an element of the interface appears or gets updated with a delay. elements disappearing right before their click.
INTERACTION INTERFERENCE	4 (baseline)	22% (4)	popup- and update-type. 2 SLIPS (clicking on something out of habit in the wrong context) 1 WRONG SYSTEM INTERPRETATION (a click meant to select something within a window just performs a focus switch to that window). trouble discriminating between INTERRUPTIONS and INTERACTION INTERFERENCES , as they associated it with the same objects (popups and notifications).
AIMED ACTION	1 (HCI experts)	44% (8)	pointing/typing errors due to the proximity of commands, with a trackpad and on touchscreens. One SLIP , confusion of keyboard shortcuts.
AIMED ACTION	2 (baseline)	33% (6)	pointing errors, one of which in a game setting (wrong aiming), scrolling at the wrong place with imbricated scrolls. One SLIP : swiping in the wrong direction out of habit on a meeting application.
AIMED ACTION	3 (IT professionals)	17% (3)	fat finger problem, aiming in a video game, measuring of scroll distance (going too far).
AIMED ACTION	4 (baseline)	6% (1)	trouble using a touchscreen with cold or wet fingers. 2 SLIPS , describing a change of keyboard causing errors out of typing habits.
SLIP	1 (HCI experts)	26% (6)	typing errors when changing keyboard layout (AZERTY to QWERTY), confusing keyboard shortcuts of different Operating Systems (OSs) or softwares.

¹¹The group was very active in the activity, which the numbers from the other categories bring to evidence. However, the workshop was taking longer than expected as a consequence, so the experimenter elected to cut short the examples section for **INTERACTION INTERFERENCES**. The group was composed of researchers in HCI, and was familiar with the phenomenon in that regard. They could thus discuss the name and the description of the category knowingly, even with spending less time thinking about examples.

SLIP	2 (baseline)	30% (7)	keyboard shortcuts, cleaning out notifications or popups too fast, and confusing layouts of controllers. Two examples described not slips, but a spike in attention, potentially causing stress, due to a certain trigger (a music one uses as a ring-tone or notifications sounds as pranks).
SLIP	3 (IT professionals)	30% (7)	typographical errors due to a change of keyboard, deleting messages “too fast” as a reflex of deleting the useless ones (emails or notifications), slips due to habits in layout (the location of applications).
SLIP	4 (baseline)	13% (3)	keyboard shortcuts changing from one software to another, accepting conditions out of habit, forgetting to put the attached file to an email.
FAIL-UNSAFE	1 (HCI experts)	25% (7)	data backups, updates, preventing the shortcoming of an interface (writing on a text editor before copy-pasting to a web page), not thinking ahead on the battery, risks taken while in administrator mode (“rm *” in the terminal).
FAIL-UNSAFE	2 (baseline)	36% (10)	not doing a backup on a different device, not saving a document or a game, not plugging a charger before an update or postponing updates until the very last, likely inconvenient, moment. case of the FAIL-UNSAFE not being sufficient.
FAIL-UNSAFE	3 (IT professionals)	14% (4)	postponing updates, sometimes because they did not want to have to adapt to changes in the interface before a work deadline. errors while setting up a fail-safe measure, namely deleting the distant repository using git.
FAIL-UNSAFE	4 (baseline)	25% (7)	expecting the system to save their documents automatically, errors occurring while setting up or using a fail-safe measure (e.g. the backup failed to load).
OMISSION	1 (HCI experts)	33% (8)	missing steps while using git in command line, skipping steps in other contexts (updating a selection before doing an action, or including the attached file to an email for example).
OMISSION	2 (baseline)	13% (3)	forgetting to set up their alarm, to switch layer while drawing on a tablet, to put the attached file to an email.
OMISSION	3 (IT professionals)	33% (8)	various examples: skipping a step in an action sequence in different contexts (copy-paste a link in an email, switching layer while editing an image, saving at the end of an action), forgetting to perform a single action (delete an alarm). One example: compiling without saving the changes in the code.

OMISSION	4 (baseline)	21% (5)	forgetting to save after an action, forgetting to attach a file, add the object or a formal salutation to an email.
-----------------	-----------------	---------	---

TABLE 2.11: Number and nature of correct *Incidents_W* given by CATEGORY and by workshop. Percentages represent the proportion of examples in the category in question

In the formulation of examples (see table 2.11), we can separate workshops in two groups: 1- Workshop 1 and Workshop 3, involving HCI researchers and computer scientists and 2- Workshop 2 and Workshop 4, where participants were regular devices users without being IT professionals.

The group of HCI researchers produced the highest amount of examples overall. This is perhaps not surprising, given that these individuals had the most expertise on interface issues. Workshop 1 and Workshop 3 brought up latency in **PROBLEM WITH THE EQUIPMENT** less than the 2 other workshops. Latency may be one of the main problems for most people, whereas IT professionals will deal with a wider range of issues. Those groups also produced the only correct examples for **ERROR IN PLANNING**; they had experiences relating to programming that the other workshops did not have. For the same reason (the use of Git), they also produced more examples in **OMISSION**. Both workshops underlined the existence of standards in interface design when taking about **WRONG USER INTERPRETATION**. Workshop 4 also notably mentioned a “software charter”, while talking about **WRONG SYSTEM INTERPRETATION**. Without necessarily having the vocabulary to express it, users may be aware that different applications have and share conventions. HCI researchers and IT professionals produced less examples in **FAIL-UNSAFE** than the other two workshops: when Workshop 1 and Workshop 3 mostly mentioned measures they were taking to prevent such errors, Workshop 2 and Workshop 4 recounted actual occurrences of said incidents.

In all workshops, and despite of the given instructions, some post-its did not directly describe incidents. For Workshop 1 and Workshop 3, some post-its described functionalities they wished for. These groups were more confident in their knowledge of their tools’ features, to the point where they felt comfortable questioning them. On the other hand, in the other two baseline groups, a few post-its described their reactions to incidents: spamming clicks when the interface is freezing, or emotional reactions like guilt and fear (in anticipation). This apparent display of more intense emotions will be explored in greater depth in the thematic analysis below (subsection 2.2.3).

Changes to the categorization

Some reflexions of the participants and *Incidents_W* prompted the creation of 2 new categories to complete the initial categorization. The categories **DARK PATTERNS₊** and **ACCESSIBILITY₊** are marked with a plus sign to indicate this later addition.

DARK PATTERNS₊ Workshop 2 proposed a “consent” category for all situations where the design manipulates the user. If they were the only group to create a category, all groups mentioned dark patterns in their examples, in various categories (once in **ERROR IN PLANNING**, **WRONG USER INTERPRETATION**, **WRONG SYSTEM**

INTERPRETATION, INTERRUPTION, SLIP and twice in INTERACTION INTERFERENCE).

We elected to create a **DARK PATTERN₊** category because they were brought up by participants in the 4 workshops, without being prompted by the experimenter.

Dark Patterns are studied in HCI, they are interface processes that push the user to perform an action that was not their intention [48]. They can take many forms: a fake “close” button leading to the opening of an ad, a countdown timer in online shopping, a checkbox ticked by default, and so on. Dark Patterns have proven to be highly effective, not always detected by the user, and all the more problematic for less educated users[112]. Dark patterns raise the question of intention in error, since they highlight an interface intention that conflicts with the user’s intention.

While all other CATEGORIES discriminate perceived *phenomena or events*, **DARK PATTERN₊** is defined by its perceived intention. Technically, a dark pattern can produce errors that fall into other categories. We added it to the taxonomy because we want to capture errors as perceived by people, and in this case, people were fairly unanimous in considering it a special case.

ACCESSIBILITY₊ One person (P7_{W2}) brought up examples and situations relating to accessibility. They could sometimes not use devices or softwares because they were visually impaired, and those systems either did not have accessibility measures, or unsatisfactory ones. The *Incident_W* was written on a post-it and placed in the **PROBLEM WITH THE EQUIPMENT** category at the DISCUSSION PHASE, with the argument that **ACCESSIBILITY₊** is a functionality of the tool made unavailable to the user. The group argued later for the creation of an **ACCESSIBILITY₊** category.

Accessibility issues are designated as errors in the HCI literature [190], but are not mentioned in the global studies on error.

We elected to create a category for **ACCESSIBILITY₊**, meant to include incidents due to the system not being adapted to the user’s specific physiological limitations. We can note however that disabilities can participate in other kinds of errors: for example, visually impaired people can be particularly sensitive to dark patterns [107].

The table 2.12 summarizes the categorization after the workshops.

If **ACCESSIBILITY₊** and **DARK PATTERN₊** do not benefit from feedbacks as categories presented to the different groups, they were however present in the thematic analysis of the workshops. **Dark Pattern** and **Accessibility** are indeed codes that emerged from the participants’ speech, as presented in the following.

2.2.3 Thematic Analysis

After providing feedback to the categorization, the transcripts of the workshops were analyzed following a Thematic Analysis, as described by Braun and Clarke [23]. When discussing their incidents, participants indeed also talked about their circumstances, occasionally justified their actions or explained those of the system, and expressed emotions related to their experience.

Coding

Process and method After familiarizing ourselves with the data, we proceeded in the coding process, consisting in “*identify[ing] segments of data that appear potentially*

Categories names and definitions
PROBLEM WITH THE EQUIPMENT: The system slows down to the point where it becomes difficult to use, or stops responding completely.
ERROR IN PLANNING: <i>The algorithm you put in place</i> did not lead to the desired results; the plan was flawed.
WRONG USER INTERPRETATION (YOUR INTERPRETATION): <i>The user's</i> interpretation of the effect of a command or action is incorrect.
WRONG SYSTEM INTERPRETATION: The system interprets <i>the user's</i> actions, commands, or gestures the wrong way.
INTERRUPTION: Cognitive disruption during a task. <i>The user is</i> distracted by something on the interface and forgets what they were doing.
INTERACTION INTERFERENCE: The interface changes immediately before a click or keystroke: a pop-up appears, a list is updated, another application takes focus, etc. As a result, the click occurs “in the void” or on an unwanted object.
AIMED ACTION (ACTION): Clicking or tapping next to the intended object: a button, a link, an email, a target, a keyboard key, etc. The click can either occur “in the void” (without consequence) or activate another object or window that is not the intended target.
SLIP: Using a familiar action in a context where it does not apply, and may even produce undesirable effects.
FAIL-UNSAFE: Forgetting or postponing a preventive action, a “just in case” measure
OMISSION: Forgetting a step in the execution of a task or sequence of actions.
DARK PATTERNS₊: The user is nudged to act against their own interests, according to the intentionality present in the system's design
ACCESSIBILITY₊: The system is not properly adapted to the user's specific physiological limitations.

TABLE 2.12: After the workshops, 12 categories of errors. Parentheses indicate our own label before the workshops suggested name changes. Changes made to the definitions are in italics. Plus signs indicate categories added as a result of the workshops.

interesting, relevant or meaningful for [our] research question, and apply pithy, analytically-meaningful descriptions” [23]. We employed both inductive (top-down) and deductive (bottom-up) approaches.

Some codes were deduced from our research questions. For instance, we aimed to gather the participants' understanding of the categories, and their potential comments on their names and descriptions. The categories themselves were thus first codes that we sought in the data. Other codes were then emerged from the data in an inductive approach. What were people expressing when talking about errors? Was there recurring observations that could be made in their speech? These were questions that remained opened for a bottom-up investigation in the transcript.

I started the coding process by looking for specific words and expressions in the transcript. For **Irritation** for instance, I started by searching all the lines that had words starting with “frustrat-” or “annoy-” in it. The first codes came from our inductive approach, and some were deductive, coming from the notes I took during the sessions that were then summarized in reports after each workshop (see appendix B). Words and expressions were added to the list as I went along, emerging from reading the speech I was tagging. While marking speech as an expression of **Irritation** for example, I noticed the expression “being fed up with” that I then looked for in all workshops to potentially tag them with **Irritation** as well.

I then proceeded to code speech by reviewing the transcript top to bottom. Because codes were evolving and added in the process, the order in which I reviewed

the data could bias the produced codes: the first workshop reviewed would create codes and observations, that would influence my approach on the following ones. For this reason, I tagged the speech going from Workshop 1 to Workshop 4 with the word search, and from Workshop 4 to Workshop 1 in a line-by-line reading. I formulated precise definitions for the codes progressively, *i.e.* code labels, as new codes emerged and possible ambiguities were revealed. The codes (and their labels by which we will refer them in the following) are presented in table 2.13.

Code labels and codes

Caused by user: The comment clearly expresses or implies that the problem originated from an action (or lack of action) by a person using the system, speaker or other.

Blaming the user: The speaker specifically assigns blame to the user, or clearly implies it. *Parent code:* **Caused by user.**

Lack of time: The comment clearly expresses or implies that the error stems from a lack of time, from “going too fast,” or from an attempt to save time/go faster.

Lack of money: The comment clearly expresses or implies that the error could have been avoided by spending money.

Incompetence¹²: The comment clearly expresses or implies that the error is due to a lack of competence/expertise, or to incorrect knowledge, or to forgetting a piece of knowledge or a procedure.

Uncertainty: The comment clearly expresses or implies uncertainty, a lack of information, or “losing track” of an error situation.

Lack of attention: The comment clearly expresses or implies that the user is distracted (either by themselves or by something else), not focused, and not being careful.

Carelessness: The comment clearly expresses or implies that the user is consciously acting in a risky, dangerous, or suboptimal manner (either due to fatigue or procrastination).

Awareness: The comment clearly expresses or implies that the user is aware of the possible negative consequences of their actions.

Habit: The comment clearly expresses or implies that the user is acting out of habit or reflex, accustomed to certain interface standards.

Caused by the system: The comment clearly expresses or implies that the error stems from the system, due to its design, its material limitations, or its documentation.

Blaming the system: The speaker assigns blame to the system, or clearly implies it. *Parent code:* **Caused by the system**

Bad design: The comment clearly expresses or implies that the system is poorly designed, the designers made mistakes, or the system should offer additional features.

Breakdown: The comment clearly expresses or implies that the system freezes, crashes, stops working, or a feature becomes unavailable.

Latency: The comment clearly expresses or implies that the system is slow to respond.

Dark Pattern: The comment describes a dark pattern or denial of consent (when identified as problematic, it is also tagged **Blaming the system**)

Miscommunication user/system: The comment clearly expresses or implies that there is a misunderstanding, that the system and the user do not understand each other, do not use the same vocabulary or the same codes.

Accessibility: The comment clearly expresses or implies that the system is not adapted to with disabilities.

Bad Luck: The comment clearly expresses or implies that the problem occurred through bad luck or by accident.

¹²Best translation in French is “impéritie”

Difficulty: The comment clearly expresses or implies that the user is experiencing difficulties; the task is overly difficult or even impossible.

System rationality: The language of the commentary emphasizes the objective, deterministic, fixed nature of the system (“it works as it is programmed”)

System personification: The language of the commentary attributes sensitive or human qualities (intentions, emotions, states such as “struggling”) to the machine or software.

Expectation/reality gap: The comment clearly expresses or implies that there is a discrepancy between the user’s expectations and the responses of the system/interface.

Irritation: The comment clearly expresses or implies irritation or annoyance on the part of the user.

Surprise: The comment clearly expresses or implies that the user is surprised, that an event is sudden and unexpected.

Stress/Apprehension: The comment clearly expresses or implies that the user is stressed, either in anticipation or as a result of their actions.

Depreciation: The comment belittles the user and their qualities: they are stupid, know nothing, are incapable.

Relativization: The comment clearly expresses or implies that a situation is not as risky as it might seem, or that an error or its consequences are “not that serious,” can be remedied, will not last, or do not pose a particular problem.

Criticality: The comment clearly expresses or implies that an error or its consequences are critical, serious, or will last over time.

Focus loss: The comment clearly expresses or implies that the user is distracted as a consequence of an incident.

Time loss: The comment clearly expresses or implies that the user is wasting time.

Data loss: The comment clearly expresses or implies that the user is losing data or work.

Learning: The comment clearly expresses or implies that an incident is part of a learning process.

Solution: The comment clearly expresses or implies an attempt to find a solution to a problem (which does not necessarily imply that a solution is found).

Trial and error: The comment clearly expresses or implies that the user is trying different things in order to eventually succeed in completing the task. *Parent code:* **Solution.**

Frequent: The comment clearly expresses or implies that an incident happens often, or happens to many people.

Sometimes: The comment clearly expresses or implies that an incident happens from time to time, or happens to certain people.

Rare: The comment clearly expresses or implies that an incident is rare, unlikely, or happens to few people.

TABLE 2.13: Codes and code labels from the workshops

The parent/children relationships mentioned in Table 2.13 indicates that all instances of the child code are also instances of the parent code.

Mathieu Nancel then reviewed the codes and the tagging by looking at about 100 random lines in the transcript of the 4 workshops, providing a second opinion on the code labels and on the tagged speech. This prompted discussions from which codes were split, added, or enriched. As stated by Braun and Clarke [23], “*Having only one person coding – usually the researcher – is normal practice, and indeed good practice, for reflexive TA. Where multiple coders can be useful is in developing richer and more complex insights into the data*”. We consequently did not quantify the level of agreement in the

coding between Mathieu and me (IRR). Our goal for the workshops was to identify patterns in the discourse on errors, rather than capturing the exact frequencies at which participants expressed certain things.

Code definitions We coded the data according to what the speech expressed, or clearly implied, and not what we could be tempted to interpret from it. However, we took our expertise into account for technical codes, codes that were descriptive of the error event. Sometimes a participant could use a word erroneously to qualify an event, a word that did not correspond to the described situation. Consequently, if the explanation of an event was clear and detailed enough to clear up any doubts, we could overlook the use of certain erroneous words. For example in the Workshop 4, a participant used the word “latency” when the problem at hand did not involve latency at all. They were in fact describing the pointing difficulties one can experience with thick screens: “[...] *There are applications where the text is very small. And so, depending on your screen, the tool, or your interface, you won’t have the same reaction as with another one. [...] the screens are thick like this [gestures], which means that [...] every time you want to press A, it gives you a Z. [...] That’s what I call “tool latency”, or “hardware latency”*” [P13_{W4}]. In this instance, and despite an explicit instance of the **Latency** code into the speech, we did not tag it as such.

Codes coverages and relations The tags **Caused by user** and **Caused by the system** do not assign blame, but simply an origin or a trigger for a bad event. **Blaming the user** is an assignment on blame on the user by the speaker, either explicitly (“*It’s like a clear zone of “It’s your fault”*” [P5_{W2}]) or implicitly (saying for instance what the user should do: “*Well, you should anticipate and, as soon as you type “attached file”, you should attach your file*” [P15_{W4}]). **Blaming the system** is similar, but pointing at the system: “*the computer is too slow*” [P15_{W4}] or “*the design is so bad*” [P3_{W1}]. Those tags show that talking about error situations, participants had a tendency to point fingers.

Lack of time and **Lack of money** are tagging the circumstances or reasons for the error; the user made decisions with that context in mind. **Incompetence** refers to situations where the user did not know how to do something (“*I don’t know how to do it. It must be possible, but I don’t know how to do it.*” [P13_{W4}]). **Uncertainty** tags speech where the user was confused or lost in front of the interface (“*you don’t know what context you’re in*” [P3_{W1}]). **Lack of attention** is tagged where the user was not paying enough attention, or did something without thinking about it. **Carelessness** is a conscience that the course of action that one chose is not the optimal one, and **Awareness** is an awareness of the risk. The two do not always overlap: sometimes a user chooses a course of action while being aware of the possible bad consequences (“*Sometimes it asks me to update something, and I think to myself, “It’s working fine, I have deadlines to meet, I’m not going to do the update”*” [P12_{W3}]). **Carelessness** is sometimes with good reasons, and sometimes not, and **Awareness** can surround any course of action, careless or not. In all the workshops, descriptions of (self-assigned) laziness were tagged with both. With **Habit**, participants explained their behavior with automatisms or habits that they built, that conflict with what they should be doing in the context. A lot of participants used **Habit** it to explain **SLIPS**, as P5_{W2} talking about mixing buttons layouts on different controllers for instance: “*I know I’m not on the same console, it’s just that I reflexively do one or the other*” [P5_{W2}]. Another explanation of behavior could transpire in **Trial and error**, where the participants explained “testing things out”, trying out actions to see their outcome, knowing that there was a possibility of an unintended result.

In speech tagged with **Bad design**, participants complained about the system not being adapted or efficient, implying that the system should be different (*"It's really poorly done"** [P16_{W4}]). **Breakdown** applies to the device not working anymore, freezing, or lacking an important functionality (for instance, a laptop that cannot function on battery). **Latency** refers to the interface being slow, either because of an aging device, software defects or heaviness or internet connection.

The **Dark Pattern** tag was present in all the workshops, even though it was not presented as a category. Participants sometimes described a dark pattern by describing a bad design (*"for accepting cookies, it wasn't a matter of interpretation; there is no obvious interpretation. The thing is so... you can't tell if it's positive or negative."** [P3_{W1}]). **Miscommunication user/system** refers to difficulties for the user to understand the system (*"you interpret wrongly what is going on"** [P1_{W1}]), or for the system to understand the user (*"Sometimes when you tell the car's GPS where you want to go, it doesn't understand anything you're saying"** [P7_{W2}]). The **Accessibility** tag refers to the system not being adapted to the user's disability (*"There are virtually no websites that are optimized for people with disabilities"** [P7_{W2}]) or having sub-optimal accessibility features (*"When I zoom in, I lose a lot of the information because I no longer have all the text. The text has been enlarged, but I only have the middle of the text, for example, and that's really annoying"** [P7_{W2}]). **Bad Luck** can also explain the cause for an error, without it necessarily coming from the user or the system. Sometimes *"things just happen"** [P12_{W3}].

Expectation/reality gap is the expression of a distance between what the user expected and what happened in the system. **Surprise** is not exactly the same thing: users expressed here the suddenness of an incident, the fact that it was unpredictable, that they were caught off guard (*"[Things] that happen a bit without warning"** [P12_{W3}]). There could be a **Expectation/reality gap** without **Surprise**: the person would have liked the system to act a certain way but was not surprised to be disappointed. An error could create struggle for the user, tagged **Difficulty**. Participants pointed out that some tasks were hard: *"I write in Persian. And Persian is written from [...] right to left. [...] So when I put it in, the layout and everything, it doesn't turn out the way I had it in my head. And I look at a lot of things on the internet, like you said, tutorials and everything. But in the end, it doesn't work."** [P6_{W2}]). Sometimes, tasks were even deemed impossible (*"We found it really difficult to use, even unusable"** [P11_{W3}]).

System rationality tags speech underlying the fact that the system is a machine that "does what it's told" as in *"A system is a strict, straightforward thing. When you tell it to do something, it does what you ask it to do"** [P6_{W2}]). **System personification** is the other side of the same coin, where users attributed human qualities to the system: *"just before clicking, it thinks, 'Wait, I haven't finished searching'"** [P16_{W4}].

Relativization and **Criticality** assess the gravity of the error, being not serious (*"I've clicked on it before, but it's no big deal"** [P7_{W2}]) or grave (*"you drop a nuclear bomb"** [P2_{W1}]).

Irritation was attached to speech expressing anger or frustration: *"It's super annoying"** [P7_{W2}], *"It's infuriating"** [P9_{W3}] or *"That's really the thing that frustrates me the most"** [P1_{W1}]). **Stress/Apprehension** can be in reaction to an error, or in anticipation of its possibility. Participants expressed being stressed (*"the notification 'plug in your computer'. Basically, your computer will shut down. It's scary. You get stressed out, and on top of that you actually don't save anything"** [P10_{W3}]), or feeling guilty, either before (*"I feel guilty for not having the skills to use the equipment"** [P6_{W2}]) or after (*"I think I must have declined a lot of iOS updates at one point; I think I was really behind. I even started to feel guilty for the developers."** [P9_{W3}]) an error. **Depreciation** is directed at the user, in reaction to a failure: *"Yeah, we're a bit stupid there"** [P12_{W3}]. It could

be an immediate or a long term consequence, between feeling stupid in the moment and lowering one's confidence. **Focus loss** is another possible consequence of an incident, especially to **INTERRUPTIONS**: *"Well, it disconnects you, I mean, it disconnects you from the task"** [P5_{W2}]. **Time loss** refers to the speakers expressing they spent a lot of time on something, either doing a task (*"the journey was much longer than originally planned"** [P4_{W1}]) or as a result of an error (*"You gotta start over, yeah"** [P14_{W4}]). An error could also cause **Data loss** to the user. It concerned mainly cases of **FAIL-UNSAFE**: *"you forgot to make a backup and your hard drive crashed and now you have nothing left"** [P2_{W1}].

All consequences to errors were not negative, one could be **Learning** from them. There were mentions of getting acquainted to a system in long-term, doing less and less mistakes: *"That's the kind of thing that happens at the beginning when you start using a new tool. After that, things settle down"** [P10_{W3}]. There were also mentions of incidents that cause the user to never take such a risk again (relating to a **FAIL-UNSAFE** event: *"It happened to me, I lost everything, it pissed me off, I didn't do it again"** [P7_{W2}]). A lot of times participants explained how they looked for a **Solution** to the problem, without necessarily finding it (*"I don't like it, I tried to remove it"** [P1_{W1}]).

The presence of these tags already sheds light on the user's perception of the error. Even though the instructions did not ask them to do so, participants described their experiences, including their circumstances, emotions, and their own judgment of the situation. The following section compares the presence of these codes in the transcripts in order to observe which ones are most prevalent.

Global observations on codes

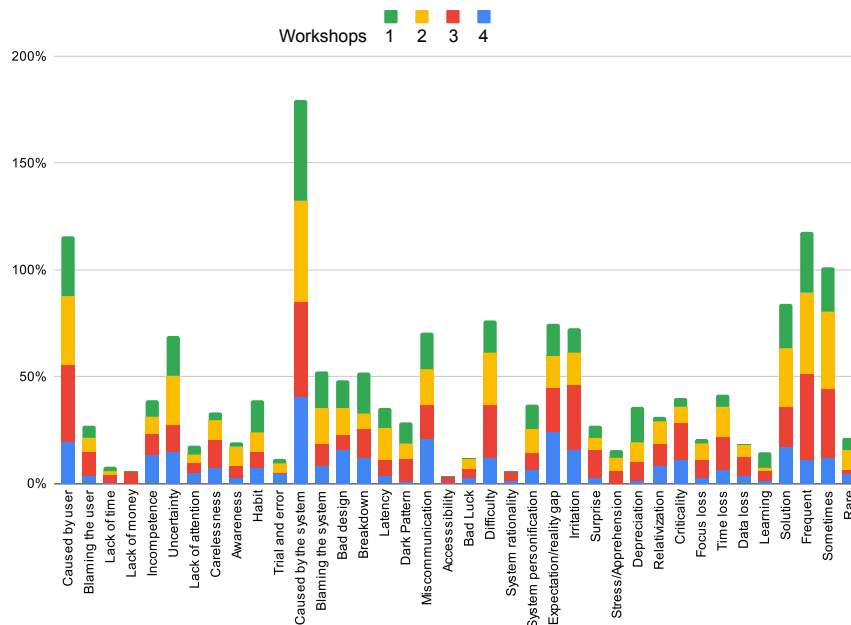


FIGURE 2.2: Percentages of tags within each workshop's examples

Caused by user and **Caused by the system** are the codes that were tagged the most across the transcript; almost every example that was brought up was assigned one or the two. **Blaming the user** and **Blaming the system** are also very present. The notion of blame was deliberately absent from the error categorization in order to discriminate errors by phenomenon rather than by their interpretation, which

varies from person to person and from situation to situation. However, in this case, participants had a strong tendency to point the finger at someone else.

Interestingly, **Caused by the system** had higher numbers than **Caused by user** for every single workshop, as did **Blaming the system** over **Blaming the user**. It appears that in their expression, participants pointed most of the time to the system to be the cause and culprit of problems. It would be interesting to be able to track that trend through time, and in particular to assess whether it has changed compared to the past, but the raw data for this type of investigation is hard to come by.

Expectation/reality gap was one of the most present tags over all the examples. It reinforces our definition of error based on the gap between the user's expectations and their experience.

Among the codes characterizing consequences on the user, **Irritation** was one of the most present throughout examples, along with **Difficulty**. This confirms the commonality of frustration as a user experience (covered in the first chapter, subsection 1.4.1).

With **System rationality** tags, participants were usually talking about the system directly, as P8_{W2}: *"For me, saying that the computer is doing something it's not supposed to do is not very common; there's very little chance involved in a computer. For me, you just clicked on it, and it's not so much that the computer made a mistake as that you didn't know that this behavior would lead to that. And in a way, it's not the system's fault"*. The expression of rationality often takes the form of an assessment while talking about the interaction dynamic in general.

For **System personification** on the other hand, people were usually describing the system behavior in an error situation. Talking about their laptop overheating, P13_{W4} said for instance that *"It's looking for air"*. Other participants attributed qualities such as stupidity (*"Because the system is too stupid, it deletes the file at the same time as it transfers it. It says, 'OK, I've transferred it, now I'll delete it'"* [P1_{W1}]) or thinking (*"the machine thinks you're in one place and you're in another"* [P5_{W2}]). Tags of **System personification** rather come from *how* the participant express themselves rather than *what* they are saying, contrary to **System rationality**.

In all workshops, **System personification** was more present than **System rationality**. It appears that, even though participants knew the system was a machine, they tended to anthropomorphize it while talking about error situations.

Finally, participants expressed their examples to be **Frequent** or to happen with a sort of regularity with the code **Sometimes**. **Rare** incidents were still talked about, even if they happened only once. While one could have imagined that people would talk more about serious rare incidents, this was not the case. Repetition was another reason for them to remember a specific error and share it during the workshops. **Frequent** was also used at times to reinforce an expression of irritation as in *"every time, it's annoying"* [P5_{W2}].

Comparing the categories through codes

Because of the richness of participants' speech, workshops allowed for a first look into what the different CATEGORIES imply in terms of perceived causes, consequences and frequencies. These results came from the expression of participants, they are the aspects of errors that came up without people being specifically directed to them.

The tables 2.14 and 2.15 report the number of *Incidents_W* tagged with each code per CATEGORY, the first in raw numbers and the second in terms of percentage in the CATEGORY. We elected to rectify the CATEGORY of the few incidents that were wrongly sorted by participants. Tables 2.14 and 2.15 report in turn *Incidents_W* that were correct to begin with, or corrected by the experimenters.

	Problem with the equipment	Error in planning	Wrong user interpretation	Wrong system interpretation	Interruption	Interaction interference	Aimed action	Slip	Fail-unsafe	Omission	Means	SD
Number of correct examples	42	6	37	40	28	17	32	25	27	24		
Caused by user	6	4	10	7	3	0	10	10	13	12	7,5	4,2
Blaming the user	3	1	0	0	1	0	2	1	9	3	2,0	2,7
Lack of time	0	1	1	0	0	0	0	0	2	2	0,6	0,8
Lack of money	1	0	0	0	0	0	0	1	2	0	0,4	0,7
Incompetence	4	1	9	4	1	0	1	2	2	2	2,6	2,6
Uncertainty	11	1	12	5	3	1	2	2	6	1	4,4	4,1
Lack of attention	1	0	1	1	2	0	1	4	2	3	1,5	1,3
Carelessness	1	1	5	2	0	0	0	0	13	3	2,5	4
Awareness	1	1	2	1	0	0	0	0	8	0	1,3	2,5
Habit	0	0	6	1	0	0	1	13	0	2	2,3	4,2
Trial and error	1	0	4	1	0	0	0	0	1	1	0,8	1,2
Caused by the system	37	1	14	21	6	7	11	6	10	6	11,9	10,4
Blaming the system	5	0	4	7	1	2	2	2	5	2	3,0	2,2
Bad design	5	0	6	3	0	2	3	5	2	2	2,8	2
Breakdown	25	0	0	2	1	0	0	0	8	1	3,7	7,9
Latency	8	0	0	3	2	4	5	0	0	1	2,3	2,7
Dark Pattern	1	0	5	5	2	2	2	1	0	1	1,9	1,8
Miscommunication user/system	5	0	13	16	1	2	3	6	1	1	4,8	5,5
Accesssibility	1	0	0	0	0	0	1	0	0	0	0,2	0,4
Bad Luck	1	0	1	1	0	2	2	0	2	1	1,0	0,8
Difficulty	5	1	12	5	2	2	5	6	7	3	4,8	3,2
System rationality	0	0	1	4	1	0	0	0	0	0	0,6	1,3
System personification	3	0	2	11	1	1	2	2	3	2	2,7	3,1
Expectation/reality gap	2	4	14	14	1	2	6	2	3	5	5,3	4,8
Irritation	9	0	5	8	4	2	5	7	8	3	5,1	2,9
Surprise	1	0	2	4	3	4	5	1	2	1	2,3	1,6
Stress/Apprehension	1	0	3	1	2	0	0	0	2	1	1,0	1,1
Depreciation	3	0	8	0	1	0	1	4	0	4	2,1	2,6
Relativization	3	0	1	2	3	2	3	2	3	4	2,3	1,2
Criticality	5	0	3	3	4	3	3	2	10	4	3,7	2,6
Focus loss	1	0	0	1	13	0	0	0	1	1	1,7	4
Time loss	5	1	7	2	4	0	2	4	2	1	2,8	2,1
Data loss	3	0	0	0	1	0	0	1	8	1	1,4	2,5
Learning	0	0	2	0	0	0	0	3	4	0	0,9	1,5
Solution	11	0	11	6	4	1	1	4	15	7	6,0	5
Frequent	6	0	11	11	8	6	11	15	8	5	8,1	4,2
Sometimes	6	2	15	9	6	4	9	7	5	7	7,0	3,5
Rare	2	0	1	1	1	1	3	0	2	3	1,4	1,1

TABLE 2.14: Number of *Incidents_W* tagged with each code per CATEGORY. The shades of yellow highlight the biggest values in each lines, *i.e.* the categories for which each code has the most *Incidents_W*.

PROBLEM WITH THE EQUIPMENT Compared to other CATEGORIES, **PROBLEM WITH THE EQUIPMENT** had the most *Incidents_W* (37) featuring an expression of **Caused by the system** ($mean = 11,9, sd = 10,4$ of **Caused by the system**-tagged *Incidents_W* per CATEGORY). In fact, about 88% of the **PROBLEMS WITH THE EQUIPMENT** were tagged with **Caused by the system**. Participants did express quite some **Blaming the system**, but not the highest number accross categories (5 *Incidents_W*, $mean = 3, sd = 2,2$); a **PROBLEM WITH THE EQUIPMENT** seems perceived as coming from the system, but not necessarily something it should be blamed for. Participants assigned their problem to a **Bad design** quite often compared to other categories (5 *Incidents_W*, $mean = 2,8, sd = 2$).

They described by far the most **Breakdown** (25 cases, about 60% of **PROBLEMS WITH THE EQUIPMENT** tagged as such) and **Latency** (8 *Incidents_W*, $mean = 2,3, sd = 2,7$) in this CATEGORY, which is consistent with its given description and examples.

	Problem with the equipment	Error in planning	Wrong user interpretation	Wrong system interpretation	Interruption	Interaction interference	Aimed action	Slip	Fail-unsafe	Omission
Number of correct examples	42	6	37	40	28	17	32	25	27	24
Caused by user	14%	67%	27%	18%	11%	0%	31%	40%	48%	50%
Blaming the user	7%	17%	0%	0%	4%	0%	6%	4%	33%	13%
Lack of time	0%	17%	3%	0%	0%	0%	0%	0%	7%	8%
Lack of money	2%	0%	0%	0%	0%	0%	0%	4%	7%	0%
Incompetence	10%	17%	24%	10%	4%	0%	3%	8%	7%	8%
Uncertainty	26%	17%	32%	13%	11%	6%	6%	8%	22%	4%
Lack of attention	2%	0%	3%	3%	7%	0%	3%	16%	7%	13%
Carelessness	2%	17%	14%	5%	0%	0%	0%	0%	48%	13%
Awareness	2%	17%	5%	3%	0%	0%	0%	0%	30%	0%
Habit	0%	0%	16%	3%	0%	0%	3%	52%	0%	8%
Trial and error	2%	0%	11%	3%	0%	0%	0%	0%	4%	4%
Caused by the system	88%	17%	38%	53%	21%	41%	34%	24%	37%	25%
Blaming the system	12%	0%	11%	18%	4%	12%	6%	8%	19%	8%
Bad design	12%	0%	16%	8%	0%	12%	9%	20%	7%	8%
Breakdown	60%	0%	0%	5%	4%	0%	0%	0%	30%	4%
Latency	19%	0%	0%	8%	7%	24%	16%	0%	0%	4%
Dark Pattern	2%	0%	14%	13%	7%	12%	6%	4%	0%	4%
Miscommunication user/system	12%	0%	35%	40%	4%	12%	9%	24%	4%	4%
Accessibility	2%	0%	0%	0%	0%	0%	3%	0%	0%	0%
Bad Luck	2%	0%	3%	3%	0%	12%	6%	0%	7%	4%
Difficulty	12%	17%	32%	13%	7%	12%	16%	24%	26%	13%
System rationality	0%	0%	3%	10%	4%	0%	0%	0%	0%	0%
System personification	7%	0%	5%	28%	4%	6%	6%	8%	11%	8%
Expectation/reality gap	5%	67%	38%	35%	4%	12%	19%	8%	11%	21%
Irritation	21%	0%	14%	20%	14%	12%	16%	28%	30%	13%
Surprise	2%	0%	5%	10%	11%	24%	16%	4%	7%	4%
Stress/Apprehension	2%	0%	8%	3%	7%	0%	0%	0%	7%	4%
Depreciation	7%	0%	22%	0%	4%	0%	3%	16%	0%	17%
Relativization	7%	0%	3%	5%	11%	12%	9%	8%	11%	17%
Criticality	12%	0%	8%	8%	14%	18%	9%	8%	37%	17%
Focus loss	2%	0%	0%	3%	46%	0%	0%	0%	4%	4%
Time loss	12%	17%	19%	5%	14%	0%	6%	16%	7%	4%
Data loss	7%	0%	0%	0%	4%	0%	0%	4%	30%	4%
Learning	0%	0%	5%	0%	0%	0%	0%	12%	15%	0%
Solution	26%	0%	30%	15%	14%	6%	3%	16%	56%	29%
Frequent	14%	0%	30%	28%	29%	35%	34%	60%	30%	21%
Sometimes	14%	33%	41%	23%	21%	24%	28%	28%	19%	29%
Rare	5%	0%	3%	3%	4%	6%	9%	0%	7%	13%

TABLE 2.15: Percentage of $Incidents_W$ of each CATEGORY tagged with each code. The shades of yellow highlight the biggest values in each lines, *i.e.* the categories for which the percentage of $Incidents_W$ tagged with each code is the highest.

Participants largely described problems of the system being slow, freezing or crashing.

They expressed a lot of **Uncertainty** compared to other CATEGORIES (11 $Incidents_W$, $mean = 4.4$, $sd = 4.1$): they lacked information to understand what was going on. They also felt frustrated or annoyed, with quite some $Incidents_W$ tagged with **Irritation** (9 $Incidents_W$, $mean = 5.1$, $sd = 2.9$) and sometimes shared losing time because of the event (5 **Time loss**-tagged $Incidents_W$, $mean = 2.8$, $sd = 2.1$). They also at times shared the actions they put in place to deal with the problem: more than 25% of $Incidents_W$ were tagged **Solution**.

ERROR IN PLANNING This CATEGORY only counted 6 corrects $Incidents_W$ accross all workshops, as the only situations that fitted its description were related to the design of an algorithm by the user. From this small number of occurrences, one can however remark that most of them (4) were seen as **Caused by user** ($mean = 7.5$, $sd = 4.2$) and a **Expectation/reality gap** ($mean = 5.3$, $sd = 4.8$).

WRONG USER INTERPRETATION This CATEGORY had $Incidents_W$ that were talked about as **Caused by the system** (14, $mean = 11.9$, $sd = 10.4$) more so than **Caused**

by user (10, $mean = 7.5, sd = 4.2$), but by a small margin. Those codes represented respectively about 38% and 27% of *Incidents_W* of this category. In terms of blame, participants sometimes (4 *Incidents_W*) expressed a **Blaming the system** ($mean = 3, sd = 2.2$), but never a **Blaming the user**. Compared to other categories, **WRONG USER INTERPRETATION** had the most *Incidents_W* (4) tagged as **Trial and error** ($mean = 0.8, sd = 1.2$); those errors could be caused at times by the user testing things out. The presence of 5 *Incidents_W* tagged with **Carelessness** ($mean = 2.5, sd = 4$) supports this depiction of the user behavior taking risks.

Errors in this CATEGORY were described as **Expectation/reality gap** the most with 14 tagged *Incidents_W* (38%), tied with **WRONG SYSTEM INTERPRETATION**, compared to others ($mean = 5.3, sd = 4.8$). There were also 13 *Incidents_W* mentioning a **Miscommunication user/system** ($mean = 4.8, sd = 5.5$), representing about 35% of the *Incidents_W* of this category. This category had the most *Incidents_W* depicting a **Bad design** (6, with $mean = 2.8, sd = 2$). People depicted the most **Dark Patterns** for **WRONG USER INTERPRETATION** with 5 *Incidents_W* ($mean = 1.9, sd = 1.8$), along with **WRONG SYSTEM INTERPRETATION**.

As consequences, participants of the workshops expressed the most **Difficulty** interacting with the system for instances of this category (12 *Incidents_W* with $mean = 4.8, sd = 3.2$), along with the most **Uncertainty** (12 *Incidents_W* with $mean = 4.4, sd = 4.1$) and **Incompetence** (9 *Incidents_W* with $mean = 2.6, sd = 2.6$). About 32% of *Incidents_W* in this category were tagged with **Difficulty**, similarly to **Uncertainty**, and 24% with **Incompetence**. Overall, this category had people speak about struggle the most, as they were in situations where they lacked information or skills to solve. Some of those cases provoked **Irritation** (5 cases, $mean = 5.1, sd = 2.9$). When comparing to other CATEGORIES, **WRONG USER INTERPRETATION** had the highest number of *Incidents_W* featuring **Stress/Apprehension** (3, with $mean = 1, sd = 1.1$) and expressions of **Depreciation** (8 cases, $mean = 2.1, sd = 2.6$). The situations where people felt anxiety, anticipation, and where they belittled themselves the most were mostly described in **WRONG USER INTERPRETATION**. This category also reported the most cases tagged with a **Time loss** consequence (7 with $mean = 2.8, sd = 2.1$). People also mentioned looking for a **Solution** in about 30% of cases.

Events were expressed as quite regular, as 41% of *Incidents_W* happened **Sometimes**, and 30% were tagged **Frequent**.

WRONG SYSTEM INTERPRETATION Similarly to **WRONG USER INTERPRETATION**, this CATEGORY presented more *Incidents_W* tagged with **Caused by the system** (21, $mean = 11.9, sd = 10.4$) than **Caused by user** (7, $mean = 7.5, sd = 4.2$) and had no cases where participants were **Blaming the user**. **WRONG SYSTEM INTERPRETATION** and **WRONG USER INTERPRETATION** mirror each other, and one could have expected to see this reflected in who or what people saw as the cause of incidents. Instead, in both cases, the system was more often seen as the source of the problem than the user. **WRONG SYSTEM INTERPRETATION** also had the highest number (7) of *Incidents_W* **Blaming the system** ($mean = 3, sd = 2.2$).

Most **Miscommunication user/system** problems were reported in this CATEGORY (16 *Incidents_W*, with $mean = 4.8, sd = 5.5$); representing about 40% of **WRONG SYSTEM INTERPRETATIONS**. As stated before, participants expressed more **Expectation/reality gaps** (14, $mean = 5.3, sd = 4.8$, representing 35% of **WRONG SYSTEM INTERPRETATIONS**), and described more **Dark Patterns** (5, $mean = 1.9, sd = 1.8$) in **WRONG SYSTEM INTERPRETATION** compared to other categories, along with **WRONG USER INTERPRETATION**.

These kinds of errors caused **Irritation** (8 *Incidents_W*, *mean* = 5.1, *sd* = 2.9) and **Surprise** (4 cases, *mean* = 2.3, *sd* = 1.6) at times. More importantly, most of the expressions of **System rationality** (4 *Incidents_W*, *mean* = 0.6, *sd* = 1.3) and **System personification** (11 cases, *mean* = 2.7, *sd* = 3.1) accompanied *Incidents_W* of this category. In fact, about 28% of **WRONG SYSTEM INTERPRETATIONS** bore the **System personification** code. For these kinds of events, participants shared an idea of the system as “a machine that only does what it is told to do” and, at the same time, assigned it in their speech the most human qualities. When **System rationality** and **System personification** seem to oppose one another, they in fact appear to go hand in hand; the events for which participants rationalized the system as a machine are also the ones for which they described it as “struggling”, “understanding”, “waiting” etc.

Finally, occurrences in **WRONG SYSTEM INTERPRETATION** were somewhat frequent, as 9 *Incidents_W* bore the code **Sometimes** (23%), and 11 **Frequent** (28%).

INTERRUPTION Interestingly, this CATEGORY did not present strong tendencies for cause assignation or blame in the workshops. Only about 21% of cases bore the code **Caused by the system**, and 11% **Caused by user**. **Blaming the user** and **Blaming the system** are both tagged on about 4% of **INTERRUPTION** examples. Contrary to other CATEGORIES, participants did not point much at an origin, or a culprit for these kinds of errors. That could be because, consistently with the literature, examples of incidents included messages from both the system and other people.

The consequence expressed the most for **INTERRUPTIONS** is a **Focus loss**, which tagged 13 *Incidents_W* (*mean* = 1.7, *sd* = 4), about 46% of the *Incidents_W* for that category. Instances of **Relativization** were also present in 11% of cases, which may explain the low number of tags for this category (other than for **Focus loss**); the consequences might not seem serious or noteworthy to participants. We do not observe a strong **Irritation** trend compared to other CATEGORIES.

INTERACTION INTERFERENCE For this CATEGORY, the assignation of blame is very clear: 41% of *Incidents_W* are tagged **Caused by the system**, 12% **Blaming the system** and none are tagged **Caused by user** or **Blaming the user**. **INTERACTION INTERFERENCES** were perceived as coming from and because of the system, and not once were assigned to the user’s actions. With 2 *Incidents_W* mentioning **Bad Luck**, the category contains 12% of the cases tagged with that code.

The incident was assigned to **Latency** at times, with 4 *Incidents_W* tagged as such in the category (*mean* = 2.3, *sd* = 2.7), representing 24% of **INTERACTION INTERFERENCES**. 12% of those incidents described **Dark Patterns**.

These kinds of events provoked **Surprise** (4 *Incidents_W* *mean* = 2.3, *sd* = 1.6). Finally, they sometimes, in 12% of cases, were accompanied by expression of **Relativization**, as the consequences were not perceived as critical.

ACTION For this CATEGORY, the expressions of cause and blame went equivalently in both directions: 10 *Incidents_W* (31%) are tagged with **Caused by user** (*mean* = 7.5, *sd* = 4.2), and 11 (34%) with **Caused by the system** (*mean* = 11.9, *sd* = 10.4). The numbers for **Blaming the system** and **Blaming the user** are identical, with 2 *Incidents_W* for each. Compared to other categories, **Bad Luck** was more presented as a cause (2 *Incidents_W* tagged as such, with *mean* = 1, *sd* = 0.8).

The phenomenon described involved a depiction of a **Latency** or slowness to the interface 16% of the time (2 cases, *mean* = 2.3, *sd* = 2.7).

ACTIONS had the most *Incidents_W* mentioning a feeling of **Surprise**, with 5 cases ($mean = 2.3, sd = 1.6$). 3 incidents also fostered expressions of **Relativization** ($mean = 2.3, sd = 1.2$).

SLIP Descriptions of **SLIPS** came with *Incidents_W* tagged more with **Caused by user** (10, $mean = 7.5, sd = 4.2$) than **Caused by the system** (6, $mean = 11.9, sd = 10.4$). Expressions of blame were not very strong: about 8% of **SLIPS** were tagged **Blaming the system**, and 4% **Blaming the user**. The event was caused at times by a **Lack of attention** (4 *Incidents_W*, $mean = 1.5, sd = 1.3$) on the part of the user, along with the description of performing out of **Habit** 52% of the time (13 *Incidents_W*, $mean = 2.3, sd = 4.2$), both more so than other CATEGORIES. There were also depictions of **Bad designs** for **SLIPS** for 20% of its *Incidents_W*. In terms of perceived and shared causes of **SLIPS**, participants pointed to their habits, their lack of attention, but also the system design. The overall perceived responsibility seems shared.

24% of incidents were accompanied by speech about **Miscommunication user/system**, which is coherent with a problem coming both from the system and the user.

Mentioned consequences of **SLIPS** included **Irritation** in 28% of cases (7 *Incidents_W*, $mean = 5.1, sd = 2.9$) and struggle (24% **Difficulty**-tagged *Incidents_W*). 16% of *Incidents_W* included **Time loss**-tagged speech. The same proportion featured expressions of **Depreciation**. However, 12% of cases also came with the **Learning** code: participants described the error as being part of a learning process. Finally, **SLIPS** had the most **Frequent**-tagged *Incidents_W*, both in number (15, $mean = 8.1, sd = 4.2$) and in percentage in the CATEGORY (60%).

FAIL-UNSAFE This CATEGORY had the most *Incidents_W* tagged with **Caused by user** (13, with $mean = 7.5, sd = 4.2$ for the **Caused by user** tag across all CATEGORIES), and a bit less **Caused by the system** (5, $mean = 11.9, sd = 10.4$). Mostly, it had the most **Blaming the user**-tagged *Incidents_W* compared to other CATEGORIES, both in number (9, $mean = 2.0, sd = 2.7$) and proportion (33%) among **FAIL-UNSAFE** acts. There was however also expressions of **Blaming the system** in 19% of cases. The pointing to the user was frequent for these kinds of incidents, but not undisputed. Participants mentioned a **Lack of time** (2 *Incidents_W*, $mean = 0.6, sd = 0.8$) or a **Lack of money** (2 cases, $mean = 0.4, sd = 0.7$) at times to explain their behavior. Interestingly, the CATEGORY had the highest number of *Incidents_W* with both **Carelessness** (13, $mean = 2.5, sd = 4$) and **Awareness** (8, $mean = 1.3, sd = 2.5$). These codes were tagged on 48% and 30% of **FAIL-UNSAFE** acts respectively. Participants described taking risks, and acting hastily, while being aware of doing so. **Bad Luck** was also mentioned as a possible cause (2 cases, $mean = 1, sd = 0.8$).

Those errors provoked **Irritation** (8 *Incidents_W*, $mean = 5.1, sd = 2.9$), and more **Data loss** (30% with 8 cases, $mean = 1.4, sd = 2.5$) than in any other CATEGORY. *Incidents_W* of this CATEGORY had the most expressions of **Criticality** (37% of cases), along with some **Relativization** (11%). The consequences of incidents were serious, but sometimes put in perspective. Participants reported **Learning** from the error more than in other categories: 15% of *Incidents_W* were tagged as such. They also shared looking for **Solutions** to manage the error more than half of the time (56% of cases), which is the highest percentage for that code across all CATEGORIES.

The frequencies of incidents were quite varied, with no strong tendency compared to other types of errors.

OMISSION For this CATEGORY, 50% of $Incidents_W$ were expressed as **Caused by user**, and very few as **Caused by the system** (8%). There was no big difference between **Blaming the system** and **Blaming the user**, and not a lot of blame-tagged examples overall (3 and 2 respectively). Events were sometimes caused by a **Lack of attention** on the part of the user (13% of cases).

OMISSIONS provoked expressions of **Depreciation** in 17% of $Incidents_W$ for this CATEGORY. However, people also took a step in regard to these incidents, 17% of cases were accompanied by speech of **Relativization**.

Finally, **OMISSION** had more **Rare** $Incidents_W$ than any other category (3 cases, $mean = 1.4, sd = 1.1$).

Comparing the workshops populations through codes

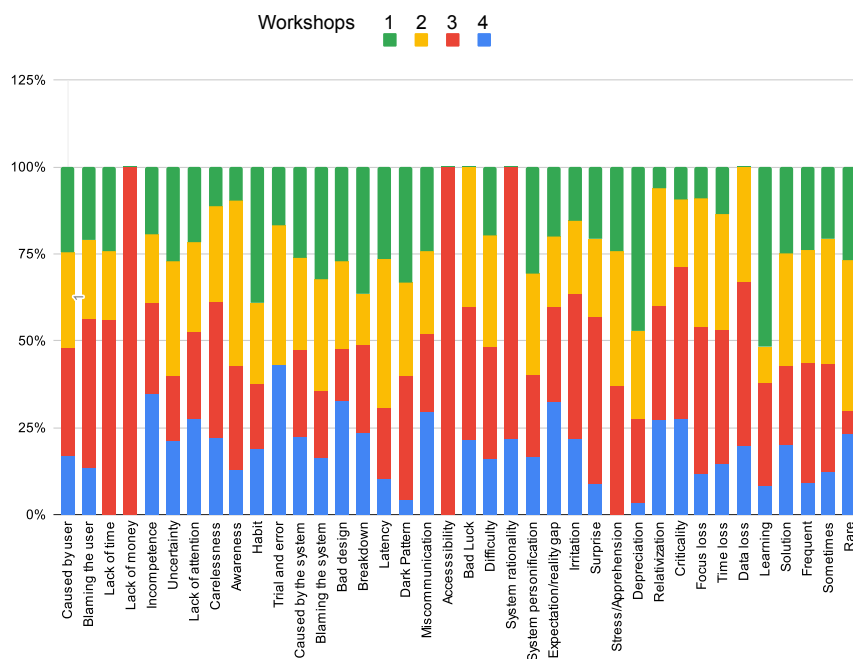


FIGURE 2.3: Percentages of examples within each workshop tagged with each code

Workshops 2 and 4 (baseline) vs. 1 and 3 (computer scientists) Expressions of **Uncertainty** were sometimes absolute in Workshop 2 and Workshop 4, more so than in the other two workshops, with comments like *“But you wouldn’t know anything about that, so...”* [P16_{W4}] or *“I act like I know what I’m talking about, but I don’t know anything about this stuff”* [P5_{W2}]. Workshop 1 and Workshop 3’s examples were more often explained with a **Lack of attention** on the part of the user (28% and 26% vs. 25% and 22%), which could be interpreted as a more relaxed attitude using their devices; maybe knowing *when* to lower their guard. In the other 2 workshops however, the code was more often and more intensely accompanied by self blame (*“That’s right, it’s my fault. Like I’m the one who loses focus”* [P5_{W2}]). **Depreciation** in baseline workshops was often associated with an expression of incompetence with digital tools: *“it always happens to me, but that’s because I’m lousy at computers”* [P7_{W2}].

Participants reported functioning in **Trial and error** a lot more in Workshop 1 (43% of **Trial and error** examples) and Workshop 3 (40%). Expertise with their interfaces probably gives them the confidence to try things out, knowing that they can

revert their actions. They may also be more aware of their limits and when to stop experimenting. They notably explained operating in this mode in reaction to the **ERROR IN PLANNING** category.

Finally, baseline workshops reported more examples involving a **Learning** process: 30% and 52% of **Learning** tagged examples were in Workshop 2 and Workshop 4 respectively. They reported learning keyboard shortcuts and not repeating postponing a **FAIL-UNSAFE** after an incident.

Workshop 2 specifically was the only one tagged with **Lack of money**, for which participants were mostly talking about hard drives in **FAIL-UNSAFE**. This workshop also contained all speech relating to **Accessibility**, only P7_{W2} reported problems relating to their visual impairment. A lot of times, examples bearing the code **Accessibility** are also tagged **Bad design**; the person was blaming the system and not themselves for those difficulties (*"there are virtually no websites that are optimized for people with disabilities. Even government websites aren't. I want to rant a little, even though it won't make any difference."* [P7_{W2}]).

Workshop 1 (HCI researchers) The speech tagged with **Blaming the user** was a little different for this workshop than for the others. Here, "the user" in those citations was usually a theoretical one, either at the third person (talking about their students) or a generalized "you" (*"Often the thing tells you 'Hey, you haven't done a backup in 10 days.'"* [P1_{W1}]). With this general idea of a user, they had the most examples tagged with **Incompetence** (35% of those examples, against 26% at most in the other workshops). Probably in part because they did not embody the user, they also expressed less **Depreciation** (only 3%).

As we could expect with HCI researchers, they mentioned **Miscommunication user/system** and **Expectation/reality gap** a little bit more than other workshops: 29% of all **Miscommunication user/system** tagged examples were in Workshop 1 (against 24% max. for the other three), along with 33% of all **Expectation/reality gap** (against 27% max.). Participants blamed the system less than other workshops (16% vs. 19% min. for **Blaming the system**).

They also did not express **Stress/Apprehension** at all.

This workshop also brought up **DARK PATTERNS**₊ less than the others – only 4% of examples with this code were from this group. One hypothesis to explain this is that this group may have been more respectful of the specifications for each category, and deviated from them less. In the other groups, dark patterns were often the subject of conversation deviations. In those groups, some **Dark Pattern** were recognized as intentional, as manipulations from the designer as in *"I think your ad is the system that makes you click on purpose."* [P16_{W4}].

Codes and themes

The workshop exercise proved to be very rich, not only in terms of the *Incidents_W* that were brought up, but also in the way people talked about them. While tagging the transcripts, codes emerged from participants speech, relating to aspects I did not directly ask for. Interestingly, most codes came up this way, during the thematic analysis, in a bottom-up approach.

I divided the codes that emerged from the workshops in 6 themes: *User's state and behavior*, *System's state and behavior*, *Assigned error explanation*, *Impact on the user*, *User perception* and *Frequencies*. The table 2.16 summarizes this organization.

Themes	Codes
<i>User's state and behavior</i>	Lack of time, Lack of money, Incompetence, Uncertainty, Lack of attention, Carelessness, Awareness, Habit, Trial and error
<i>System's state and behavior</i>	Bad design, Breakdown, Latency
<i>Assigned error explanation</i>	Caused by user, Caused by the system, Blaming the user, Blaming the system, Dark Pattern, Miscommunication user/system, Accessibility, Bad Luck, Expectation/reality gap
<i>Impact on the user</i>	Difficulty, Irritation, Stress/Apprehension, Surprise, Depreciation, Focus loss, Time loss, Data loss, Learning, Solution, Relativization, Criticality
<i>User perception</i>	System rationality, System personification
<i>Frequencies</i>	Frequent, Sometimes, Rare

TABLE 2.16: Workshop codes grouped by themes

When talking about their incidents, people justified their behavior and state of mind. They had constraints (**Lack of time** or **Lack of money**), they lacked focus (**Lack of attention**), were unsure (**Uncertainty**) or lacked skills or informations (**Incompetence**). They also acted in a certain state of mind, they were purposely taking risks (**Carelessness**), knowingly or not (**Awareness**), were trying out different things (**Trial and error**) or simply acted out of **Habit**.

User's state and behavior paints an interesting picture of the user, who makes decisions based on their constraints and skills, but also on perceived risks. However, their behavior is not always the result of rational negotiation between these elements, but also of simple automatisms. Sometimes, users take the easiest route, even if they are aware that it is not the most effective.

The system had fewer justifications for its behavior. Participants either described it as slow (**Latency**), poorly designed) (**Bad design**), or simply broken (**Breakdown**). **Latency** did not come with solutions, just as a state of facts. Descriptions of **Bad design** did not foster inferences on the competence of the designer. In the same way, **Breakdown** was often described as definitive.

Overall, *System's state and behavior* shows that the system was described in a rather static manner, meaning that participants did not often perceive it as having the potential to change.

While describing errors, people shared their interpretation of why they happened. First, they described the error as originating from the user (**Caused by user**), the system (**Caused by the system**), or both, and at times expressed blame in either directions (**Blaming the user** and **Blaming the system**). Other causes were assigned, such as the incident happening by **Surprise**, or through **Bad Luck**. They also characterized different types of situations, with **Dark Pattern**, **Accessibility** or **Miscommunication user/system**. Other explanations were simply describing a **Expectation/reality gap**, on which our definition of error is based.

Participants shared the impact errors had as well. A number of codes refer to immediate material consequences, as **Data loss** or **Time loss**. They were however not very present throughout the workshops. Negative emotions and experiences, like

Difficulty, **Focus loss**, **Irritation**, **Stress/Apprehension** or **Surprise** were a lot more expressed. They could go as far as being long term, creating self-**Depreciation**. People also talked about looking to solve errors, with success or not. This could potentially lead to **Learning**. Some speech finally shared an idea of the error's criticality (**Relativization** and **Criticality**).

Impact on the user contains mostly negatively charged codes, which makes sense while talking about negative experiences. People however saw sometimes the big picture, acknowledging an ongoing learning process, or the small significance of some incidents.

Participants showed opposite views of the system **System rationality** and **System personification**, and went from one to the other throughout the discussion.

In the *User perception*, those concepts likely co-exist. They know the system to be a machine that “only does what it is told”, but they also perceive it as expressing human emotions or behaviors. The fact that **System rationality** generally concerns speech about the system in general and **System personification** characterizes it in an error experience allows this paradox; the system is not perceived the same in knowledge and in experience.

Finally, participants included expressions of *Frequencies* in their depiction of negative events, sometimes as an argument to justify how annoying or unpractical the situation is.

2.2.4 Discussion

Implications for the categorization

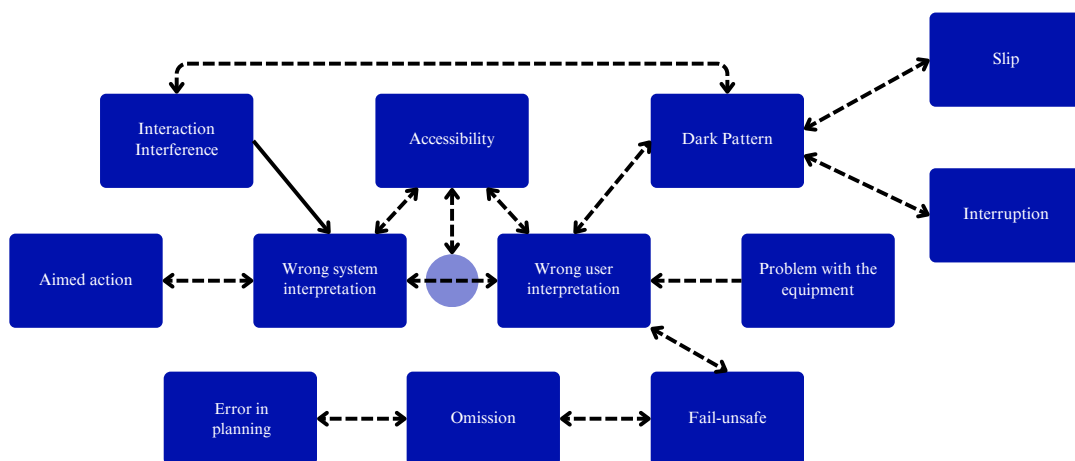


FIGURE 2.4: Schema of categories inter-relationships. Dotted arrows represent a partial overlap of categories (“Some **INTERRUPTIONS** are **DARK PATTERNS**”), and full arrows an inclusion (“all **INTERACTION INTERFERENCES** are **WRONG SYSTEM INTERPRETATION**”).

The categorization presented in Figure 2.4, and earlier in section 2.1, distinguishes different kinds of events. Contrary to most taxonomies presented in chapter 1, it is not constructed on assignment of cause or fault. Instead, those criteria are transversal to **CATEGORIES**: every one of them can have errors attributed to the user or the system. For instance, **WRONG USER INTERPRETATION** includes cases blaming a bad design, but also blunders from the user.

Interestingly, the two workshops of computer scientists were the ones to propose “Bad Design” as a category distinct from **WRONG USER INTERPRETATION**. They may feel that a design can be objectively assessed as bad, and will be seen as such by everyone. However, they were also the groups that blamed themselves with less intensity, and wouldn’t resort to the argument of being helpless with computers. The same situation of **WRONG USER INTERPRETATION** might be assigned differently in terms of blame depending on the person, their expertise, and their overall confidence with digital devices.

As established prior to the workshops, **INTERACTION INTERFERENCE** is a sub-category to **WRONG SYSTEM INTERPRETATION**: the system should not consider that a click a few milliseconds after an interface update represents the user’s intentions. However, if we were to define **ACCESSIBILITY₊** as *the system being adapted to the user’s physiological limitations*, **INTERACTION INTERFERENCE** becomes then a sub-category of it. All interferences are due to the user not having the time to react to changes in the interface, and the interface not taking that physiological reality into account. Some instances of **AIMED ACTION** could also fit in **ACCESSIBILITY₊** with that perspective, with for instance pointing tasks with targets too small for the user to select successfully. The workshops provided experiences related to visual impairment, but **ACCESSIBILITY₊** issues can happen in other ways, for instance with a hearing impairment or tremors.

However **ACCESSIBILITY₊** encompasses not only inputs that are wrongly interpreted by the system, but also some difficulties or impossibilities for the user to perceive and interpret the interface. In turn, we consider **ACCESSIBILITY₊** to be part of both **WRONG USER INTERPRETATION** and **WRONG SYSTEM INTERPRETATION**.

Some **AIMED ACTIONS** could arguably be seen as **WRONG SYSTEM INTERPRETATION** as well, with the system misinterpreting where the user wanted to point (on a thick screen for instance).

DARK PATTERNS₊ is a particular **CATEGORY**: instead of describing an event or a phenomenon, it is defined on the intention behind said phenomenon. Because of this, they can be the cause of other types of categories of error, typically a **WRONG USER INTERPRETATION**, an **INTERRUPTION** or a **SLIP** made on purpose to nudge the user towards certain actions. Paradoxically, their authors also have to rely on every other aspect of the interaction to go as planned: the user has to be paying attention, to click where and when expected, etc.

SLIPS are not a matter of **WRONG USER INTERPRETATION** or **WRONG SYSTEM INTERPRETATION**. They are reflexes and automatisms being triggered in the wrong context. Because they happen on this level, *without users thinking about it*, they are not dependent on user interpretation. Likewise, because they do not reflect an actual intention of the user, they cannot be seen as actions misinterpreted by the system either.

PROBLEM WITH THE EQUIPMENT questioned all groups, in that they wanted to split it into different kinds of such problems. They however did not reach a consensus, sometimes even among themselves. Participants wanted to characterize these problems according to their assigned causes, on which they could disagree. We however wished for **CATEGORIES** to encompass phenomena or kinds of events, not their interpretation in that sense. **PROBLEM WITH THE EQUIPMENT** can also be a **WRONG USER INTERPRETATION**, in cases where the user does not understand in what mode

it is, and is unable to use it, as for instance P13_{W4} not understanding that their laptop needed to be plugged in: “[the laptop]’s very old, it’s 8 years old. So I turned it upside down so it could breathe and start up, and it made airplane-like noises, and then I thought, maybe it just needs more juice. I spent fifteen minutes struggling. And then it turned out that there just wasn’t enough battery power for it to restart.”*

Examples of **FAIL-UNSAFE** can be **WRONG USER INTERPRETATION** in the same way, if the user does not know or understand from the system what fail-safe measures are necessary. Other cases of **FAIL-UNSAFE** can also be **OMISSIONS**; the person knows what to do and intends to, but simply forgets to get to it.

ERROR IN PLANNING, initially proposed in [81], is still a part of the categorization, but in our answers the only examples that participants provided consisted in algorithmic planifications. The other examples given indeed did not correspond to the definition. Some groups (Workshops 1 and 3) underlined that interaction is more about reaction, and “thinking as you go”, than precise planning susceptible to go wrong. Among programming issues, **OMISSIONS** can happen, and be the reason for a code not to work for instance.

Some of the categories did not change names, even if participants brought up issues and proposed alternatives. Some phenomena are indeed characterized in the literature with a specific denomination, such as **INTERRUPTIONS**, **INTERACTION INTERFERENCES**, **ERRORS IN PLANNING** or **SLIPS**. Changing the name in the categorization would complexify links between research projects, hence why we elected to not change them here. However, these feedbacks have real value for user studies. To explain the concepts of **INTERACTION INTERFERENCES** or **INTERRUPTIONS** to non-specialists, one might better use “Irruptions” and “Distractions” respectively, or other denominations that convey more meaning for them.

The representation of the **CATEGORIES** and their potential overlaps in Figure 2.4 have clear delimitations. Some situations can however be difficult to assign. For instance, forgetting to put the attached file to an email can sometimes be an **OMISSION** (telling oneself that one will for sure do it later, and then forgetting), and sometimes a **SLIP**, if the person clicked the Send button reflexively.

While being a simplification¹³, this representation is the first to categorize errors according to user experiences, and not assigned causes.

The analysis of workshops also painted a first portrait of each **CATEGORY**. **PROBLEMS WITH THE EQUIPMENT**, **WRONG USER INTERPRETATIONS**, **WRONG SYSTEM INTERPRETATIONS**, **INTERACTION INTERFERENCES** and **FAIL-UNSAFE** acts were for instance more expressed as coming from the system, when **ERRORS IN PLANNING**, **SLIPS** and **OMISSIONS** seemed to be more attributed to the user. Other **CATEGORIES** presented fewer expressions (1 or 2) of cause or blame in the participants’ speech. People justified and described errors in different ways depending on the type of incidents: **WRONG USER INTERPRETATIONS** were more prompted by a lack of information or skill, **FAIL-UNSAFE** acts by carelessness, **SLIPS** by built habits and lack of attention, **PROBLEMS WITH THE EQUIPMENT** by the system breaking or slowing down, **INTERACTION INTERFERENCES** by latency etc. The consequences shared among them also differed depending on the situations discussed. Participants talked about loosing focus more when talking about **INTERRUPTIONS**, loosing

¹³as all models are

data with **FAIL-UNSAFE** acts or belittled the user more with **WRONG USER INTERPRETATIONS** for instance. Those tendencies were observed within conversations on the different **CATEGORIES**, not prompted particularly by the instructions, which frames them in a specific way: they are expressions and remarks that people shared “organically” while describing their errors. Those conversations included 16 people in total, which encourages a larger quantitative study to corroborate or discuss these observations.

While our categorization presents intersections, it also introduces categories with specific user experiences: all of them describe perceived phenomena, with the exception of **DARK PATTERN₊** covering perceived malicious intent from the system. We therefore elected to keep it as such for the following study.

Observations of users facing errors

The workshops had users sit around a table and talk about their negative experiences with digital devices. The exercise sparked rich discussions, and their analysis revealed certain trends in people’s statements, sometimes based on their expertise in computer science.

The two groups of computer scientists were more likely to talk about things they wished for, for instance a functionality they thought was lacking in a tool. These statements described proper negative experiences, errors in the eyes of these users, even if the system behavior was not against its specifications. The knowledge that “things could be different” could be a source of frustration for them. On the other hand, the “laypeople” workshops did not reveal such causes for frustrations, but instead more uncertainty regarding the system behavior, which could provoke frustration as well.

Computer scientists were also a lot more prone to describe **Trial and error** in their functioning. This exploratory approach to the interface suggests a certain confidence, or at least a sense of control over the setting in which an error might occur. These participants may be more confident in their ability to handle the consequences of an error in a chosen context. The baseline groups rarely mentioned “testing things out”, likely in fear of triggering something they might not be able to control.

For **PROBLEM WITH THE EQUIPMENT**, the baseline groups were also more likely to talk about **Latency** than the other two. Slowness in an interface might be one the most shared experiences in **PROBLEMS WITH THE EQUIPMENT**. People with expertise in technology might however not just describe this slowness itself, but interpret further what is going on, as in *“it’s a file explorer that’s slow because there are loads of files. It’s just a web app that’s struggling.”** [P3_{W1}].

Discussing errors brings to light profound aspects of the relationship between the user and their tool. Self-confidence (**Depreciation**), for example, influences the user’s behavior upstream, before an incident even happens. An event will be more or less critical (**Criticality**) depending on the user’s relationship with the tool, the data, or their time in the given situation. If these elements are not important to them, the error is not critical for them. An error acts as a magnifying glass over existing bonds between the user and their task, and the user and their tool. They reveal the things that matter to the user.

The manner in which people speak about the system is also indicative of the nature of this relationship. The system is a fixed object fulfilling a function (**System**

rationality). In contrast, when described in error situations, the system may be attributed irrationality, emotions, or human-like intentions (**System personification**). It is interesting to see that, as much as people rationalize the system as a machine, they relate to it differently when things do not go as planned, potentially feeling betrayed. This type of relationship may explain why certain errors provoke emotions that are unrelated to the severity of their consequences, such as certain interaction interferences (see subsection 1.4.2).

Participants were generally prone to blame the system or the user while describing negative incidents. The perceived importance of finding who or what was responsible also characterized participants' proposed changes to the categorization. When they perceived that a CATEGORY encompassed events that they interpreted as opposed blame-wise, they were tempted to split it to mark that difference. This blame however depends on their perception of the event, and is not consistent throughout the types of phenomena (some INTERRUPTIONS are annoying, others are wanted) or even from one person to the next (if an interface is unclear, some people might blame themselves for not understanding it). For those reasons, blame is transversal to the categorization.

All the reported incidents corresponded to the definition of error in this thesis. If the code **Expectation/reality gap** was only tagged when that aspect was explicit in participants' speech, the concept transpired for every given *Incident_W*.

Overall, these workshops generated lively discussions, with participants bouncing ideas off each other and inspiring one another. A recurring piece of feedback was that the exercise "felt good", creating in some ways a sort of group therapy on interface problems.

People talked about incidents a posteriori, meaning that their narration was influenced by their knowledge of the outcome. Some reported user behavior such as **Lack of attention** or **Habit** are not problematic most of the time, but in the case of an error, can become a reason for the user to blame themselves. The results of this study would be interesting to confront with controlled experiments on error situations with a think-aloud protocol, or an electroencephalogram.

This study gave qualitative insights about the perception of error for different types of users. If they allow for a deep understanding, they cannot be taken as representative of the population. For this, a more quantitative approach is appropriate.

2.3 Study 2: User polling on precise error situations

Following the workshops, we set out an online survey to poll a large number of participants on the CATEGORIES. The objective was to uncover the material and emotional costs of incidents perceived by users. Can we draw tendencies from those responses, about demographics, CATEGORIES or examples?

2.3.1 Online survey

To complement the workshop study, which delved in depth into the experiences and opinions of a small number of participants, we took a more quantitative and global approach for the second study. To that end, we designed an online survey to investigate further the points raised by the workshops with a large number of participants. This survey had an English and a French version.

Structure

The goal of the survey was to collect data specific to context, people and tasks. We wanted to present participants with examples of every CATEGORY in our error taxonomy, as updated after the conduction of the workshops. However, we were also aware that a single example may accidentally misrepresent a whole family of errors. To circumvent that issue, we split all the workshop examples in each CATEGORY of errors into sub-categories, then selected two such *Sub-categories_Q* per category. Finally we selected three representative *Examples_Q* per *Sub-category_Q*, based on our own impression of understandability and universality (“is it likely to concern a majority of respondent?”). In some *Sub-categories_Q* (see table 2.17), we had to provide one or two additional examples to reach the required number. In others (**ERROR IN PLANNING**, **DARK PATTERNS₊** and **ACCESSIBILITY₊**) there were simply too few examples to build two *Sub-categories_Q*, so these categories had only one *Sub-category_Q*.

PROBLEM WITH THE EQUIPMENT

1) **Your device stops working completely.** For instance:

A_W- Blue screen (Windows) or black screen (Mac).

B_W- Your actions on the keyboard or mouse no longer have any effect.

C- Your smartphone turns itself off.

2) **You’re having trouble connecting to the Internet via a wifi hotspot.** For instance:

A_W- You can’t connect.

B_W- Your connection is unstable.

C_W- Your Internet access is abnormally limited.

ERROR IN PLANNING

The code you write does not give the desired result. For instance:

A- It performs operations in the wrong order.

B_W- It could be much more efficient.

C_W- It presents a logical error in relation to what you were trying to do.

WRONG USER INTERPRETATION

1) **A symbol or icon is unclear.** For instance:

A_W- It’s not the same icon as the one you’re used to.

B_W- The symbol strongly resembles another, that does something different.

C_W- The symbol is not clear and you don’t understand the information it conveys (microphone active/inactive, video paused/playing).

2) **You don’t understand a text or don’t understand it entirely.** For instance:

A_W- The explanation of backup options or personal data management is too long.

B_W- An error message is confusing.

C_W- An alert or error message is not explained in simple terms.

WRONG SYSTEM INTERPRETATION

1) **Speech recognition software misunderstands your speech.** For instance:

A_W- Its transcription of your words is incorrect.

B_W- It activates when you haven’t given the activation command.

C- It interprets other speech in the background as your instructions.

2) **A system interprets your action differently than you are used to.** For instance:

A_W- You click on a browser window currently in the background to bring it to the foreground. The browser, in addition to taking the focus, follows the link on which your click has ‘fallen’, without this having been your intention.

B- Your sideways scroll on a web page triggers the action *Previous page* or *Next page*.

C_W- The scrolling gesture is reversed.

INTERRUPTION

1) **You are distracted by an urgent message.** For instance:

A_W- A low battery notification

B- A notification from your calendar about a deadline

C_W- An imminent update that will shut down the computer unless the user takes action.

2) **You are distracted by a useless message.** For instance:

A_W- The information is not relevant to your current task.

B_W- The information is not urgent.

C_W- You would prefer not to be notified of this information at all.

INTERACTION INTERFERENCE

1) **A list is updated just before you click, causing the selection of an item normally higher than the one you were aiming at.** For instance:

A_W- In your email list, just after a new email is received.

B- In a chat, just after a new message has been received.

C- In a playlist, just after the current song ends.

2) **You're about to select an icon or link, but a new element appears just before you click.** For instance:

A_W- A pop-up window above a link you were about to click on.

B- A notification in the corner of your screen, above a tab you were about to close.

C_W- Alternatively, the notification you were about to click on disappears just before you click.

AIMED ACTION

1) **You unintentionally click or type in the wrong place.** For instance:

A_W- Just near a button or on a nearby button.

B_W- In the wrong part of a text you're trying to select.

C_W- On the wrong enemy (or even ally) in a game.

2) **You make a typing error on the keyboard.** For instance:

A_W- You use the wrong letter in a shortcut because your hand has shifted on the keyboard.

B_W- You type a letter close to the one you were aiming for on the keyboard.

C- You wanted to press Enter but pressed Delete instead.

SLIP

1) **You do one action instead of another in an inappropriate context, not because you didn't know the right action, but out of habit or lack of concentration.** For instance:

A_W- Ctrl+S in a web application that automatically saves changes.

B- Double-click on a hypertext link.

C- Touching a target on a non-touch screen.

2) **You perform an action by reflex before realizing that you didn't mean to do it.** For instance:

A_W- You close a window as soon as it appears, before realizing that you don't know what you've just closed.

B_W- You reflexively close a notification, without knowing what it contains.

C_W- You swipe to the next item/image/account on a social network, without realizing its content.

FAIL-UNSAFE

1) **You repeatedly refuse or postpone an update.** For instance:

A_W- It becomes mandatory at a time when you really need to use your computer.

B_W- The version of a tool is no longer compatible with recent files or formats.

C- Due to a lack of recent security updates, your computer has been compromised by a virus.

2) **You don't plug in your tool to account for your need for autonomy.** For instance:

A_W- You don't plug in your tool before an update.

B- You leave home with a weak smartphone battery.

C_W- You give a public presentation with a low battery.

OMISSION

1) **You skip one step.** For instance:

A_W- You send a message without its attachment

B- You forget to connect your headphones or speakers before listening to music

C_W- You forget to change layers before drawing on a different surface of your digital drawing.

2) **You forget to save.** For instance:

A_W- You close an application without having saved the latest changes to the last file edited.

B_W- You close a game without having validated a recent save point.

C_W- You forget to save an event in your schedule for a meeting that has just been decided.

DARK PATTERN₊

The system nudges you to act in a certain way. For instance:

A_W- Cookies are difficult to refuse.

B_W- A box must be unchecked to avoid receiving promotional emails.

C- A product is automatically included in your online shopping cart.

ACCESSIBILITY₊

The tool or platform is not adapted to your disability. For instance:

A_W- There is no accessibility option.

B_W- The accessibility option is faulty or poorly designed.

C- There are some accessibility options, but not the one you need.

TABLE 2.17: Questionnaire *Sub-categories_Q* (in bold) and *Examples_Q* (A, B, C for each *Sub-category_Q*). The *Examples_Q* taken from the workshops are marked with a W, others were filled out by the researchers.

As one can see in table 2.17, each *Sub-category_Q* is detailed with 3 *Examples_Q*. For **PROBLEM WITH THE EQUIPMENT** for instance, the *Sub-category_Q* 1 (“Your device stops working completely”) has 3 *Examples_Q* A, B and C (“A- Blue screen (Windows) or black screen (Mac)”, “B- Your actions on the keyboard or mouse no longer have any effect” and “C- Your smartphone turns itself off”).

We investigated 12 CATEGORIES of errors with 21 *Sub-categories_Q* in total. Each featured numerous questions regarding errors’ context, task when they happen, emotions felt and consequences.

For the questionnaire to be feasible, we chose first to allow participants not to go through every *Sub-category_Q*: each participant could encounter only one *Sub-category_Q* per CATEGORY, selected at random. Because **ERROR IN PLANNING**, **DARK PATTERNS₊** and **ACCESSIBILITY₊** only feature one *Sub-category_Q*, all participants would see the same one for those CATEGORIES.

Second, even with one *Sub-category_Q* per category, the whole questionnaire still consisted in 369 questions¹⁴. We could not reasonably expect online participants we did not know, and without compensation, to fill such a questionnaire entirely. We therefore offered participants to end the questionnaire every time they finished responding to an *Sub-category_Q*’s questions. In practice, at the end of every *Sub-category_Q*’s set of questions, participants had a question saying “There are still N types of examples left on our list. Would you like to move on to the next one?”¹⁵, to which they could respond Yes or No. That way, a person could more or less control the time they wanted to spend on the survey.

¹⁴9 demographic questions + (13 questions on the incidents + 17 questions on feelings and emotions)

* 12 categories (not including the “There are still N types of examples left on our list. Would you like to move on to the next one?”).

¹⁵where $0 < N < 12$

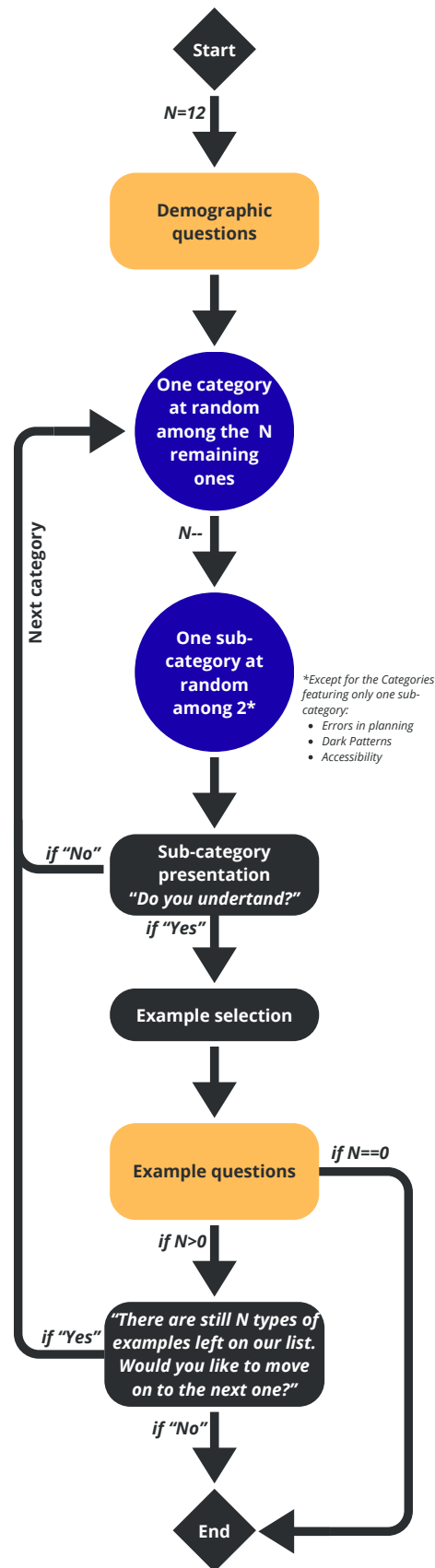


FIGURE 2.5: Questionnaire process diagram

In order to reduce the order effects and to balance the number of responses per CATEGORY's *Sub-category_Q*, the order of the CATEGORIES was randomized for each participant.

A diagram of the questionnaire procedure is presented in Figure 2.5. The survey was long to complete but did not require participants to complete it whole, instead, its implementation ensures a viable and tunable completing time while balancing the number of responses gathered for each CATEGORY and *Sub-category_Q*.

Demographic questions

Each participant was first asked their age and gender identity. Because we questioned them on vocabulary (what words they would use to describe a *Sub-category_Q*, see table 2.19 below), we also asked to indicate their mother tongue.

We asked as well for the use frequency of their different devices, their reported confidence in using them, digital technology in general, and in handling a new digital tool.

We were also interested in studying the potential effects of profession and education on error perception, so they entered their activity and study level.

Survey questions	Response type
What is your age?	Positive integer
What is your gender identity?	Single choice: Man, Woman, Non-Binary, Does not want to answer, Prefer to self-define (Free text)
Enter your mother tongue(s) separating them with commas (American English, British English, German, Japanese, Korean, Mandarin, Italian, Spanish...)	Free text
What's your profession?	Free text
What is your level of education?	Compulsory schooling up to age 16 Maximum 1-year training after secondary school CAP, BEP, <i>BTEC National Diploma</i> Baccalauréat, <i>High-School Diploma</i> DEUG, BTS, DUT, DEUST, <i>12th grade, Two-year university degree in technology</i> Licence, Licence LMD, licence professionnelle, Maîtrise, <i>Bachelor's degree</i> Master, DEA, DESS, diplôme d'ingénieur, <i>Master's degree</i> Doctorat, habilitation à diriger des recherches, <i>PhD and up</i> Other (Free text)
How frequently do you use these digital devices? PC, Laptop, Tablet, Smartphone and Virtual reality system	For each device, single choice: Up to once a month Up to once a week Up to once every two/three days Up to once a day Several times a day
How would you rate your general level of knowledge and expertise regarding digital technologies?	Likert scale: 1-Complete neophyte 2-Novice 3-Capable/normal 4-Competent 5-Expert

How would you rate your confidence, on average, in using the digital tools you use regularly (at least once a week)?	Likert scale: 1-No confidence 2-Limited confidence 3-Moderate confidence 4-High confidence 5-Total confidence
In general, do you feel you can easily handle new digital tools (software, devices), or new functions in your usual tools?	Likert scale: 1-No confidence 2-Limited confidence 3-Moderate confidence 4-High confidence 5-Total confidence

TABLE 2.18: Demographic questions at the beginning of the survey.
The English equivalents to the French diplomas are in italics.

Questions about each incident

After demographic questions, participants were presented *Sub-categories_Q* one by one, one for each CATEGORY, in a randomized order.

First, we asked whether or not they understood the *Sub-category_Q* described. If so, they could move one to the other questions, otherwise, they were sent to the next *Sub-category_Q* in random order.

Vocabulary We were interested in the lexicon one might use to describe different types of errors. To that end, participants could select words within a selection to characterize the presented *Sub-category_Q*, visible in table 2.19. Those words were chosen carefully according to the linguistic analysis presented in chapter 1, sub-section 1.1.1. As a reminder, in the study of the synonyms of the word “error”, I extracted 6 themes standing out (see table 1.3). The words presented in the questionnaire were selected to cover those 6 themes, both in English and in French. The table 2.19 shows the choice of words given to participants. Emphasis was put on mirroring the *words* from one language to the other, rather than the *codes*, which explains that the codes can differ in the translation.

Ideally, we wanted participants to report their memories and impressions relating to an incident in particular. However, even when recognizing an event that they had already experienced, it is not always easy for people to remember a specific occurrence. For this reason, we gave participants the option for to select between referring to a specific incident, or assessing a type of incident that happened to them in general. The presence of *Examples_Q* could also facilitate participant recollection. The following questions were adapted accordingly (see table 2.20). When the participant could refer a specific occurrence, we questioned them about the context and task at play when the incident occurred.

English	French
Error	Erreur
Deception [DEC]	Duperie [DEC]
Slip/Miss [SKI]	Maladresse [SKI]
Fault [FLA]	Faute [STR]
Breakdown [CON]	Plantage [CON]
Flaw [FLA]	Défaut [FLA]
Blunder [SKI]	Bêtise [SKI]
Confusion [INT]	Confusion [INT]
Mistake [STR]	Gaffe [SKI]

TABLE 2.19: Words given to participants of the questionnaire to characterize each *Sub-category_Q*. The codes are described in table 1.3. It is of note that the words in English are not always equivalent to the words in French.

Overall, we questioned participants on the interest points raised by the workshops: their perceived expertise, confidence and familiarity with the context of an error occurring.

Survey questions	Response type
Do you understand the above examples?	Single choice: Yes, Kind of, Some not all, No ¹⁶
In your opinion, which of the following words best describe the general idea conveyed in the examples above?	Multiple choice between words in table 2.19 + Other: [free text]
<i>Example and sub-examples description</i>	
What do you remember?	Single choice: 1- You can remember a specific example. It may be the most recent case, or an example feel is particularly representative of these incidents. (SPEC) 2- You cannot remember a specific example, but you can think of this incident in general. (GEN) 3- You cannot think of an instance of any of these examples ever occurring to you (skip to NEXT)
SPEC&GEN Please choose the letter that corresponds to the specific incident or kind of incident that you remember. If the incident you are thinking about differs from the examples above (A, B, C), please explain the difference in the comment field.	Single choice: A, B, C + Free text (corresponding to the 3 Examples _Q that we selected for each Sub-category _Q)
SPEC In the following questions, we call “incident” the precise situation you thought about in the What do you remember? question. GEN In the following questions, we call “incident” the type of general situations you thought about in the What do you remember? question.	
SPEC&GEN Regarding the example you chose, in your own experience: how often does this incident, or a similar incident, happen to you?	Single choice: Never/Not Applicable, Exceptionnally, Up to once a month, Up to once a week, Up to once every 2 or 3 days, Up to once a day, Several times a day
SPEC When did that particular incident occur?	Single choice: More than one month ago, More than a week ago, This week, Yesterday, Today, Don’t know / remember
SPEC&GEN Is this the most recent instance of this type of event?	Single choice: Yes, No
SPEC&GEN On which device(s) does this type of events occur most frequently?	Multiple choice: Personal computer, Laptop, Tablet, Other [free text]
SPEC How would you rate your own level of expertise in the specific context in which this incident occurred? (device, software, task, etc.)	5 points Likert scale: 1-No expertise, 2-Novice, 3-Intermediate, 4-Proficient, 5-Expert
SPEC Just before the incident occurred, did you think you were capable of carrying out most of the tasks you needed ?	5 points Likert scale: 1-Not at all, 2-Little confidence, 3-Medium confidence, 4-High confidence, 5-Absolutely
SPEC&GEN How would you rate your capacity to correct, manage, or fix that incident?	Single choice: Entirely incapable, Partially capable, Entirely capable

¹⁶Only the choice of “No” here would not lead to the rest of the questions. We did not want to exclude participants if they did not understand all Examples_Q.

SPEC Regarding the context in which this incident occurred (i.e. task, environment, tools, device, software, etc.):	
SPEC Regarding the context in which this incident occurred (i.e. task, environment, tools, device, software, etc.): Is this context usual to you?	5 points Likert scale: 1-Not at all, 2-A little, 3-Moderately, 4-Very, 5-Extremely
SPEC Regarding the context in which this incident occurred (i.e. task, environment, tools, device, software, etc.): In general, how comfortable are you with this context?	5 points Likert scale: 1-Not at all, 2-A little, 3-Moderately, 4-Very, 5-Extremely

TABLE 2.20: Questions about the type of incident. The questions marked with SPEC only appear when the participant selected a specific example to describe, the questions marked with GEN only appear when they elected to rate the type of incident in general.

Questions about feelings and emotions

Participants then answered questions about their feelings and impressions about the selected *Example_Q*. These questions were directly inspired from the codes extracted from the workshops, and are presented in table 2.21.

Survey questions	Response type
We will now ask you some questions about how you felt during and after this incident.	
SPEC&GEN What are the consequences of that event?	Multiple choice: Waste of time, Loss of data, Damaged hardware, Loss of money, Damaged self-confidence, Loss of credibility, Loss of confidence in the tool, No specific consequences, "I don't remember", Other + [Free text]
SPEC&GEN You can elaborate on those consequences in you own words here:	Free text
SPEC&GEN Are the consequences of that incident serious?	5 points Likert scale: 1-Not at all, 2-A little, 3-Somewhat, 4-Very, 5-Extremely
SPEC&GEN Do you consider this incident is the fault of someone or something?	For each: Yourself, The system, An input device (mouse, keyboard, controller, etc.), An exterior element (person, distraction, etc.), The system's designer: No, Maybe/Partially, Yes
SPEC About the task you were performing, was it:	Multiple choice: The end of a long wait, An interesting task, A chore, The end of a long project, An important task, A boring task, A mundane task, Other: [Free text]
SPEC&GEN Do you think this incident is intentional on the part of the system or its designer?	5 points Likert scale: 1-Not at all, 2-Unlikely, 3-Maybe, 4-Likely, 5-Certainly
SPEC&GEN Regarding your emotions after the incident occurs: Is the question of "who or what is responsible for this incident" important to you?	5 points Likert scale: 1-Completely disagree, 2-Rather disagree, 3-Neither agree nor disagree, 4-Rather agree, 5-Completely agree

SPEC&GEN Regarding your emotions after the incident occurs: Do you feel guilt for the way the incident played out?	5 points Likert scale: 1-Completely disagree, 2-Rather disagree, 3-Neither agree nor disagree, 4-Rather agree, 5-Completely agree
SPEC&GEN Regarding your emotions after the incident occurs: Do you feel frustration as a result of this incident's consequences ?	5 points Likert scale: 1-Completely disagree, 2-Rather disagree, 3-Neither agree nor disagree, 4-Rather agree, 5-Completely agree
SPEC&GEN Regarding your emotions after the incident occurs: Do you feel frustration as a result of this incident occurring ?	5 points Likert scale: 1-Completely disagree, 2-Rather disagree, 3-Neither agree nor disagree, 4-Rather agree, 5-Completely agree
SPEC&GEN Talking about the odds of this incident occurring: Do you think this incident is predictable?	5 points Likert scale: 1-Completely disagree, 2-Rather disagree, 3-Neither agree nor disagree, 4-Rather agree, 5-Completely agree
SPEC&GEN Talking about the odds of this incident occurring: Would you say that this incident comes about through bad luck?	5 points Likert scale: 1-Completely disagree, 2-Rather disagree, 3-Neither agree nor disagree, 4-Rather agree, 5-Completely agree
SPEC&GEN Does this incident surprise you when it occurs?	5 points Likert scale: 1-Not at all, 2-A little, 3-Moderately, 4-A lot, 5-Extremely
SPEC&GEN Would you say that, one way or another, you deserve this incident to happen?	5 points Likert scale: 1-Completely disagree, 2-Rather disagree, 3-Neither agree nor disagree, 4-Rather agree, 5-Completely agree
SPEC&GEN Would you say that you could prevent that incident?	5 points Likert scale: 1-Completely disagree, 2-Rather disagree, 3-Neither agree nor disagree, 4-Rather agree, 5-Completely agree
SPEC Keeping aside the incident itself, would you say that what you were trying to achieve was difficult?	5 points Likert scale: 1-Very easy, 2-Rather easy, 3-Moderately easy, 4-Rather difficult, 5-Very difficult
SPEC&GEN Are there any other emotion or thought that you had during or after that event, that you would like to share here?	Free text
NEXT There are still <i>N</i> types of examples left on our list. Would you like to move on to the next one? If not, we'll just ask you one more question.	Single choice: Yes, I can see another example, No, I want to be done with this questionnaire

TABLE 2.21: Questions about a specific occurrence of the selected *Example_Q*. The questions marked with SPEC only appear when the participant selected a specific example to rate, the questions marked with GEN only appear when they elected to describe the type of incident in general (see table 2.20).

Because an idea of **Criticality** was present in the workshops, we asked about the seriousness of the incident. Questions about the assignment of fault, responsibility, guilt, preventability and deservedness stemmed from the codes **Blaming the system** and **Blaming the user**. We also instigated about the detection of **Dark Pattern** with the assignment of a system intention. **Irritation** was also an important code, which we investigated in the questionnaire. In the same way, there were questions about **Bad Luck** causing the incident, and about the **Surprise** that could be felt.

Some questions also focused on the task that was being carried on when the incident happened: was it important, was it difficult? This survey was interested in the stakes present when an error happens, and how they impact the criticality, emotional and material, of an incident.

Participants' recruitment

The questionnaire was first shared using our personal and professional networks. The communication department of our research center shared it internally and externally too.

We also shared the link on Reddit, in subreddits about computer questions and surveys in general.

Finally, we shared the questionnaire with a number of structures in an attempt to diversify our respondents pool. We contacted 18 different associations in France (mostly around Lille) that aim to facilitate digital devices access and use. To get more senior participants, we reached out to 3 retirement homes and 1 association helping seniors with computer and smartphone use. Finally, we sent the questionnaire link to an association working to aid people with disabilities in the access of new technologies.

2.3.2 Results

Participants

133 people responded to at least one full set of questions corresponding to a *Sub-category_Q* in the survey. They were aged 18 to 79 years old, with a mean of 33.4 and a standard deviation of 12.7. Genders were very balanced, with 64 men, 65 women, 3 non-binary people and 1 self-defined.

25 people took the survey in English, among which 10 were English native speakers. In French, we had 108 respondents, and among them 101 French native speakers.

Most participants held a Master's degree equivalent or up (72.2%, Figure 2.6).

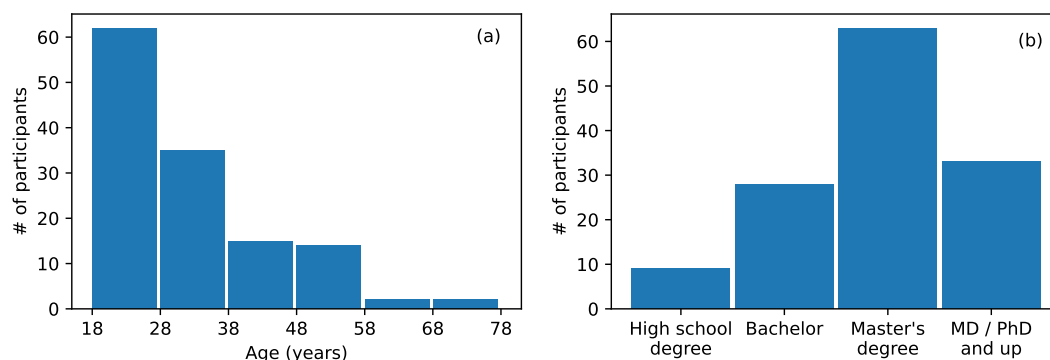


FIGURE 2.6: Participants' age groups (left) and last diploma (right).

A very vast majority of participants used personal computers (94.7%) and smartphones (99.2%) everyday. Most of them declared themselves highly (42.1%) or totally (44.4%) confident using their usual digital tools. Fewer were totally confident (30.8%) about handling *new* digital tools.

Participants ranked their level of expertise regarding digital technologies rather uniformly across “Capable/normal”, “Competent” and “Expert”.

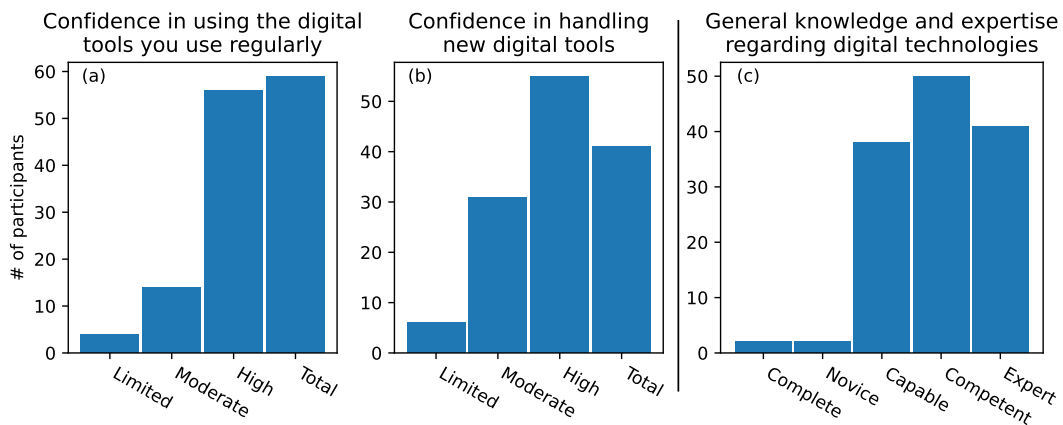


FIGURE 2.7: Number of participants for each level of (left) confidence in using their usual tools, (middle) confidence in handling new tools, and (right) overall expertise in handling digital technology.

Participants could stop the survey after every example, so they did not complete the same number of questions. Figure 2.8 shows this distribution, for participants who answered the whole question set of at least one category.

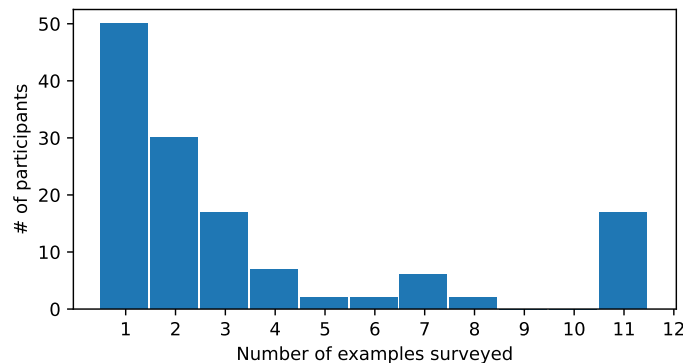


FIGURE 2.8: Participant count for every number of answered examples

Data Pre-processing and Analysis Approach

The data gathered in this questionnaire is entirely nominal (categories, labels, etc.) or ordinal (age groups, Likert scales). For that reason, we analyzed them using non-parametric tests on value ranks: significant effects are explored using the Kruskal-Wallis test, and significant differences (post-hoc tests) are assessed using the Dunn test with Bonferroni correction. Effect sizes are reported as epsilon-squared (ϵ^2), as recommended in [145], and interpreted as per practice:

When reporting the details of a posthoc test inline, the direction of significant differences are explained using the means of Likert scale answers, even though the tests were calculated on rank sums; that is because other statistics, like median or mode, will often be equal between two levels of independent values—even if there exists a statistically significant difference between their rank distributions.

0.00 < 0.01	Negligible
0.01 < 0.04	Weak
0.04 < 0.16	Moderate
0.16 < 0.36	Relatively strong
0.36 < 0.64	Strong
0.64 < 1.00	Very strong

TABLE 2.22: Interpretations of the ϵ^2 effect sizes, as prescribed in [145].

Overall, the reader will note that all significant effects that we observed *across all error CATEGORIES at once* had small effect sizes. There are also, generally, a lot fewer significant effects detected across all CATEGORIES than within individual CATEGORIES, reinforcing the intuition that “error” in itself is not a concept adapted to this type of analysis as a whole, and that it needs to be broken down into specific phenomena.

In what follows, we choose to only discuss significant effects with effect sizes “Moderate” or above, and that lend themselves to linear interpretation: for instance, with an ordinal independent variable V with three levels $A < B < C$, if there exists a significant difference between levels B and C , but not between A and B or A and C , then we refrain from inferring about the respective ranks of V . Variable levels with less than 10% of the total sample are ignored (*e.g.* if level).

Lastly, the results on the vocabulary questions reported here include all responses, both in French and in English. While no synonyms can be exact per se from one language to another, we made sure to choose as close translations as possible (see table 2.19). This limitation must be taken into account when examining and generalizing these results.

General correlations with demographics

When exploring how demographic parameters influenced responses in general (regardless of categories), we found few effects that passed the criteria above.

The older participants were, the more likely they were to fault an exterior element for an incident happening. Age had a significant moderate effect on faulting an exterior element for the incident being discussed ($H = 45, p < 0.0001, \epsilon^2 = 0.07$). Age groups 18-28 (mean 0.3), 28-30 (0.38), and 38-48 (0.27) all faulted external elements less than the 48-58 group (mean 0.86, all $p < 0.005$). Age groups older than 48-58 all had fewer than 3 respondents, not enough to allow such a comparison.

General observations

Differences between workshops and questionnaires Accross all reported incidents, the most reported consequence was “Waste of time”. However, while it existed as a code in my analysis of the workshop transcripts, it was far from being the most frequently mentioned consequence. We can imagine that when presented with a fixed list of potential consequences, most people will decide that any error cost them time, but it is not the primary consequence that they will volunteer when prompted to simply describe an incident’s consequences.

A “loss of confidence in the tool” was a rather common reported consequence in the questionnaire, when it was not recurrent enough to be observed as a code in the workshops. Similarly to above, this loss of confidence, while present, may not come up when talking about an incident. There is also the possibility that this consequence may be more long term, over the repetition of said incident, and that answers to the workshops were more event-specific than the experiences shared in the questionnaire.

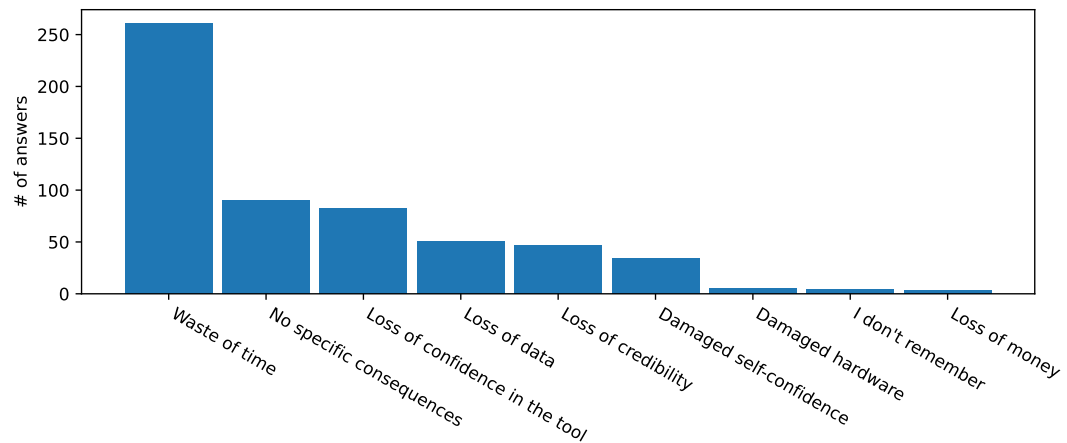


FIGURE 2.9: Number of incidents for every consequence (participants could tick more than one per incident)

About attribution of cause and blame, the workshops and the questionnaire also presented different tendencies. In the workshops, people expressed blame for the system more than for the user, when in the questionnaire, they more often blamed themselves than any system component (see Figure 2.10). We can imagine that the collective nature of the workshop discussions might have encouraged people to “rally” against the system in general. Sharing their experiences and realizing their shared nature may have further reinforced their criticism of the system, whereas the questionnaire was an individual exercise.

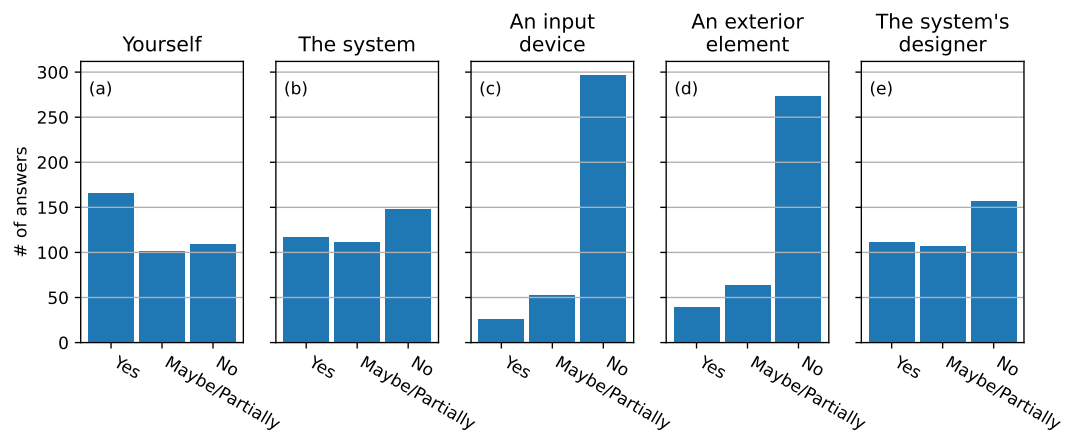


FIGURE 2.10: Number of responses for every response of “Do you consider that this incident was the fault of someone or something?”

Possibly for a similar reason, questions about guilt or the feeling of deserving incidents generally had low scores in the questionnaire (see Figure 2.11), while **Depreciation** was a recurrent code in the workshops. People might have similarly encouraged each other collectively to express such judgements in the workshops. On the other hand, answering a questionnaire by themselves, it is possible that participants further put this feeling into perspective.

Context of the incidents For all categories, participants mostly reported the most recent incident of the example (91% overall, mean 88% over each CATEGORIES).

When the example they chose was a precise event in their past, as opposed to a general notion of the described incident, participants describe having been familiar and comfortable with the task at hand (see Figure 2.12). This tendency is particularly clear for **INTERRUPTION**, **OMISSION** and **AIMED ACTION** (see below).

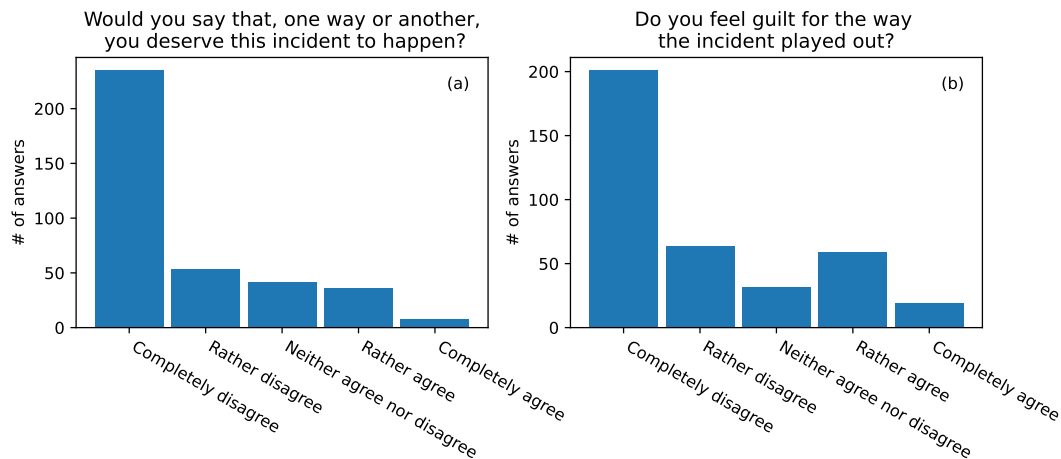


FIGURE 2.11: Participants assessments of whether they thought they deserved the incident to happen (left), and whether they felt guilty (right).

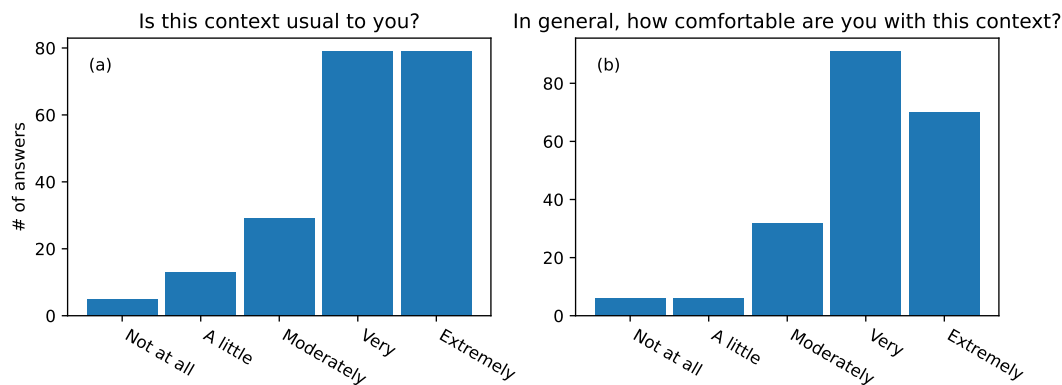


FIGURE 2.12: Number of situations for every level of habit (left) and comfort (right) in the context

Overall, the incidents on which participants based their answers occurred majoritarily during mundane tasks, and during interesting or important tasks (Figure 2.13). This is to be interpreted carefully, as we do not know the general ratios of these task types (with or without incidents) in daily life.

Categories results

Questionnaire results drew a portrait of each CATEGORY, as well as some tendencies among certain participants responses.

PROBLEM WITH THE EQUIPMENT Respondents could generally provide answers in this group: 60% could recall a specific example, 38% answered from a general idea of the CATEGORY, and only 2% could not answer. This shows that the presented *Examples_Q* were familiar events. In the perception people have about incidents in this category, it appears that “break-down”, “flaw” and “error” were the preferred terms to describe them (see Figure 2.14 -a).

Those *Examples_Q* mostly happened exceptionally (73%). People could select several common devices on which **PROBLEMS WITH THE EQUIPMENT** happened, and chose laptops (selected 68% of the time), smartphones (55%) or PCs (23%) for the most part. Reported incidents tend to happen during any kind of task, but rarely with time stakes: only about 6% of people selected “The end of a long wait” or “The end of a long project” as a task, as shown

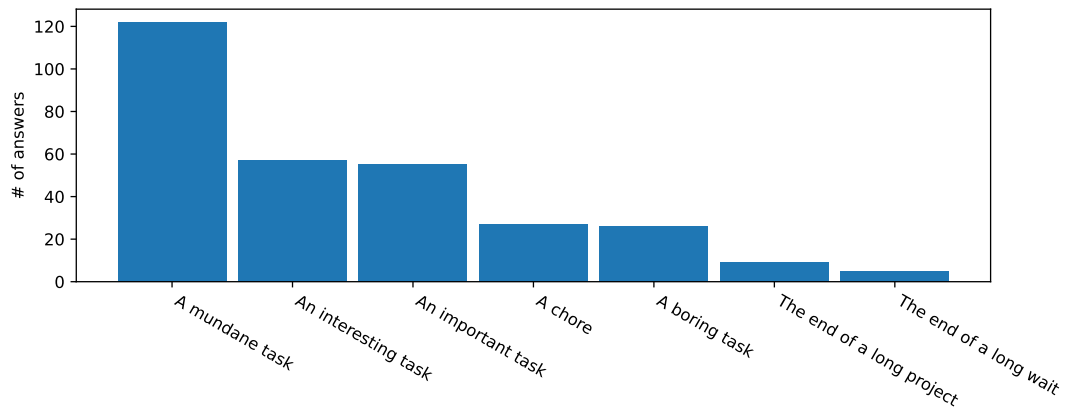


FIGURE 2.13: Types of tasks during which the reported incidents occurred.

of answers in **PROBLEM WITH THE EQUIPMENT** incidents

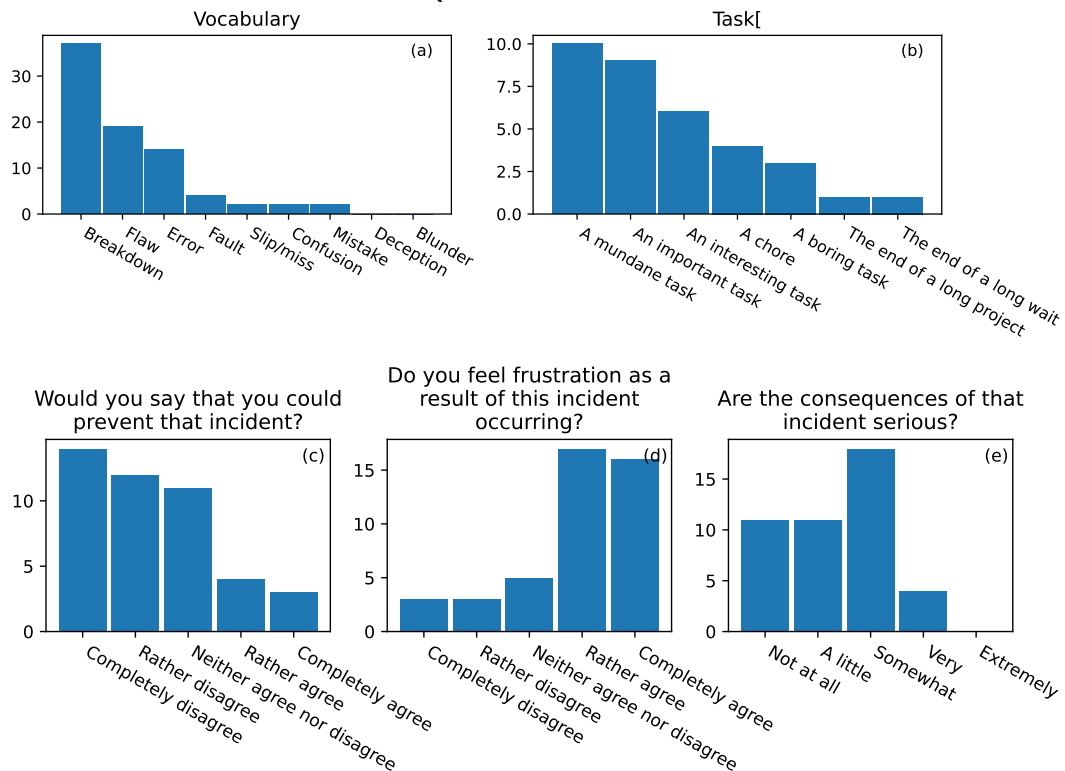


FIGURE 2.14: Selected answers for **PROBLEMS WITH THE EQUIPMENT**.

in Figure 2.14 -b. Most of them were somewhat unpreventable according to the participants: on Figure 2.14 -c we see that most responses are on the left side of the Likert scale.

As a result of the incident, people did not feel much guilt; 66% of participants selected “Completely disagree” (the far end of the Likert scale) for this question. They tended to feel frustration, as one can see, most of the Likert responses are on the right side of the Figure 2.14 -d.

In terms of consequences, 86% of people experienced a waste of time, 39% a loss of confidence in the tool, and 27% a loss of data from **PROBLEMS WITH THE EQUIPMENT**. Those consequences were rated from “Somewhat-” to “Not at all serious” (see Figure 2.14 -e). However, they created frustration (43% of “Rather-” and 36% “Completely agree”).

64% of people did not blame themselves, and 59% blamed the system (not including people who responded “Maybe” to those questions). 86% did not think that they deserved the incident to happen at all (selected “Completely disagree”).

ERROR IN PLANNING People remembered 40% of the time a general memory of **ERRORS IN PLANNING**, but 25% also did not relate. Considering the *Sub-category_Q* that participants were presented, relating to programming, it makes sense that a lot of them did not recollect such an event. As shown in Figure 2.15 -a, people mostly associated the word “error”, but also chose other words regularly (except for “Deception”).

of answers in **ERROR IN PLANNING** incidents

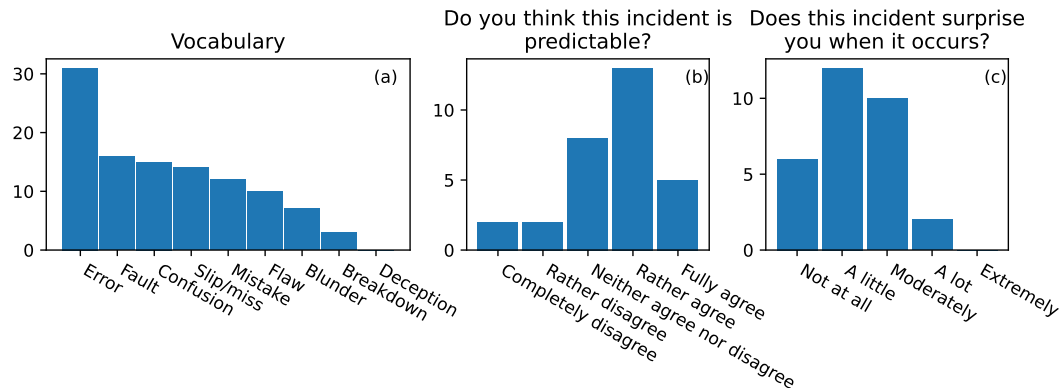


FIGURE 2.15: Selected answers for **ERRORS IN PLANNING**.

There was no noticeable tendency regarding frequencies or moments of last occurrence. *Examples_Q* were reported to happen mostly on laptops (73%) and PCs (67%), which is the tool people usually use for programming. The task was usually interesting (selected 67% of the time), important (38%) and/or mundane (19%). For participants, the event generally did not come through bad luck (with 53% of participants having selected “Completely disagree” to that question), nor was it intentional (83% “Not at all”). They also thought it to be somewhat predictable, as we see that Likert responses are mostly on the right side of Figure 2.15 -b.

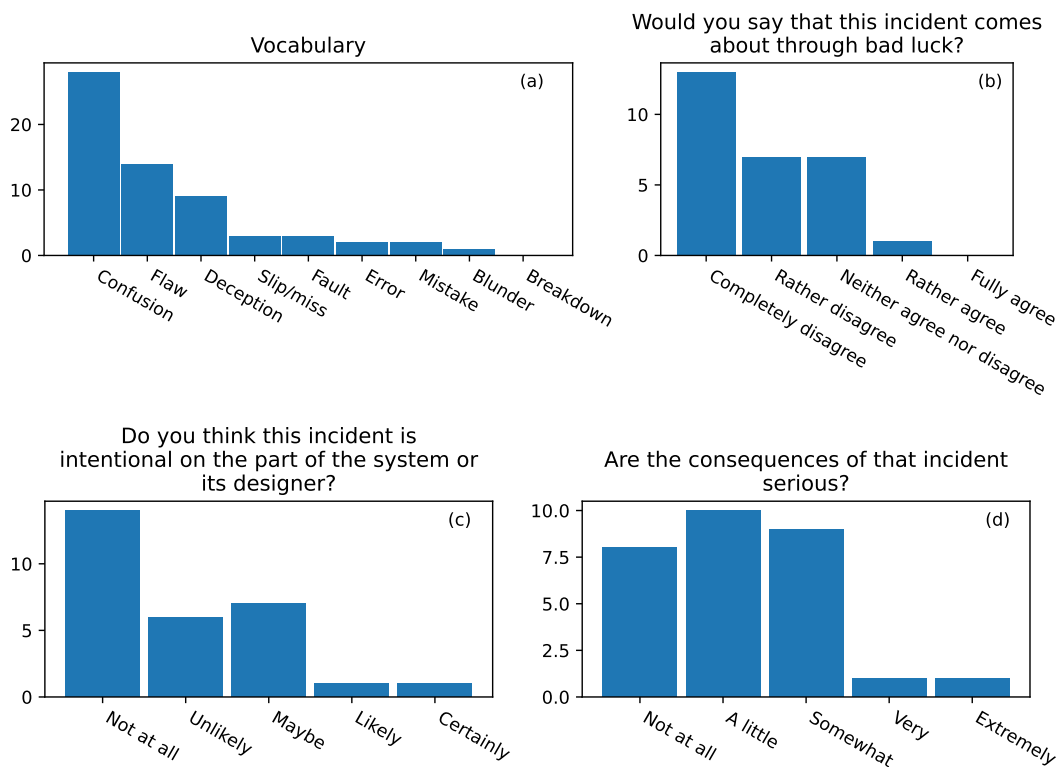
In terms of emotions on the moment, participants were in majority a little or moderately surprised, as one can see on Figure 2.15 -c.

People experienced a waste of time 90% of the time, a damaged self-confidence 30% of the time and a loss of credibility 20% of the time. These consequences were never rated as extremely serious. We found a strong effect size of confidence in handling new tools on the reported seriousness of consequences ($p < 0.01$, $\varepsilon^2 = 0.41$). People totally confident reported less serious consequences ($p < 0.05$) than people with a high confidence, who themselves reported less serious consequences ($p < 0.01$) than people with moderate confidence. People felt that they were either partially (53%) or entirely (47%) capable of fixing the incident.

90% of people blamed themselves. We also found a strong effect of confidence in using one’s usual tools on self-blame ($p < 0.005$, $\varepsilon^2 = 0.44$). People with high confidence faulted themselves more ($p < 0.01$) than people with moderate confidence. There is however not a clear shared positive response for the feeling of having deserved the incident. Most participants did not blame the system (63%), nor the designer (80%), nor an input device (90%), nor an external element (83%). We found a strong effect of confidence in using one’s usual tools on input device blame ($p < 0.001$, $\varepsilon^2 = 0.42$). Totally confident people and people with high confidence were both faulting an input device less than people with moderate confidence – respectively $p < 0.01$ and $p < 0.005$.

WRONG USER INTERPRETATION For this CATEGORY, 56% of people could gather a general memory of an *Example_Q* happening to them, and 25% a specific occurrence. Participants described *Examples_Q* as “confusions”, “flaws” or “deceptions” mostly (see Figure 2.16 -a).

There was no strong tendency regarding frequencies, moments of last occurrence or type of tasks. For participants, these incidents did not come about through bad luck: they mostly

of answers in **WRONG USER INTERPRETATION** incidentsFIGURE 2.16: Selected answers for **WRONG USER INTERPRETATION**.

selected the left side of the Likert scale in Figure 2.16 -b. Observing Figure 2.16 -c, they were also unlikely to be intentional.

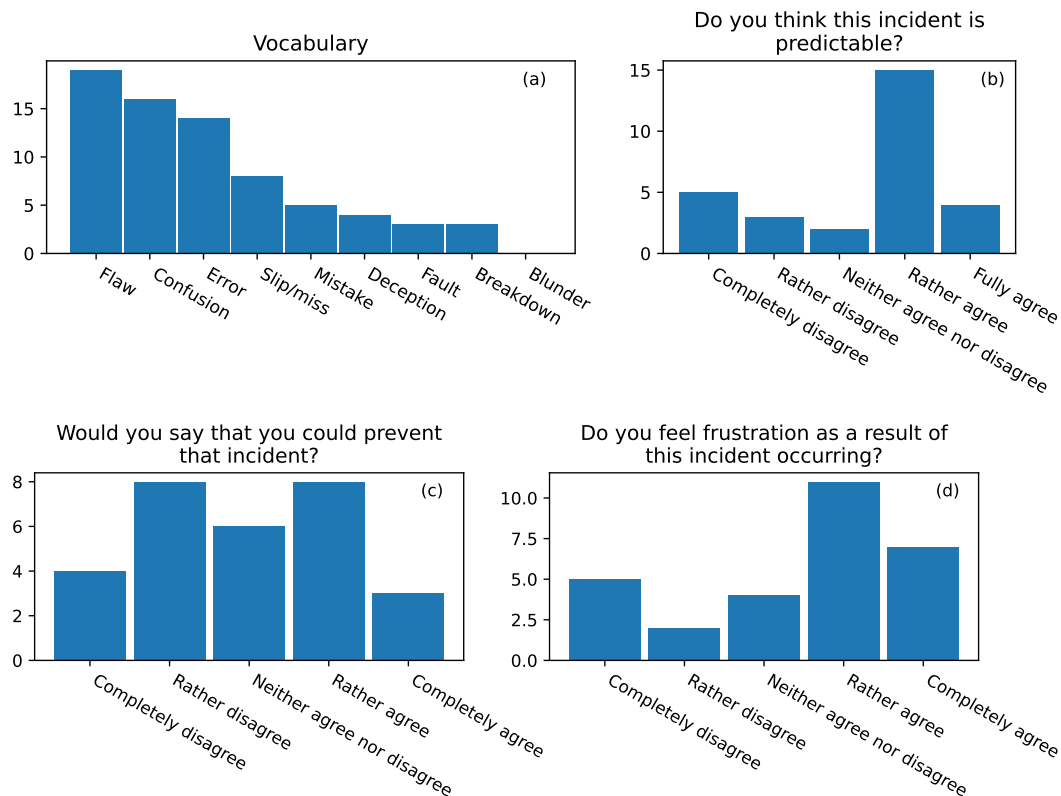
When the incident happened, the respondents reported to be somewhat surprised, with 45% of them reporting "A little". 66% of participants did not feel guilt ("Completely disagree"), but most of them felt some frustration: 62% "rather" agreed with feeling frustrated, and 24% "completely".

The most reported consequences were a waste of time (83%), a loss of confidence in the tool (45%) and a loss of data (21%). Consequences were somewhat to not at all serious for the most part (see Figure 2.16 -d). 48% of people "rather agree" with feeling frustration at those consequences.

People blamed the system designer 55% of the time. 83% disagreed with blaming an exterior element and not one participant blamed an input device. For blaming oneself, we found a relatively strong effect size of confidence in using one's usual devices ($p < 0.001, \epsilon^2 = 0.24$). People with total confidence faulted themselves more ($p < 0.05$) than people with high confidence. 66% thought they did not deserve the incident to happen (selecting "Completely disagree").

WRONG SYSTEM INTERPRETATION For this category, 41% of participants remembered a specific occurrence and 38% a general memory. To describe *Examples_Q*, they preferred the words "flaw", "confusion" and "error" (see Figure 2.17 -a).

The events were of varied frequencies, but mostly happened exceptionally (38% of the time). There was no strong tendency in terms of last occurrence. Participants reported incidents to mostly happen on smartphones (selected 55% of the time) and laptops (55% as well). The task was in fact mundane 87% of the time, and "very easy" in 60% of cases. Before the incident happened, when talking about specific occurrences, people were highly (47%) or absolutely (53%) confident they could carry out their task. They disagreed the incident came through bad luck (34% responded "rather disagree" and 38% "completely disagree"). 48% thought the incident to be "not at all" intentional on the part of the system or its designer,

of answers in **WRONG SYSTEM INTERPRETATION** incidentsFIGURE 2.17: Selected answers for **WRONG SYSTEM INTERPRETATION**.

and 28% thought it “unlikely”. They however did not clearly align on whether it was predictable or preventable, as one can see in Figure 2.17 -b and -c.

When the incident happened, people did not feel guilty (79% “completely disagree”), but they felt rather frustrated (see Figure 2.17 -d).

The most common consequences were a waste of time (reported 66% of the time) and a loss of confidence in the tool (38%). They were perceived for the most part “not at all” (59%) serious. In 76% of cases, participants felt “entirely capable” to correct, manage or fix the incident.

Participants blamed the system 90% of the time for the incident happening, and 79% did not blame an exterior element. 66% of participants “completely disagreed” with deserving the incident to occur.

INTERRUPTION 54% of participants remembered a specific *Example_Q* of an **INTERRUPTION**, and 38% a general idea of the phenomenon, which shows its familiarity. There was not a clear commonality in the vocabulary (see Figure 2.18 -a, except for a light preference for the words “flaw”, “confusion” and “slip/miss”). The question of responsibility also did not present a shared common answer, but the participants were generally opined: they dismissed the middle of the Likert scale in Figure 2.18 -b.

There was no strong tendency in frequencies or temporal proximity of the incident. The events mostly happened on laptops (ticked 71% of the time) and smartphones (62%). When reporting specific incidents, the contexts were usually very familiar and comfortable (45% of “extremely usual” and “extremely comfortable” responses). 60% of people were “absolutely confident” about their task before the incident happened. Those tasks were mostly mundane (75%), but sometimes important (25%) and/or interesting (20%). Participants thought the event did not come through bad luck (35% “rather” and 44% “completely disagree”) and was predictable (35% “rather” and 41% “fully agree”).

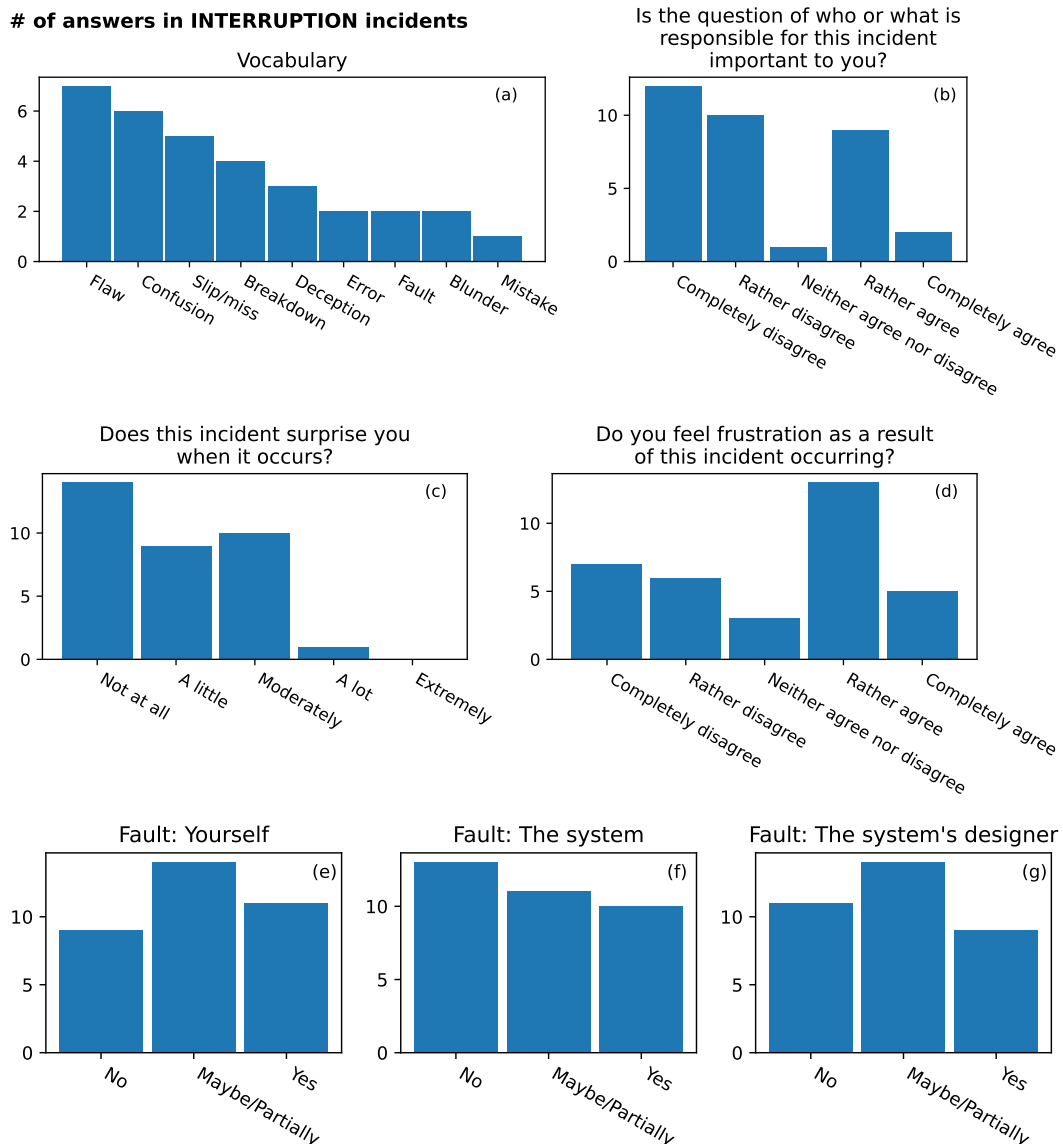


FIGURE 2.18: Selected answers for INTERRUPTIONS.

Interestingly, **INTERRUPTIONS** did not seem to provoke much surprise, as Figure 2.18 -c shows. It also did not induce guilt (62% “completely disagree”), and frustrations reports were varied (see Figure 2.18 -d).

The reported consequence were mostly a waste of time (65%), or no specific consequences (41%). These consequences were usually “not at all” (50%) serious or “a little” (26%), and 71% of respondents felt “entirely capable” to fix the incident. Those consequences felt either not frustrating at all (32%) or rather frustrating (44%).

Participants did not particularly blame themselves, the system or the designer (see Figure 2.18 -e, -f and -g). They never blamed the input device. 53% “completely disagreed” with deserving the incident to happen.

INTERACTION INTERFERENCE For this category, 51% of people could remember a general idea of the incident, and 33% a specific *Example_Q*. In Figure 2.19 -a, one can see that there was no strong tendency for the choice of vocabulary, with a slight preference for “flaw”, “deception” and “confusion”. An **INTERACTION INTERFERENCE** was never described as a “fault”.

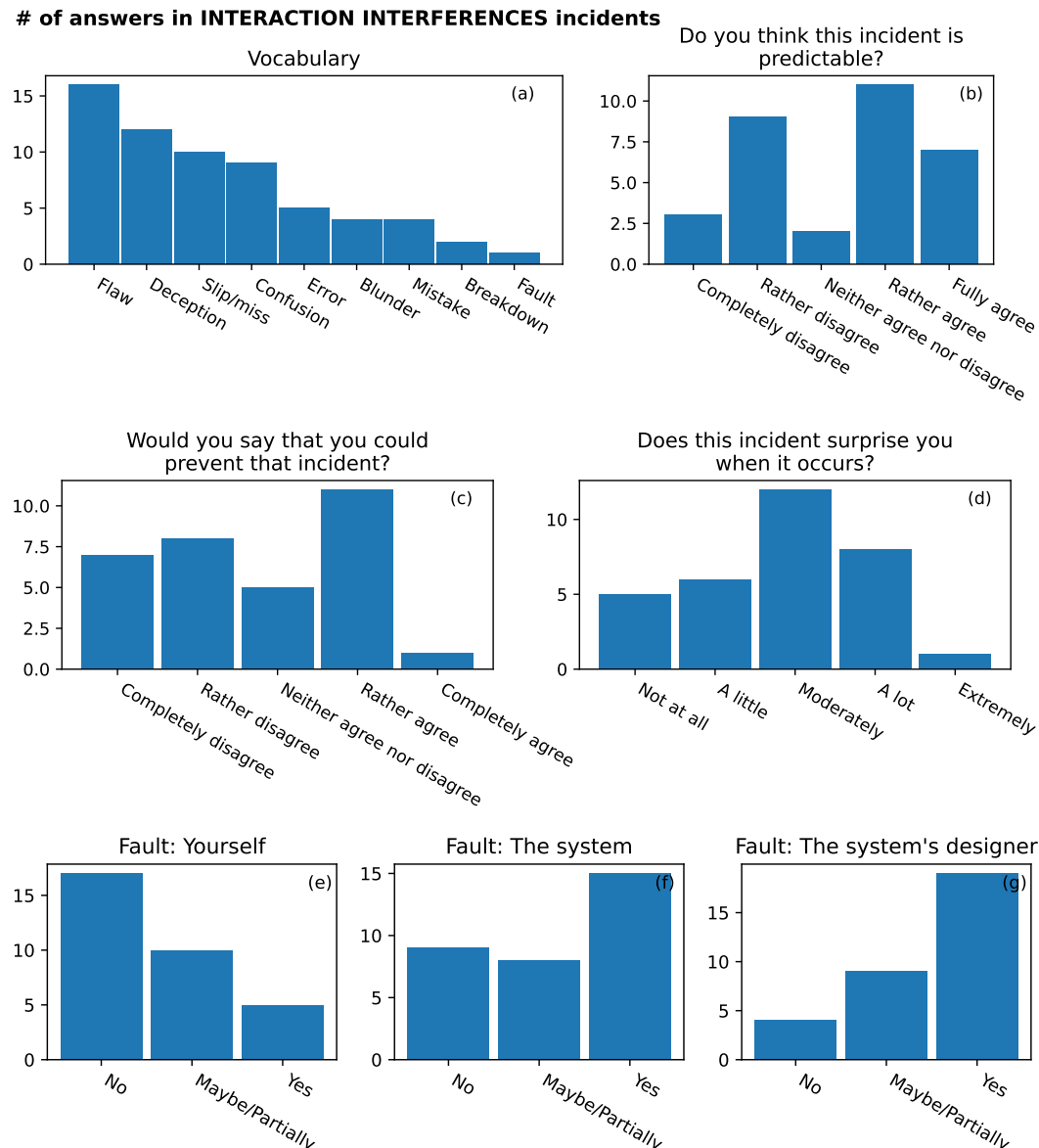


FIGURE 2.19: Selected answers for INTERACTION INTERFERENCE.

These phenomena happened at different frequencies depending on the participant, with a preference for “up to once a week” (24%) and “up to once a month” (27%). Participants reported that most occurrences happened on smartphones (67%) and/or laptops (61%), and at times on PCs (24%). Before the incident happened, 62% of people felt “absolutely” confident they could carry out their task. Most of those tasks were mundane (62%) and/or important (46%), and felt very easy (62%). People did not think the event was caused by bad luck (48% “completely disagree”), most even thought it to be rather predictable and preventable (see Figure 2.19 -b and -c).

People were somewhat surprised when the incident happened: 36% were “moderately” surprised (see Figure 2.19 -d). They did not feel guilty (64% “completely disagree”), but felt rather frustrated (42% “rather” and 39% “completely agree”).

INTERACTION INTERFERENCES mostly provoked a waste of time (70%) and/or a loss of confidence in the tool (33%). Those consequences were not at all serious 55% of the time, and 61% of participants felt entirely capable of fixing the incident. Those consequences were however frustrating: 42% and 39% of people respectively “rather” and “completely” agreed with that statement.

Participants did not blame an exterior element (70% “no”) nor an input device (73%), they blamed the system and its designer a bit more than they blamed themselves (see Figure 2.19 -e, -f and -g). We found a relatively strong effect size of confidence with one’s usual tools on self-blame ($p < 0.001, \epsilon^2 = 0.21$). Totally confident people faulted themselves less ($p < 0.1$) than highly confident people. 70% thought they did not deserve the incident to happen at all (“completely disagree”).

AIMED ACTION For this category, 55% of participants could remember a precise instance of an incident, and 45% general memory, which means all participants were able to relate to the presented *Sub-categories*_Q. This goes to show their commonality. Participants used the words “slip/miss”, “error” or “mistake” to describe it (see Figure 2.20 -a).

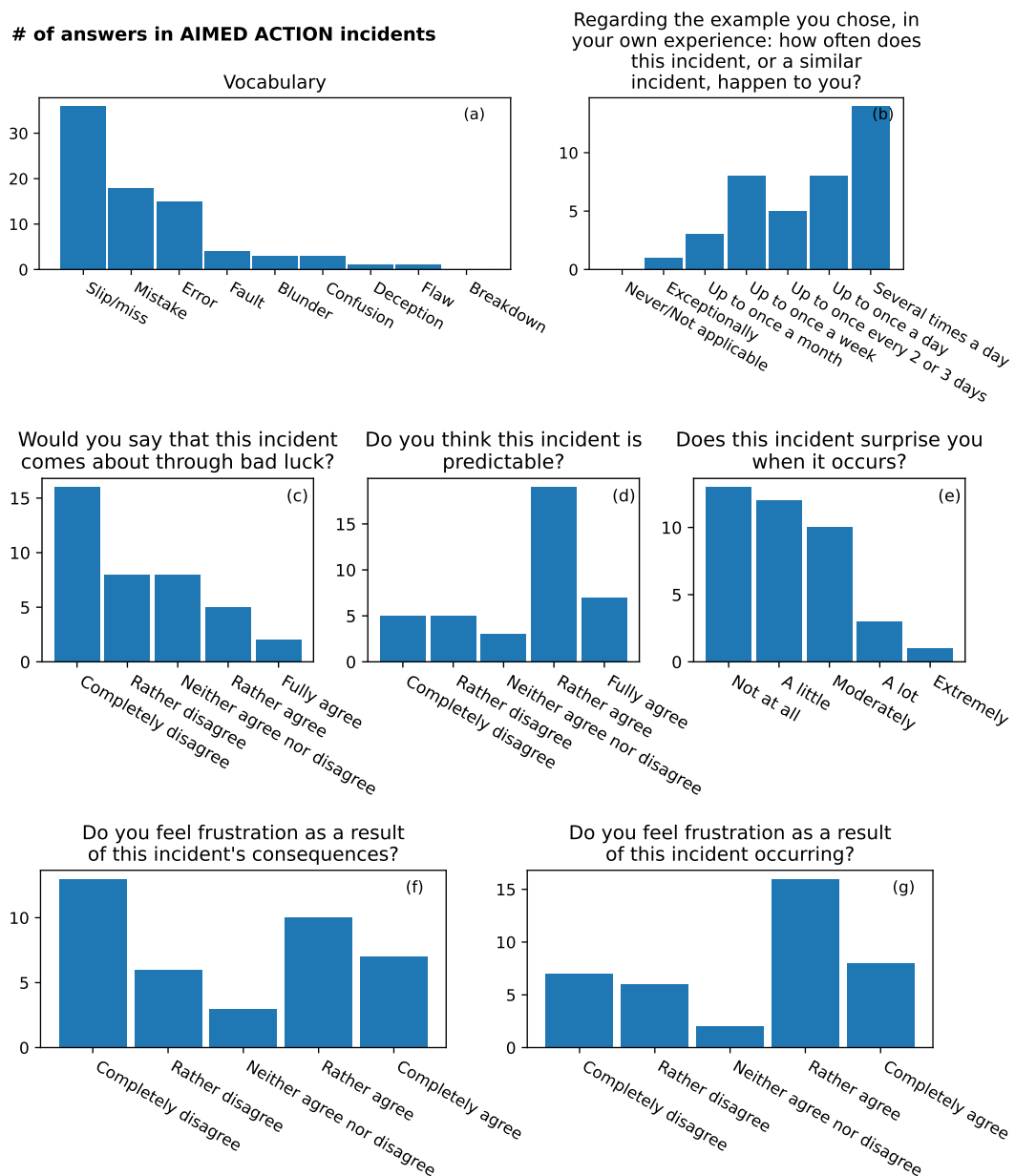


FIGURE 2.20: Selected answers for **AIMED ACTION**.

Instances were rather common (see Figure 2.20 -b), and rather recent (32% “today” and 36% “this week”). Before the incident occurred, 59% of participants felt absolutely capable of carrying out their task, and 41% highly confident. Those tasks were mostly mundane

(68%), but sometimes also important (23%) or interesting (18%). They were very easy (82%), extremely comfortable (45%) for the most part, and took place in extremely usual contexts (73%). Attribution to bad luck were rather low, and predictability rather high, as per Figure 2.20 -c and -d. Events were thought to be not intentional at all 80% of the time.

Participants did not feel guilty (63% “completely disagree”), felt surprise ranging from “not at all” to “moderately” and various levels of frustration (see Figure 2.20 -e and -g). We observed a strong effect size of gender on the report of surprise ($p < 0.05$, $\epsilon^2 = 0.41$). Men were more surprised than women ($p < 0.001$).

Consequences included a waste of time for the most part (60%), when there was specific consequences at all (43% of “no specific consequences”). The consequences were for 70% not at all serious and 88% of participants felt entirely capable of fixing the incident. A moderate effect size of gender appeared on the report of serious consequences ($p < 0.05$, $\epsilon^2 = 0.16$), with men reporting consequences to be more serious ($p < 0.05$) than women. The consequences prompted varying levels of frustration, as one can see on Figure 2.20 -f.

Participants blamed themselves in 75% of cases, and the system in 15% of cases. They did not blame an exterior element (78%). There is no strong tendency for blaming the system’s designer or an input device. They thought they did not deserve the incident to happen, with 58% of participant “completely disagreeing” with that statement.

SLIP 47% of people could remember a specific *Example_Q*, and 44% a general memory. To describe it, participants mostly selected the words “slip/miss”¹⁷ or “confusion”, as per Figure 2.21 -a.

This kind of event tended to happen exceptionally to quite regularly (see Figure 2.21 -c), on laptops (76% of participants), PCs (42%) and/or smartphones (33%). People remembered occurrences from the day before to more than one month ago, as one can see on Figure 2.21 -b). Before the event happened, people were quite confident they could manage their task: 47% felt “highly” confident and the same proportion “absolutely” confident. Reported incidents happened in the middle of any kind of tasks, but never the end of a long task or project. Those tasks were for 65% very easy. For participants, the event was mostly quite predictable and preventable, as per Figure 2.21 -d and -e.

People did not feel guilt (58% “completely disagree”), and felt various levels of surprise and frustration as a result of the incident occurring (see Figure 2.21 -f and -h).

Consequences mostly included a waste of time (55%), if no specific consequence at all (45%). Those consequences were not at all serious 67% of the time, and 61% of respondents felt entirely capable of fixing the incident. In face of the consequences, frustration was very variable (see Figure 2.21 -g).

People were likely to blame themselves (61%), about half (52%) did not blame the system, excluding “maybe” responses. 73% did not blame an input device and 82% did not blame an exterior element. They were more mixed on blaming the system’s designer (see Figure 2.21 -i). We found a relatively strong effect size of confidence with one’s usual tools on blaming the system designer ($p < 0.001$, $\epsilon^2 = 0.25$). People totally confident with their usual tools faulted the designer less ($p < 0.05$) than highly confident people. Participants thought they did not deserve the incident to happen (61% “completely disagree”).

FAIL-UNSAFE For this category, 51% of people remembered a specific example, and 27% a general memory. There was no apparent consensus on vocabulary, as Figure 2.22 -a shows.

Those events mostly happened exceptionally (62%), and occurred between the day before and more than one month before. **FAIL-UNSAFE** had in proportion the most older reported instances compared to other CATEGORIES: 42% happened “more than one month ago”. The task at hand was usually mundane (58%), but sometimes interesting (26%) and/or important (21%). For participants, the event was not caused by bad luck: 41% “rather disagree” and 45% “completely disagree”. It was also rather predictable (55% “rather agree” and 31% “fully agree”). 45% of participant thought that it was absolutely preventable. There is also a relatively strong effect of confidence with one’s usual devices on feeling the event was preventable ($p < 0.001$, $\epsilon^2 = 0.2$). People totally confident reported more preventability

¹⁷It is of note that we never provided the names of the CATEGORIES in the questionnaire.

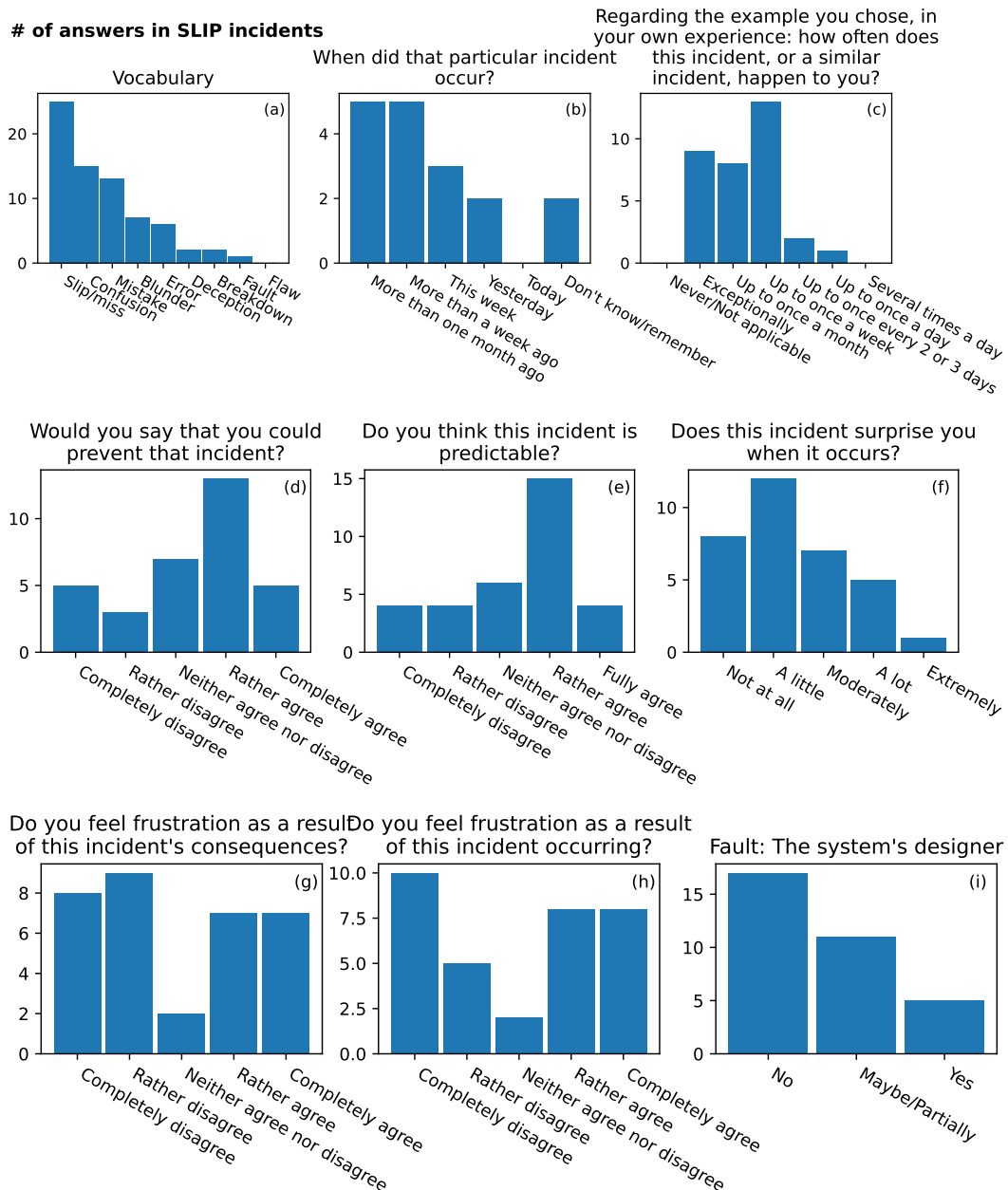


FIGURE 2.21: Selected answers for SLIP.

($p < 0.05$) than people with high confidence. 52% of participants thought the incident to be not intentional at all on the part of the system.

When the incident happened, people reported various level of surprise and guilt (see Figure 2.22 - and -c), and some frustration: 34% “rather agree” and 31% “completely agree”.

The consequences included a waste of time (52%), loss of credibility (17%) and loss of confidence in the tool (17%), if there are consequences at all (28% of “no specific consequences”). We found a relatively strong effects of confidence with one’s usual devices on reporting a loss of confidence in the tool ($p < 0.001, \epsilon^2 = 0.22$) or no specific consequences ($p < 0.001, \epsilon^2 = 0.36$). Loss of confidence with the tool was reported less ($p < 0.05$) by totally confident people, compared to highly confident people. People with total confidence also reported more ($p < 0.05$) no specific consequences, compared to people with high confidence. Surprisingly, there was not much loss of data (only 7%). 72% of participant felt entirely capable of fixing the incident. Frustration in face of those consequences was rather high, with 34% “rather agree” and 31% “completely agree”.

of answers in FAIL-UNSAFE incidents

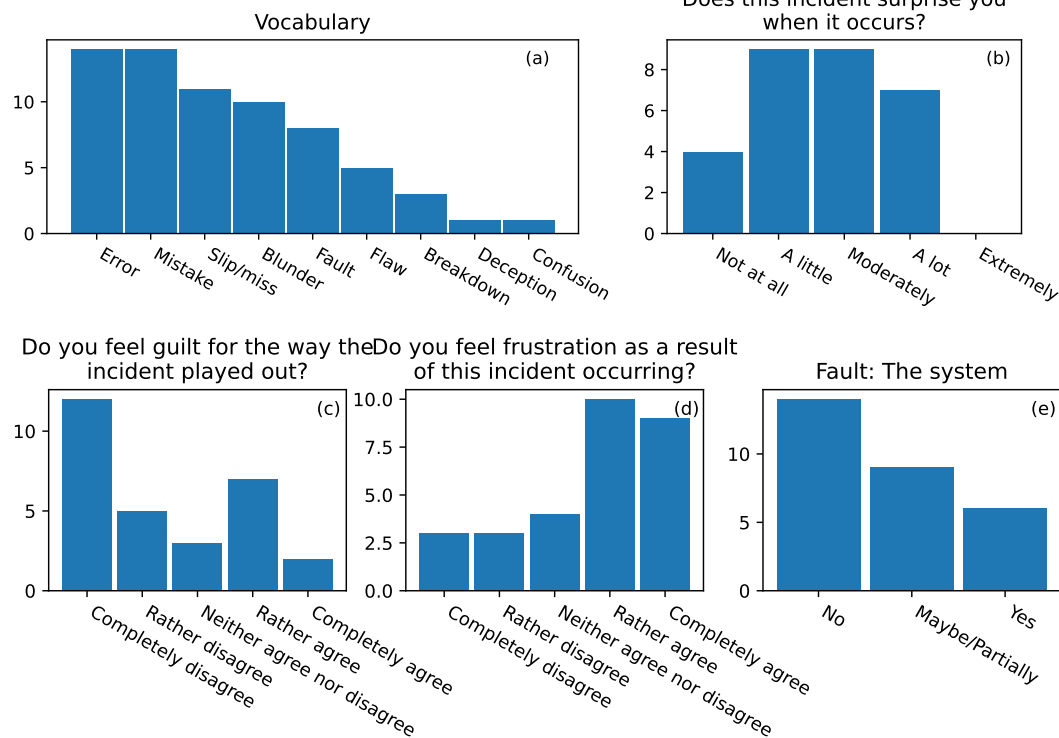


FIGURE 2.22: Selected answers for FAIL-UNSAFE.

Participants blamed themselves 76% of the time, and almost never blamed an input device (97% “no”) or an exterior element (66%). The blame on the system was more mixed (see Figure 2.22 -e). There was no strong tendency in feeling of deserving the event to happen.

OMISSION For the *Examples_Q* of this CATEGORY, 56% of participants could remember a specific occurrence, and 41% a general memory. To describe those incidents, they preferred the words “slip/miss”, “mistake”, “blunder” and “error”, as Figure 2.23 -a shows. Determining who or what was at fault was largely not important (51% “completely disagree”).

The reported incidents tended to happen exceptionally (63%). The last occurrence was more than one month ago 74% of the time. The events happened on laptops (63%), PCs (35%) and/or smartphones (33%). Before a specific incident happened, participants were highly (52%) or absolutely (39%) confident in completing their task. Those tasks were varied in type, but mostly mundane (57%). This context was very (48%) or extremely (30%) usual to participants, and felt very (43%) or extremely (43%) comfortable in great part. There was no clear consensus on the bad luck aspect, but most participants thought the event to be rather predictable (38%) and rather preventable (50%).

When the incident happened, people felt quite surprised (see Figure 2.23 -b), and frustrated: 45% “rather agree” and 30% “completely agree”.

The consequences of the event were mostly a waste of time, a loss of credibility and/or a loss of data (see Figure 2.23 -c). We found a relatively strong effect size of confidence with one’s usual devices on reporting a loss of confidence in the tool ($p < 0.001$, $\epsilon^2 = 0.19$). People with total confidence reported more “Loss of confidence in the tool” ($p < 0.05$) compared to people with high confidence. Their seriousness was variable, but overall low, as per Figure 2.23 -d. Those consequences were rather frustrating: 35% “rather agree” and 25% “completely agree”. 68% of respondents felt entirely capable of fixing the incident.

A vast majority of people blamed themselves (88%), 63% did not blame the system, 80% did not blame the system’s designer, none blamed an input device, 75% did not blame an exterior element. We found a moderate effect size of gender on faulting an external element

of answers in OMISSION incidents

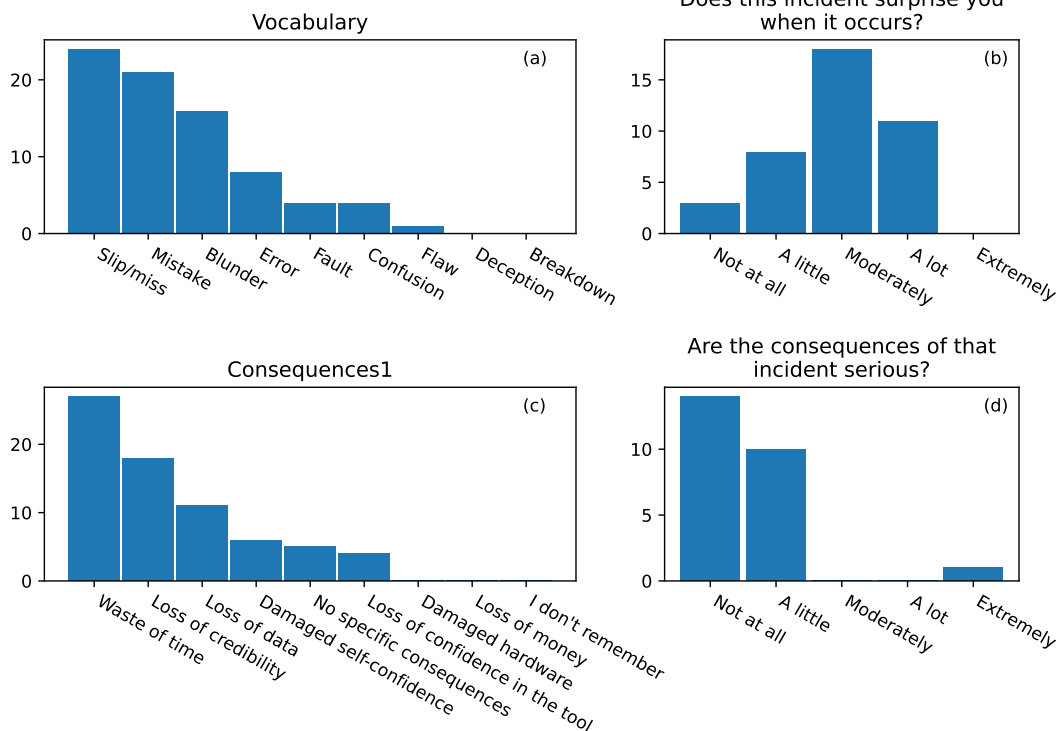


FIGURE 2.23: Selected answers for OMISSION.

($p < 0.01$, $\epsilon^2 = 0.11$). Men faulted an exterior element more for OMISSIONS ($p < 0.05$) than women.

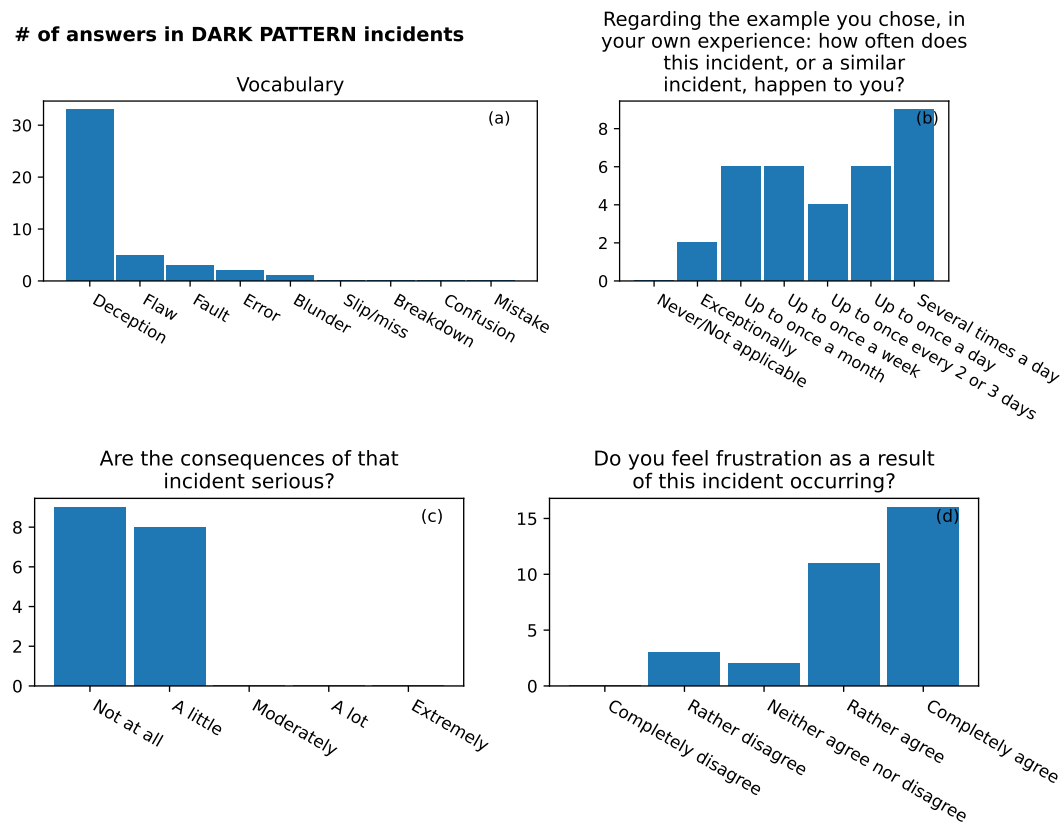
DARK PATTERN₊ 73% of people for this CATEGORY reported a specific incident, and no participant reported not having ever encountered such an incident before. The word strongly preferred to describe them was “deception” (see Figure 2.24 -a).

Incidents happened quite regularly, as Figure 2.24 -b shows. They mostly happened on smartphones (82%) and/or laptops (76%), but also on PCs (33%). Before a **DARK PATTERN₊** happened, 63% of participants were highly confident they could finish their task. 79% of those tasks were mundane, and/or sometimes interesting (42%). They were also easy in great part: for 42% rather and 50% very easy. 82% of participants did not attribute the event to bad luck at all. They perceived it as predictable (24% rather and 48% fully). In 79% of cases, they thought the incident was sure to be intentional on the part of the system or its designer.

Respondants were not surprised (27% a little and 45% not at all), felt no guilt (58% not at all), and quite frustrated (33% “rather” and 48% “completely agree”).

Consequences included mostly a waste of time (70%) and a loss of confidence in the tool (36%). We found a relatively strong effect of confidence with one’s usual devices on reporting no specific consequences ($p < 0.001$, $\epsilon^2 = 0.25$). People with total confidence never reported “No specific consequences” as a consequence to a **DARK PATTERN₊** ($p < 0.05$), as opposed to people with moderate confidence. Those consequences were rather not serious (see Figure 2.24 -c). People felt partially (45%) or entirely (45%) capable of fixing the incident. Those consequences also provoked frustration as one can see on Figure 2.24 -d.

Participants mostly did not blame themselves (67% “no”) or an exterior element (73% “no”); they blamed the system (48%) and most of all the system designer (91%). They almost never blamed an input device (6%). 91% completely disagreed with deserving the incident to happen.

FIGURE 2.24: Selected answers for **DARK PATTERN₊**.

ACCESSIBILITY₊ Only 4 respondents reported relating to this CATEGORY, which is too few to draw tendencies on their responses.

2.3.3 Discussing the survey

Global observations on errors

This questionnaire allowed us to uncover some insights about errors in general, regardless of the specific CATEGORY.

We observed that, while participants were encouraged to lean on either the most recent or representative occurrence of an event, most of the reported incidents happened during mundane tasks, in contexts both usual and comfortable to the user. Most of those tasks were deemed rather easy. This challenges the idea that errors might come up more in new, unfamiliar environments, while the user is experimenting or taking risks. Instead, the ones most salient to our respondents are part of the everyday, common experiences.

Those incidents were also rather not serious, which confirms that there are a lot of non-critical events that are nonetheless problematic to users.

In terms of consequences, the two most common reported outcomes were a waste of time and a loss of confidence in the tool. While the first is a physical consequence that could take priority in a classical, performance-based error evaluation, the second one is subjective and influences the deeper relationship the user has with the tool. More often than a loss of data, this outcome characterizes a negative event. Future studies on errors could benefit from taking it into account.

Scores for bad luck were rather low accross the board, showing that respondents did not attribute their experiences to accidents. They did not perceive errors as cases of misfortune. The existence of these incidents is perhaps seen as inherent to interaction, whatever or whoever might be perceived as causing them.

Some emotions were shown to characterize errors more than others.

Participants did not report feeling guilty as much as they reported frustration, both for incidents happening and for their consequences. While they blamed themselves at times, this judgement did not seem to translate into guilt. One possible explanation is that “guilt” might be considered a strong word for what people can feel while using a digital device. Frustration however proved once more to be a common experience in interaction (see section 1.4.1).

The question about feeling surprised did not score high overall either. There might have been an influence of its double meaning; “surprise” can both refer to an experience of suddenness, or a break of expectation. For instance, participants reported feeling particularly not surprised in front of **DARK PATTERNS₊** compared to other CATEGORIES. This difference makes more sense when considering surprise a break of expectation: a **DARK PATTERN₊** does not appear to be particularly less sudden than other types of errors, but is likely more “unsurprising” in that sense: people expect systems to act for the benefit of their designers.

Remarks on the CATEGORIES

The survey allowed to deepen the understanding of our CATEGORIES of error, and uncover main differences based on the participants’ responses.

First, we could observe that people related to the given *Examples_Q* overall, for every CATEGORY, which lends credit to their relevance regarding participants’ experience. Using the workshops to select the *Sub-categories_Q* and *Examples_Q* likely participated in this success.

In terms of vocabulary, the words used to describe events differed from one CATEGORY to the other. Offering a list of words to choose from to characterize an incident seems to be a good way to help its understanding, or at least its belonging to certain CATEGORIES, sometimes very clearly (as “breakdown” for **PROBLEM WITH THE EQUIPMENT** and “deception” for **DARK PATTERN₊**).

Despite near-ubiquity of both personal computers and smartphones users in our participants, they reported events to occur mostly on laptops. **WRONG SYSTEM INTERPRETATIONS**, **INTERACTION INTERFERENCES**, **FAIL-UNSAFE** acts and **DARK PATTERNS₊** appeared however to happen more on smartphones. The investigation of this type of problems can benefit from this insight: it may be more relevant to study those CATEGORIES specifically on their most common devices.

Blame attribution is also a discriminating factor accross CATEGORIES. According to our results, people tend to blame themselves for **ERRORS IN PLANNING**, **ACTIONS**, **SLIPS**, **FAIL-UNSAFE** acts and **OMISSIONS**, and blame the system for **PROBLEMS WITH THE EQUIPMENT**, **WRONG USER INTERPRETATIONS**, **WRONG SYSTEM INTERPRETATIONS**, **INTERACTION INTERFERENCES** and **DARK PATTERNS₊**. Participants seemed to blame one or the other; no CATEGORY had both high scores for blaming oneself and the system.

On top of those observations, respondents tended to blame the system designer for **WRONG USER INTERPRETATIONS**, **INTERACTION INTERFERENCES** and **DARK PATTERNS₊**. They put the responsibility on the system to have a clear, understandable interface, accounting for their physiological limitations and not to nudge them while interacting. Only **DARK PATTERNS₊** had high score for appearing intentional on the part of the system however. For the two other CATEGORIES, the designer is perceived as faulty, but not malicious.

Apart from **ACCESSIBILITY₊** which did not gather enough results, only the **INTERRUPTION** CATEGORY did not appear to present a strong blame tendency, either for the system or the user. In getting distracted or disturbed, users do not seem to strongly point fingers in either direction, when one could have expected them to blame the system. We can imagine

that **INTERRUPTIONS** are triggered by elements' behavior that users generally want on their devices; they do want to be prompted when a message or a notification comes in. In turn, they might "forgive" the system when those same elements occasionally cause problems.

ERROR IN PLANNING was the only **CATEGORY** for which the ongoing task was not evidently easy. Taken that the *Sub-category_Q* described a programming situation, this result makes sense. **ERROR IN PLANNING** appears indeed to be the only **CATEGORY** for which errors have to happen in a rather complex task, of the user setting an algorithm that will be executed by the device.

Some **CATEGORIES** also stood out in terms of frequency. For instance, **PROBLEMS WITH THE EQUIPMENT**, **FAIL-UNSAFE** acts and **OMISSIONS** were mostly reported to happen exceptionally, while **AIMED ACTIONS** happened several times a day. The perceived seriousness of events did not always match rarity: **PROBLEMS WITH THE EQUIPMENT** and **DARK PATTERNS₊** registered the highest scores on serious consequences, the latter not being particularly frequent or exceptional. This can challenge the idea that grave events are generally more rare; this trend does not appear in our **CATEGORIES** results.

Comparing workshops' and survey's results

While the questionnaire completed the workshops' investigation of errors by bringing quantitative insights, some results appear to differ and sometimes contradict each other between the two studies.

As mentioned before, workshops and survey presented differences for the frequency of wasting time as a consequence for errors, as well as for a blame of the system versus a blame of the user. Similarly, while **Depreciation** was rather present in the workshops, such sentiment did not emerge from the questionnaire results; scores for feeling guilty or deserving of an incident were low across all **CATEGORIES** and populations.

As an exercise, the workshops made possible for participants to influence each other. The resulting speech and expressions did not become false or less sincere because of it, but participants could feel validated in their experiences, in a way that they expressed different judgments about the system and themselves. We could recommend to keep this in mind while studying the perceived consequences of errors. Participants' feedback will be influenced by the situation in which it is given.

This difference in methodologies may have influenced as well the differences detected between populations, that were not mirrored in the survey's results. There was indeed few differences between people with varying expertises regarding interactive systems. There is also the possibility that the number of respondents did not allow such observations on a **CATEGORY** level.

The number of participants did prevent us from drawing conclusions about people with a high-school degree and about the **ACCESSIBILITY₊** **CATEGORY**. Those populations may benefit from specialized studies, or from an even larger-scale run of our survey.

The survey and the workshops provide together a first investigation of our errors categorization, deepening the understanding of users' perceptions and lived experiences. The collected feedback highlights the relevance of a broader definition, based on user expectations. Certain categories particularly reflect this broadening of the concept of error, such as **INTERACTION INTERFERENCES**. As the questionnaire results show, these phenomena are not perceived as "faults" per se, but are sources of frustration. Also, even if we were to look at the phenomenon from a classical error perspective, there is technically no one to blame. The system behaves exactly as the user has learned to expect, and the user cannot change their own physiological limitations. How, then, can an error in the sense given in this thesis be resolved? The following chapter examines possible solutions to pop-up type interaction interferences, where we address the "deviation from expectation" aspect by changing the expectation, *i.e.* the default behavior of interactive objects.

Chapter 3

Mitigating interaction interferences and their consequences: an applied study

As presented in subsection 1.4.2, an interaction interference happens when the interface changes during the motor process of an action by the user, who can no longer interrupt or modify their gesture. The phenomenon can cause a problem to the user, even if the system is reacting as per its specifications. While it does not happen very regularly, it can cause frustration [162]. For classical approaches of error, this subject is rather minor: it is not very frequent and, most of the time, does not provoke big consequences. According to the definition of error in this thesis however, interaction interferences are errors that get in the way of a fluid user experience, for which no targeted solution exist in typical Graphical User Interfaces (GUIs).

In this chapter, we investigate how GUI toolkits could be refined to help prevent or mitigate pointing-based interaction interferences by adapting the display and behavior of their widgets, depending on the extent of information and control that they can access: does the toolkit “know” the cursor location, or the location of upcoming changes in the interface? Can it alter the location, timing, or visual characteristics of that change? Can the next actions of the user be accurately predicted?

3.1 Background: Perception and impact of interaction interferences

3.1.1 Characterization of Interaction Interferences

In Schmid et al. [162]’s survey, the majority of reported examples of interferences (57 %) occurred when selecting a target, be it via indirect pointing or on tactile surfaces. For this reason we focused our later experimentation on selection-related interferences.

3.1.2 Causes of Interaction Interferences

On the “user side”, interaction interferences stem from neurological limitations of the visuo-motor system that generate narrow time windows within which a user’s planned motor action can no longer be inhibited, even when the user perceives a change in the system’s state within that window. Movement inhibition in general has been studied extensively in the fields of psychology and neuroscience (see e.g. [121] for a comprehensive state of the art), with the model of the “horse race” (Figure 3.1) currently prevalent in this research: “a ‘horse race’ between two sets of processes, one that generates a response for the primary task and one that responds to the stop signal: If the primary-task process finishes before the stop-signal process, the response is executed; if the stop-signal process finishes before the primary-task process, the response is inhibited” [109].

This phenomenon is studied using variations of the so-called countermanding protocol (see e.g. [180] for a detailed and substantiated description) in which participants are asked

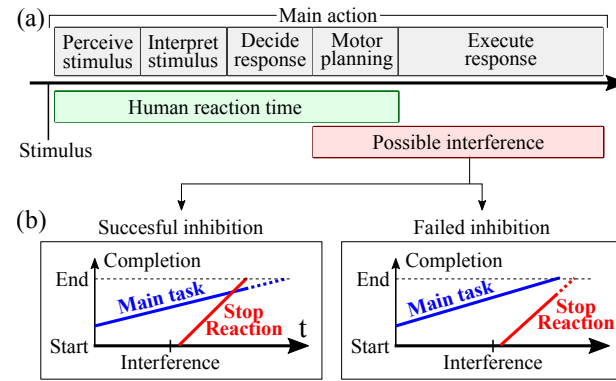


FIGURE 3.1: (a) Simplified diagram of human reaction to a stimulus. Interferences can start before movement onset. (b) The “horse race” model of movement inhibition. The user successfully suppresses his action if the stop-process ends before the main action process. Reproduced from [162] with the authors’ permission.

to perform a simple motor action as soon as a “go-signal” is emitted. In a small proportion of trials, a “stop-signal” is emitted shortly after the go-signal, in which case participants are instructed to refrain from starting the motor action.

Schmid et al. applied a modified version of this countermanding protocol to test in a reciprocal 2D pointing task the time below which a click cannot be inhibited. They found that participants reached a 50% click-inhibition success rate when the stop-signal occurred around 350 ms before the expected time of the click. Interestingly, they found that this “refrain-ability” was not affected by target distance, size, or pointing direction, indicating a process distinct from the reaching motion itself or its planning. Our experiment protocol builds on theirs, but with a simplified stop-signal timing mechanism since we are not attempting to precisely assess the duration between the stop-signal and the expected time of the click itself, or SSCD (Stop-Signal-to-Click Delay).

This can be somewhat reminiscent of the protocol used by Faure et al. [56] who, in order to study the effects of target-appearance delay on pointing performance, designed a task in which the participant always *expects* the target to appear, and knows *where* it will appear. Our protocol is substantially different: in our case it is very important that the user be surprised by the interfering window, so we took a number of steps to prevent that expectation (see Section 3.3 for details).

In the end, the human capacity to inhibit a movement is physiologically limited in time and that limitation cannot be significantly removed with training or experience. This limitation has been known since at least the early 80’s, yet GUIs are still designed with a systematic, simplistic handling of user inputs that ignores it. The vast majority of GUI widgets and window managers implicitly consider that as soon as a widget is displayed and active on screen, then the user events it receives are intentional and faithful indications of the user’s intention.

3.1.3 The Singularity of Interferences as Errors

In this section, we refer to an “error” in the general sense of a user action whose effect differs from the user’s expectations, which may or may not stem from a user mistake. An interaction interference is an error in this respect since an input (typically from the mouse or keyboard), carrying an intention, does not result in the effect intended at the time it was planned.

When discussing interferences with fellow researchers, I frequently see them likened to “slip” errors [164]. In HCI, a slip usually occurs in expert use of a tool and consists in a rushed, inattentive motor action caused by misapplied habit or practice. A slip does not only occur at the motor level, but also in the cognitive shortcut established by the repetition of a motor action. This type of error is linked to the cognitive process in action since it can be

avoided by procedural and sensory cues [11]. Interaction interferences, on the other hand, occur solely at the perceptuo-motor level.

As a last note, while most of our interference examples and the ones listed in [162] are of “honest mistakes” on the system side, *e.g.* displaying a timed notification or updating a list in real time, recurrent phenomena such as Internet ad pop-ups have primed us to associate pop-up windows with potentially malevolent intent. Pop-ups are often cited as a dark or deceptive UI pattern, *i.e.* “instances where design choices subvert, impair, or distort the ability of a user to make autonomous and informed choices in relation to digital systems regardless of the designer’s intent” [67]. Many examples of these techniques can be found on Brignull and colleagues’ website [25]. Interaction interferences are not in themselves a dark pattern, but one could imagine that some dark patterns could aim at triggering them, *e.g.* to click on an advertisement or to accept downloading a harmful file. However, similar to error taxonomies, we could not find interferences precisely described in the literature on dark patterns. “Interface interferences” in Gray et al. [68]’s classification refers UIs biased towards specific user options, such as pre-selection or false hierarchy. An interaction interference is not privileging an option for another, it forces the user to perform an action they did not intend.

The “Forced Action” category would fit better, but generally refers to trick wording or deceptive bundled consent. Pop-up windows themselves are often mentioned as dark patterns [48, 57, 113, 117] but for their very presence obstructing the interface, not in how they are triggered. The system could use the probability that a user performs a click at a certain time to change the interface immediately before and redirect the click. The user could then be tricked into clicking on a button, for example opening an add or completing a purchase. We leave to future work the demonstration of existing intentional interaction interferences. In this work we did not consider interaction interferences for the deceptive pattern they might be, but for the accidental phenomenon they are. When interaction interferences are caused on purpose, the problem lies in the designer’s intention. The designer will not seek to mitigate the consequences of the phenomenon, since their system is designed precisely to provoke them. Meanwhile accidental interaction interferences are not intended by either the user or the system designer. It therefore makes sense to implement technical solutions. We can of course imagine that these solutions and their systematic implementation could mitigate this deceptive pattern in the future.

3.1.4 Preventing and Correcting Interferences

The only previous work directly related to interaction interferences [162] focuses on the *problem*: it describes the overall phenomenon in detail, assesses its understandability and some of its qualitative aspects with a survey, and quantifies its timing for point-and-click tasks. The present study is a first exploration of the space of *solutions* that can be used to reduce their occurrences or consequences.

Interaction interferences are virtually impossible to avoid by the user alone once the motor action is planned and the inhibition threshold crossed. One could formulate naive strategies of being extra careful right before clicking a target or pressing any key, however (1) that would require constant additional attention, for most non-user-initiated interface changes cannot be anticipated; and (2) in a study wherein 15% of trials could feature an interference, *i.e.* much higher than in real life and therefore less likely to be a surprise, participants still failed to inhibit their movement 90% of the time when the change occurred less than 100 ms before the click [162]; Schmid et al. also report that their participants’ improvised “tricks” to increase their inhibition success rate, including both slowing down and accelerating before the click, failed to produce any observable result on their inhibition performance.

On a system level, we found no existing tool or technique to prevent, avoid or correct interaction interferences specifically, however that is not surprising considering that their first characterization is recent. Existing interaction mechanisms can be put to use, such as confirmation steps (*e.g.* “Are you sure you want to perform that action?”) for consequential actions, or Undo-ing editing [124] or direct manipulation [7, 8] actions. They do not however provide a universal solution to the problem, by a large margin: confirmation steps need to be explicitly implemented for any given action, and the confirmation itself is an action that

can be subject to interferences; regarding Undo, command histories do not record every user action, including some among the most consequential such as closing an app or sending an email.

3.2 Solutions to Interferences

Interaction interferences occur at temporal scales of up to a few hundred of milliseconds [162], too short to be avoided by users reacting to the change.

Solving or mitigating them therefore requires adjustments on the GUI side, by taking human capabilities and reaction time into account in the design of interactive systems.

In what follows we introduce a design space of GUI *behaviors* aimed at minimizing the chances or the consequences of interaction interferences, and primarily targeted at interaction designers and software engineers. This work is a first attempt at solving interferences and, as the reader will observe, that often implies interceding or even disrupting the usual flow of input and output events. Some of these adjustments are not trivial to implement in regular GUI frameworks, which were not designed for it, and thus we focus our design space on technical feasibility: which sensing, interpreting, and read/write access is required, at which layer of the architecture. More precisely, our design space includes dimensions on the *interpretative* and *technical* requirements necessary to enable these behaviors. From this space we implemented a set of behaviors specifically designed to prevent or mitigate *pointing-based* interaction interferences, that we evaluated in a controlled experiment.

3.2.1 A Design Space of Solutions

We define the design space of behaviors along three dimensions:

1. what it does, *i.e.* the way the behavior would solve or mitigate interferences, which we split into overall **categories** and their respective **types**
2. its **interpretative requirements**, *i.e.* how “smart” it needs to be in assessing a situation for interferences (behind or ahead of time)
3. its **technical requirements**, *i.e.* the minimal features that a GUI toolkit needs to expose to allow this behavior

Types of behavior

We distinguish categories of interference-solving behaviors based on when these behaviors occur relative to two main events: the interface change, and the moment the system responds to the user event (see Figure 3.2). This results in three categories: *Prevent* before the interface change, *Avoid* between the interface change and the sensed user event, and *Correct* after the system response.

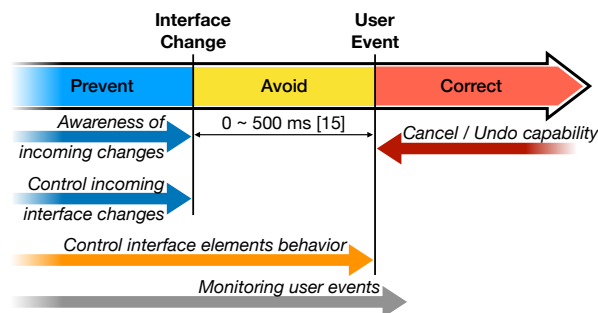


FIGURE 3.2: Time windows for the three main categories of behavior (*Prevent*, *Avoid*, *Correct*) and their corresponding technical requirements (in italics).

Ideally, systems *Prevent* interferences entirely without delaying or withholding information from the user, by making the change occur in a way that cannot interfere with their incoming actions and with little to no additional latency or visual disturbance. In particular, such behaviors can be implemented by delaying the change until the user's ongoing or incoming action is over (a type of behavior we henceforth dub "POSTPONE"), or, when the interference is location-dependent, by displaying the change away from the user action ("DISPLACE").

When the problematic GUI change cannot be prevented, it should be possible to *Avoid* the resulting interference or its consequences using GUI behaviors that disregard the change, or that do not automatically handle the subsequent events as if they were intentional. Such behaviors can be implemented by BLOCKING (*i.e.* ignoring) consequential events until enough time has passed, or by MAPPING these events to the GUI state prior to the change. For instance, when an application starts automatically it generally grabs the keyboard focus, even when the user was typing in a text field in another app; a MAPPING-type solution would be to still send the keyboard events to the text field for a short period of time. In cases where the interference is caused by an element appearing on top of another, this MAPPING could be materialized as letting events go THROUGH the foreground elements.

Note that, since interference-*Avoiding* behaviors occur after the interface change, these strategies can remain problematic to users who perceived the change and might briefly worry about the consequences of their action, before they can observe that these consequences were avoided. *Prevent*-type behaviors do not have this issue, since they affect the interface change itself.

When nothing can be done to prevent the interfering GUI change or to avoid its wrongful interpretation, post hoc *Correction* should be made available, *e.g.* , by systematically enabling ongoing actions to be cancelled and finished actions to be undone, and by making non-undoable actions to require confirmation before they start. Such corrections could be either AUTOMATIC when the system can confidently estimate that a recent action caused an interference, or PROMPTED by directly asking the user if they wish to correct its consequences, with the respective risks of false positives or annoying the user.

Finally, *a minima*, the system should be able to INFORM the user that an interference is suspected to have happened. Note that even this last-resort solution requires dependable means to detect interferences automatically.

The categories and types of behaviors are summarized in Table 3.1.

Interpretative requirements

Different types of behaviors also require different degrees of situational interpretation from the system, typically to assess the incoming "threat" of interference, or to confirm that one has just occurred. We distinguish in particular the capacity to *Predict* or *Detect* interaction interferences on the one hand, and *Naïve* behaviors that are always active on the other hand.

A peculiarity of interface interferences is that we can only be sure that a GUI change is interfering *after* a user action was interfered with. Altering a specific GUI change in order to prevent an interference therefore implies the ability to *Predict* that it will interfere with incoming events. This is conceivable in situations for which accurate user models exist, *e.g.* slowing down at the end of a unidirectional cursor movement before a click or training a gesture recognizer to detect the *likely* start of a command. In other contexts, such as keyboard shortcuts or touchscreen taps, there might be too little information to go around without additional sensors.

Similarly, *Detecting* that an interference has just occurred, *i.e.* that a couple {GUI-change ; user-event} constitutes an interference, can be nontrivial without explicit user feedback. Typical examples involve finger, hand, or device gestures for which it can be difficult to pinpoint an exact starting timestamp. For pointing tasks, there exists a grey area after an interface change wherein click or tap events might or might not correspond to an interference [162]. The behavior types in the *Correct* category typically all require *Detect* capability, since otherwise the user would be prompted for confirmation or for Undo *after every single action* they take.

Behavior Category	Behavior Type	Predict	Naive	Detect
Prevent : the change occurs in a way that cannot cause an interference.	DISPLACE*: occurs away from incoming user events.		X	
	POSTPONE: occurs after at-risk user events.		X	
Avoid : the change occurs normally but the GUI behavior averts the wrongful interpretation of the action.	BLOCK: ignores user events for a time.		X	
	THROUGH*: events skip the foreground element for a time.		X	
	MAPPING: handles events as if they preceded the change.		↑	
Correct : the system offers means to cancel or rectify the consequences of a (likely) interference.	AUTOMATIC: applies corrections automatically.			
	PROMPT: asks the user whether to apply the corrections.			
	INFORM the user of suspected interferences.			

TABLE 3.1: Possible combinations (in blue) of behavior types (rows) ordered by categories (1st column), and interpretative requirements (last three columns) in our design space. Asterisks (*) indicate behavior types that are location-dependent. Cells marked with ‘X’ indicate behaviors evaluated in the following controlled study.

In our study of pop-up interferences below, the condition implementing THROUGH (*Naive*) will also serve as a demonstrator for MAPPING (*Naive*) (↑).

In contrast, GUI elements can be assigned “*Naive*,” always-on behaviors that may by themselves be enough to prevent some interferences. All three approaches pose risks of false positives and visual disturbance, to a degree that remains to be explored.

In the remainder of this work we focus on solutions that do not require conjecture (as opposed to *Predict* or *Detect*), first because designing and assessing prediction/detection methods is a distinct endeavor that is beyond the scope of this study, and second because we postulate that many contexts and toolkits today expose too little information to confidently detect interaction interferences, let alone predict them. We therefore focus this study on “*Naive*” solutions that can be always active, as a first exploration of the design space of interference-solving behaviors.

Technical requirements

The behaviors outlined above can easily be ranked in terms of desirability (for what is better than simply preventing any problematic situation?) but there exists a trade-off with the amount of information and control they each require:

1. *Awareness of incoming interface changes*, i.e. the ability to be notified when an interface change is about to occur, is required for any behavior of the **Prevent** category. For instance: a pre-planned calendar reminder is scheduled to appear in a second; a new word suggestion was just computed and is about to be displayed above a gesture keyboard; new emails have been received and will be added to a visual list. Unless there exists native callbacks or hooks for these events, which is at best very rare, this implies access to application- or webpage-specific information that is often only available to its developers.
2. *Control over incoming interface changes*, i.e. where (DISPLACE) or when (POSTPONE) a scheduled interface change should occur first. This is distinct from e.g. translating or hiding an *existing* GUI element, which is usually feasible at the OS or toolkit level. This type of capability is rarely controllable from outside the source code of a program and generally requires access to application- or webpage-specific logic and control.
3. *Monitoring of user events*, typically recent cursor displacements, the location and timing of mouse clicks, or key presses, is required for any behavior that attempts to *Predict* or

Detect an interference. Whether this monitoring is system-wide or application-specific (e.g. JavaScript mousemove events fired from within the web browser window) defines the scope at which the corresponding interference-solving behavior can be used. For instance, the mousemove events of a cursor moving towards a web browser window will not be received by the corresponding webpage's scripts until it reaches that window.

4. *History of recent interface changes* and states, e.g. the ability to know when an interface actually changed, what the previous word suggestion was, or the state of the email list prior to its recent update, is necessary to *Detect* recent interferences and to temporally remap input events to prior states of the interface. In most use-cases this history does not need to be complete or long-lasting, since interferences only involve recent interface changes; nevertheless, this is perhaps the least likely capability to have available today since command histories generally do not include interface updates.
5. *Control over interface element behaviors*, i.e. the ability to modify their aspect, their opacity, or directly the handles associated to events occurring upon them or their sub-components, is the minimal requirement for all *Naïve* behaviors.
6. Finally, overall *Control over the resulting commands* that were triggered as a consequence of user events, typically the ability to pause, cancel, or undo a recent command, is a requirement for all behaviors of the *Correct* category.

Table 3.2 indicates the (minimum) technical requirements to implement the eight behavior types defined earlier in this section.

							Behavior type	

and assessing prediction or detection methods is a distinct endeavor that is beyond the exploratory goals of this study, and we expect that *Naive* behaviors can already offer significant improvement over many default GUI behaviors.

A majority of the participant-submitted interaction interference examples gathered by Schmid et al. involved pointing target acquisition as the triggering user event, the rest consisting of typing-related events and device gestures [162]. Even within pointing-based interferences, Schmid et al. identified three categories: *Updates* (an existing GUI element changes right before the click or tap), *Focus switch* (the user events are suddenly sent to the wrong GUI element), and *Pop-ups* (a new GUI element appears above the targeted one). We aim, ultimately, to design and evaluate behaviors for all possible types of interferences, and this study presents our first step in this experimental endeavour by focusing on the *Pop-up*¹⁸ category as a representative subset of pointing-based interferences. Based on our design space, we designed four of such naive behaviors to address pop-up-style interferences, that implement the four types of behaviors marked with a “X” in Table 3.1.

AWAY_{DISPLACE} (Fig. 3.3) is an instance of **DISPLACE (Naive)** that prevents *Pop-up*-style interferences by systematically displaying the appearing GUI element away from the path of the cursor, yet close enough that it can be accessed with minimal traveling.

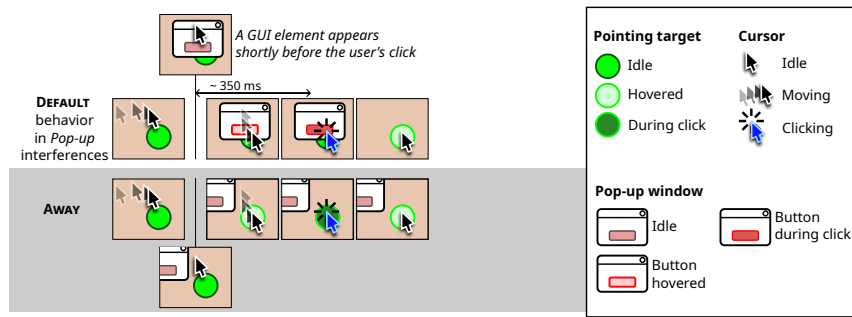


FIGURE 3.3: The **AWAY_{DISPLACE}** BEHAVIOR makes the window appear at the same time as initially planned, but some distance away from the cursor's path. Clicking as planned selects the target.

To do so, we constantly retain the recent 100 ms of cursor coordinates (in our case the last 10 cursor events at a frame rate of 1000 Hz, assumed constant for simplicity) and we display the new GUI element so that (1) its center is on the line perpendicular to the latest cursor displacement and (2) the distance between its closest edge and the last cursor location is the same as between the last and 10th last cursor location (see Fig. 3.4). As a result, the faster the cursor, the further away the new GUI element from its path to reduce the risk of accidentally reaching it. The appearing GUI element's position was corrected when necessary to remain within the screen. This systematic approach is designed to be simple with minimal assumptions about the user's behavior or the positions of targets of interest, and to solve the common case of interferences occurring while or right after reaching a target. It is currently target-agnostic for generalizability, but the display coordinates could easily be adapted to avoid known target areas altogether.

To simplify, we assume a scenario wherein the system does not know in advance where and when the new GUI element was initially scheduled to appear (*i.e.* no *Awareness of incoming changes*). One notable downside of **AWAY_{DISPLACE}** is therefore that it overrides that planned position regardless of the cursor's location. Assessing whether that brings performance advantages (the new GUI element always appears near the cursor, yet out of its path) or usability issues (the new GUI element appears in unrelated locations) is left for future work.

¹⁸We wish to clarify that the scope of the term *Pop-up* used by Schmid et al. is not limited to “traditional” pop-up windows *per se*. It designates any interaction interference caused by a GUI element appearing above the one currently targeted by the user, *e.g.* when a notification appears in the screen's corner or when a starting app appears in the foreground.

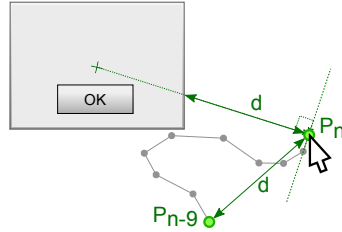


FIGURE 3.4: **AWAY** displays the new GUI element so that its center is on the line perpendicular to the latest cursor displacement, and the distance between its closest edge and the last cursor location P_n is the same as between the last and 10th last cursor location P_{n-9} .

DELAY_{POSTPONE} (Fig. 3.5) is an instance of **POSTPONE (Naive)** that avoids pointing-based interferences by systematically waiting for a pause in the stream of cursor movements. In our implementation, the behavior waits for a pause of at least 200 ms in mousemove events to display the new GUI element. This delay was calibrated in pilot studies and is on the safe side of the reaction times measured in [162]. Note that this mechanism does not guarantee systematic interference avoidance: a pause in cursor movement could still be followed by a late click, which could itself trigger an interference.

Another possible risk with this behavior is that the appearance of a new GUI element could be delayed for an extended period of time during intense cursor activity. To circumvent this possibility, we implemented the additional rule that if a click occurs while a pending GUI element is awaiting a pause in cursor events, then that element is immediately displayed, based on the hypothesis that an interference is less likely to occur immediately after a click¹⁹.

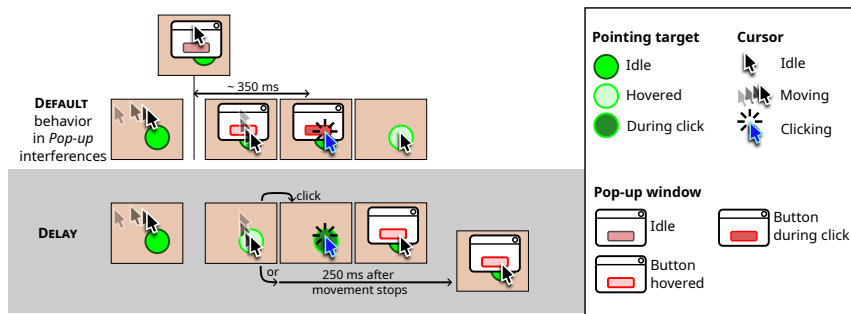


FIGURE 3.5: The **DELAY BEHAVIOR** postpones the appearance of the window until 250 ms have passed, the cursor stops moving, or until a click. Clicking as planned selects the target and immediately triggers the appearance of the window. Moving the cursor around postpones that appearance even further.

HOLEOVER_{THROUGH} (Fig. 3.6) is an instance of **THROUGH (Naive)** that lets click events pass through new GUI elements for a fixed duration of one second after they appear, calibrated in pilot testing. As feed-forward, a ‘hole’ in the new GUI element follows the cursor for the duration of the behavior. Arguably a feed-forward cannot help in situations of true interaction interference in which, by definition, the user cannot take visual cues into account before acting: we added it for situations in which the user knows in advance that a GUI element will appear (e.g. out of habit) and needs to know when the **HOLEOVER_{THROUGH}** behavior times out.

¹⁹This creates the risk of missing a double-click, but double-clicks were not part of the input vocabulary of our experiment. This rule could be lifted in double-click intensive interfaces, or a delay could be added to the rule corresponding to the system’s double-click threshold.

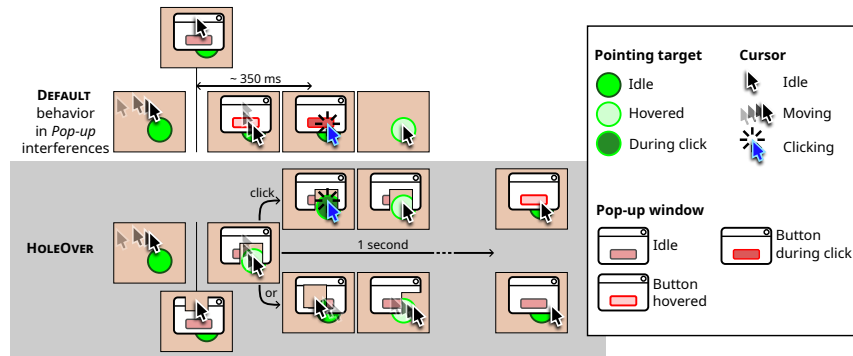


FIGURE 3.6: With the **HOLEOVER** BEHAVIOR the window appears at the planned time and place, but a hole in the pop-up follows the cursor and sends input events through for 1 second. Within that second, clicking as planned selects the target. Clicking after more than 1 second affects the window normally.

FREEZE_{BLOCK} (Fig. 3.7) is an instance of **BLOCK (Naive)** that renders new GUI elements unresponsive for a fixed duration after they appear, ensuring that a pre-planned click will not trigger their contents. However, contrary to **HOLEOVER_{THROUGH}**, the initial target does not receive the event. This arguably “safe” approach trades a potentially consequential interference (e.g. closing a notification without having read it, or unintentionally accepting an operation) for an inconsequential one (the user’s click does nothing). Through pilot experiments we chose a freeze duration of one second to prevent *non-interfering* GUI elements to remain unresponsive for too long after they appear, which could impede the user’s workflow. Visually, we gave the frozen GUI element the common visual characteristics of an inactive window, i.e. grayed-out background, text and buttons. Finally, to avoid frozen GUI elements becoming a frustrating delay *after* they have prevented an interference, an element becomes responsive again after the first click event it receives regardless of the one-second threshold. This relies on the hypothesis that an interference is less likely to occur immediately after a click, already used in the design of **DELAY_{POSTPONE}**.

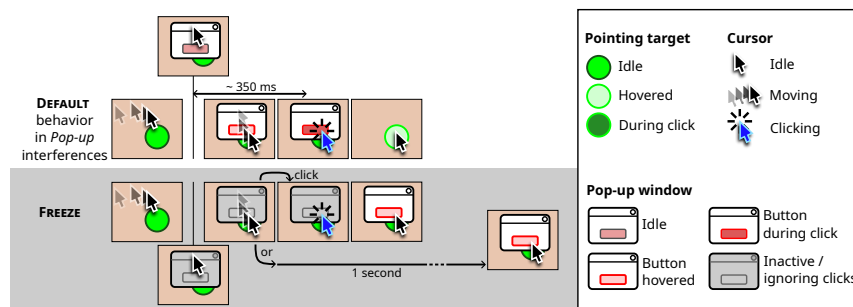


FIGURE 3.7: The **FREEZE** BEHAVIOR makes the appearing window unresponsive for 1 second or until a click. Clicking as planned deactivates the “freeze”, after which any action on the window is treated normally.

A number of these behaviors could be applied as-is to other types of interaction interferences, pointing-based or not. We discuss this further at the end of the chapter.

DEFAULT	Popups will now appear without any special behaviour. We will call this behaviour "Default".
AWAY	Popups will now appear in a different manner called "Away". They will spawn away from the current cursor position. You can try moving your cursor while waiting for the popup to spawn.
DELAY	Popups will now appear in a different manner called "Delay". They will spawn either when the cursor is not moving or just after a click. You can try moving your cursor and immobilize when you want the popup to appear. You can also move your cursor around and click.
FREEZE	Popups will now appear in a different manner called "Freeze". For a little time after it has spawned, the popup freezes and blocks the clicks. If it is clicked, the freeze effect ends. You can try clicking anywhere on the popup when it has just appeared or wait for the freezing effect to go away.
HOLEOVER	Popups will now appear in a different manner called "Hole". For a little time after it has spawned, the popup lets the clicks pass through a hole around the cursor. You will see the hole as you move your cursor over the popup. You can try clicking through the popup when it has just appeared to get the target. You can also wait for this effect to end by hovering over the popup.

TABLE 3.3: Interference-avoiding behaviors as they are implemented and explained to the participants in the experiment. As opposed to the rest of this study, the behavior names presented to the participants did not include the category they belong to, *e.g.* **AWAY** instead of **AWAY_{DISPLACE}**.

3.3 Controlled Study

We conducted a controlled experiment to evaluate the GUI BEHAVIORS defined above in terms of interference avoidance, user feedback, and overall user experience. Our main objectives were to assess whether, and how, the different *Naive* behaviors defined above can successfully prevent *Pop-up*-style interaction interferences or mitigate their effects without deteriorating user experience, and to better understand the trade-offs at hand when designing systems to address them. We later discuss how these results could generalize to other forms of interaction interferences, and what remaining issues could be solved with non-*Naive* GUI behaviors.

3.3.1 Participants

20 participants took part in the study (5 F, 15 M, ages 21 to 48, mean 31.5). All participants had normal or corrected-to-normal vision. None of the participants reported current dyslexia or dyscalculia, as the task involved character recognition (see below). Participants were of multiple nationalities and replied either in English or in French. In the latter case, quotes of their feedback are translated in what follows and marked with an asterisk.

3.3.2 Apparatus

The experiment was conducted on a 2020 13-inch Apple MacBook Pro running under macOS Monterey version 12.2.1. Cursor control was performed via a Dell M-BAC-DEL5 MY897 832205-0000 USB Wired 5 Button Optical Scroll Wheel mouse controller plugged via usb, set to 1000 dpi and with the system tracking speed setting set to 6 out of 10 levels. The experiment was displayed on the native built-in retina display 13 inches monitor with a resolution of 2560×1600 pixels (227 ppi). The experiment software was implemented as a web application using JavaScript.

3.3.3 Tasks

Each trial in the experiment featured a primary task of target acquisition and manipulation. In 40% of trials a pop-up window appeared during the primary task, in which case a secondary task consisted in accurately reporting that window's contents.

Primary task

The primary task consists in bringing numbered visual targets onto a central drop-zone via *pick-and-drop* [155]: clicking a target *picks* it and makes it follow the cursor's positions; clicking again *drops* the target at its current position. The drop-zone is a black disc (3.9 cm diameter) in the background, always visible at the center of the screen. The visual targets are colored discs (1.3 cm diameter) that appear at semi-random locations (see details below) around the drop-zone, and numbered incrementally to indicate their ordering (see Figure 3.9). Targets appear progressively as the participant completes the trials, always one target ahead of the current task. The current target to pick is displayed in yellow, the next one in cyan. If the current (yellow) target is not entirely contained by the drop-zone after it is dropped, the participant has to pick it and drop it again until it is so. The dropped target (numbered n) then becomes green, the new current target ($n + 1$) switches color from cyan to yellow, and a new target (in cyan and numbered $n + 2$) appears at a semi-random location—unless $n + 1$ is the last target in this block. Figure 3.8 shows the experiment's appearance.

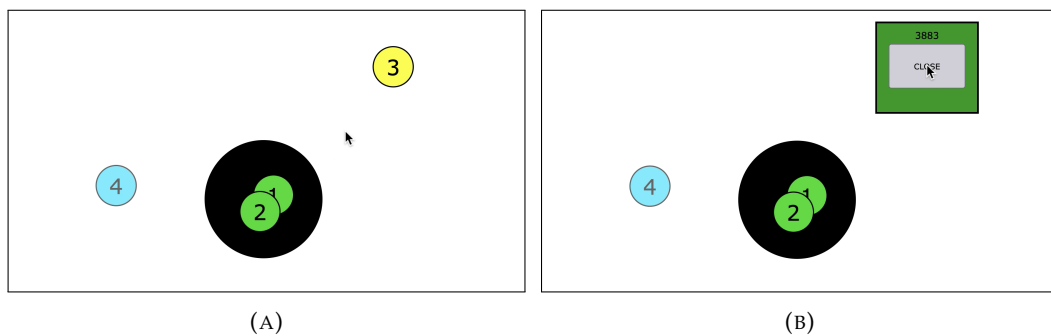


FIGURE 3.8: Experiment task (a) before and (b) after a popup is displayed (to scale, cropped)

Target locations were pre-generated to ensure that every block for every participant had a uniform distribution of distances and orientations from the drop-zone. Participants were instructed to complete the primary task quickly and accurately in each trial.

Secondary task

At the beginning of a trial, before the first click on a target (*pick*), a small window could appear, which the participants could not move around. It contained a 4-character *label* and a large [CLOSE] button which closes the window when pressed (see Figure 3.9). The timing, location, and BEHAVIOR of these windows' appearance depend on the independent variables of the experiment, detailed below. By default, windows are scheduled to appear with the [CLOSE] button on top of the ongoing target (see Figure 3.8-a). Trials with and without pop-ups are henceforth referred to as *Pop-up* and *Normal* trials, respectively. In *Pop-up* trials, after the participant successfully completes the pick-and-drop task, a multiple-choice question appears above the drop-zone displaying the pop-up window's *label* and three similar-looking strings, in random order, as well as an "I don't know" option and a [Submit] button (Figure 3.9). Participants are instructed to select the option that was displayed on the window and press 'Submit'. If they select any of the four-letter options, a last question inquires whether they are sure of their answer (yes/no). In our results a participant's answer is categorized as *Correct* if they selected the string corresponding to the window's label, *Incorrect* if they selected one of the three other strings, and *Unknown* if they selected the "I don't know" option. After that, the new trial begins. The three alternative answers were designed to look

similar to the *Pop-up* window's (actual) label, so the participant could not guess the correct answer after only a glimpse at the window before it was closed. These alternative answers are composed of the same characters as the actual label, themselves chosen to look similar to each other. Labels and wrong answers could be composed of letters (e.g. "NMMN", "MNNM", "MNMN", and "NMNM"), digits (e.g. "3883", "8338", "8383", and "3838"), or both (e.g. "6GG6", "6G6G", "G6G6", and "G66G"). For this reason, we specified during our participant recruitment process that we were looking for people who do not suffer from dyslexia or dyscalculia.

Unbeknownst to the participant, there were two types of *Pop-up* trials: *Interfering* pop-ups (20% of all trials) are scheduled to appear exactly 100 ms after the cursor enters the ongoing target, to let the cursor move inside the target while maximizing the chances of triggering an interference²⁰; *Non-Interfering* pop-ups (20% of all trials) are scheduled to appear early enough that the participant can refrain from clicking them, using a delay defined in an initial, per-participant calibration procedure (see 3.3.4 below) and previous findings [162]. *Non-Interfering* pop-ups were introduced to simulate "regular" pop-ups that do not necessarily conflict with user interaction, and to steer participants' attention away from interferences, our main focus. In doing so, it also allows us to (1) assess whether the differences between *Normal* and *Pop-up* trials are due to interferences or to the mere presence of pop-ups, (2) assess the possible negative consequences of our behaviors on normally-appearing GUI elements, and (3) obtain a baseline for some of our comparison metrics.

We acknowledge that appearing GUI elements cause interaction interferences less than half the time in real life. These 20-20-60 proportions were chosen through pilot tests to satisfy several constraints at once: *Pop-up* trials had to be sufficiently numerous to collect enough data for analysis (especially since *planned Interfering* trials do not always end up causing interferences), but also rare enough to keep the participants' focus on the main task, all while keeping manageable study duration. In particular, participants should not wait for interferences at every turn but instead maintain a pointing behavior as normal as possible in every trial.

The question about the window's label in *Pop-up* trials (Fig. 3.9-6a) appears as soon as the target is correctly dropped. With *Non-Interfering* trials, and with some of the tested BEHAVIORS, this means that the pop-up window can still be visible when the question is displayed. We did not give specific instructions about whether to close pop-up windows systematically before answering the questions. This, in turn, also means that the time between the closing of the popup and the appearance/answering of the question is variable. A trial's pop-up window closes automatically once its question(s) are answered.

Occurrences of *Pop-up* trials are semi-random: their order was pre-calculated for each participant in order to maintain the proportions of 60% - 20% - 20% for respectively *Normal*, *Non-Interfering*, and *Interfering* trials within each BEHAVIOR condition; like in [162], we also ensured that there could be no consecutive *Pop-up* trials. In the event that a participant is able to click the target before the scheduled appearance of a pop-up window (*Interfering* or not), the trial is automatically reclassified as *Normal* in our logs. If it was an *Interfering* trial, a new *Interfering* trial with the same BEHAVIOR and with the same target position is inserted at the end of the block. If the last trial of that block was already scheduled to contain a pop-up, another *Normal* trial is inserted in between, making it two additional trials.

While both Schmid et al.'s [162] and our protocol are designed to evaluate phenomena related to *pointing-based* interferences, our task differs because we are interested in different information and phenomena. First, Schmid et al.'s conditions are either classic target acquisition ("No-stop" trials in [162]) or countermanding ("Stop" trials); it lacks the intentional *Non-Interfering* trials that we use to compare our BEHAVIORS' performance to *Interfering* trials, and our BEHAVIORS' side-effects to *Normal* trials. Second, Schmid et al. focused on precisely measuring the inhibition capability of their participants, making precise variation of SSCD a significant aspect of their protocol; this is unnecessary in our case, and would have made our study longer. Third, we implemented a pick-and-drop task instead of a simple target acquisition because we wanted participants to have a goal past their first click, even

²⁰Note that this 100 ms threshold does not correspond to a 90 % chance of interference, as [162] Fig. 7a could be read; 100 ms after entering the target does not equate to an SSCD of 100 ms.

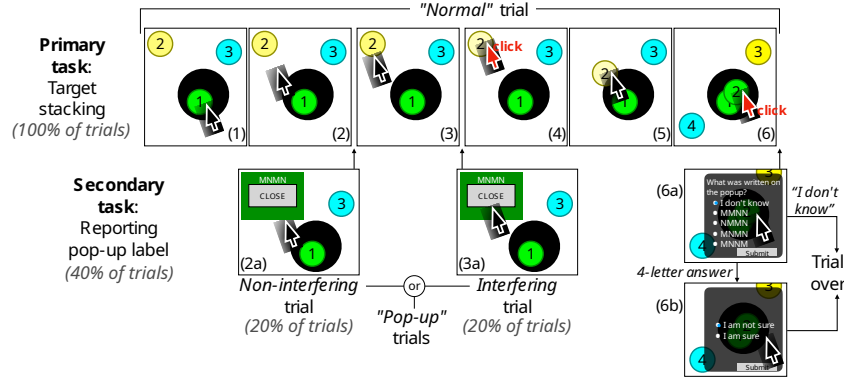


FIGURE 3.9: A trial in our study. (1-2) The participant moves their cursor towards the current target in yellow. (2a) A pop-up window may appear if the trial is *Non-Interfering*, or (3-3a) 100 ms after the user enters the target in *Interfering* trials. (4-6) In any case the participant completes the primary task. The dropped target turns green, the new current target turns yellow, and a new target appears in cyan. (6a-6b) In *Pop-up* trials, the participant is asked what is/was written on the pop-up window. Sizes and distances were adjusted for figure readability; the cursor did not actually turn red on clicks.

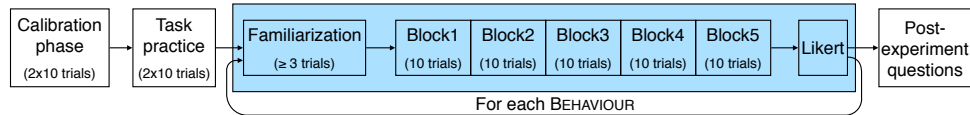


FIGURE 3.10: Main phases and blocks of our experiment protocol.

in *Normal* trials. Sequences of target acquisitions are common in real use of a computing device, and it allowed us to further divert the participants' attention away from the prospect of appearing windows. More pragmatically, it also allowed us to control the distance parameter of our trials by bringing the participant's cursor near the center of the screen at the end of every trial. Schmid et al.'s task always started from the location of the previous target, but doing so in a 2D task poses the risk of non-uniform distributions of distances or orientations.

3.3.4 Procedure

Participants were first invited to sit at a desk in front of the experiment computer and to sign a consent form. They were then explained that the goal of the study was to "evaluate different appearing behaviors for pop-up windows". They were also encouraged to take breaks between blocks or conditions, as often as they needed. The phases of the experiment are illustrated in Figure 3.10.

Calibration phase

In order to estimate a safe apparition time for the pop-up windows in *Non-Interfering* trials, and to detect whether participants started slowing down as a response to the interfering pop-ups overall [162], we started the experiment with an initial performance calibration phase that also served as training for the primary task.

Participants were instructed to perform 2 blocks of 10 *Normal* trials, i.e. with only the primary pick-and-drop task—and without knowing about the secondary task yet. For each participant P we collected the movement times between the start of the trial and the first click on the current target²¹, and calculated the regression parameters (slope and intercept)

²¹Pop-up interferences could only happen before the first click of the trial.

of the “Shannon formulation” [115] of Fitts’ Law [58]. These parameters were used in two ways.

First, previous work showed that participants of countermanding-like studies tend to slow their response down after a while [180, 162]. We therefore defined a threshold function that calculates, for any Fitts’ Index of Difficulty (ID) within the tested range, the maximum duration above which we consider that participant P is moving significantly slower than during the calibration phase. This was calculated using the following formula, appraised in pilot tests:

$$T_{\max,p} = i_p + ID \times (s_p + 1.96 * SD(MT_p)) \quad (3.1)$$

with i_p and s_p respectively the intercept and slope of the Fitts’ Law regression from participant P ’s calibration, and $SD(MT_p)$ the standard deviation of their measured Movement Times during that calibration.

When the participant took longer than $T_{\max}$ to click the target, a full-screen message informed them that they had slowed down too much and prevented them to complete the trial for 4 seconds as deterrent. The goal was to ensure that participants could not avoid interferences by altering their speed—consciously or not. We used a conservative threshold (orange line in Figure 3.11) to avoid interrupting the tasks too frequently while still penalizing obvious slowdowns.

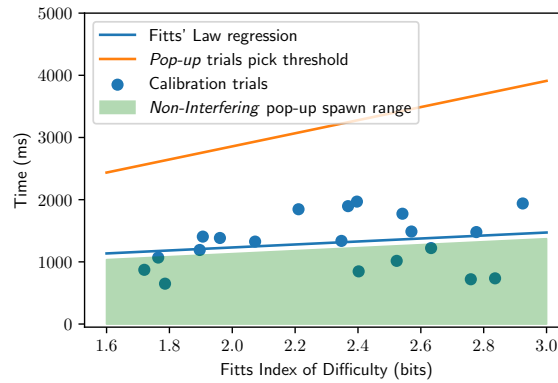


FIGURE 3.11: Fitts Law and time thresholds for Participant 17: the Fitts law regression (blue line) is calculated from the durations of the calibration trials (blue dots). The time range within which the *Non-Interfering* windows spawn (green area) spans from the beginning of the trial (0 ms) to 100 ms before the Fitts regression, depending on the Fitts Index of Difficulty of the target. The threshold to trigger the 4-second warning message also depends on the Fitts law regression and on the Index of Difficulty.

Second, we used the Fitts’ Law regression parameters to calculate the display time of *Non-Interfering* pop-up windows. For each *Non-Interfering* trial, the pop-up window is scheduled to appear after a duration chosen at random (uniform) between the beginning of the trial and 100 ms before the *expected* click on the target (green area in Figure 3.11). If the model predicts a picking action shorter than 100 ms, the pop-up window is set to appear as soon as the trial begins. This 100-ms offset was selected during pilot tests, both to avoid too many out-of-script interferences and to allow some variation in the window-spawning times, in order to maintain the illusion that our scripted *Interfering* pop-ups were simply “bad luck”.

The calibration phase occurred only once for each participant at the beginning of the experiment (see Figure 3.10).

Task practice phase

Participants were then informed that from that time on, pop-up windows could appear at random during a trial and that they were to report their contents after successfully dropping

the target. No further information regarding pop-ups was provided. Before introducing the new BEHAVIORS—the main independent variable of this study—we wanted participants to become familiar with the secondary task. We therefore introduced a task practice phase in which they would encounter *Pop-up* trials, both *Interfering* and *Not*, but without BEHAVIORS other than **DEFAULT**. This phase consisted in 2 block of 10 trials with the same ratios of *Interfering* and *Non-Interfering* trials as the main part of the study. After this began the main data collection.

BEHAVIORS phase

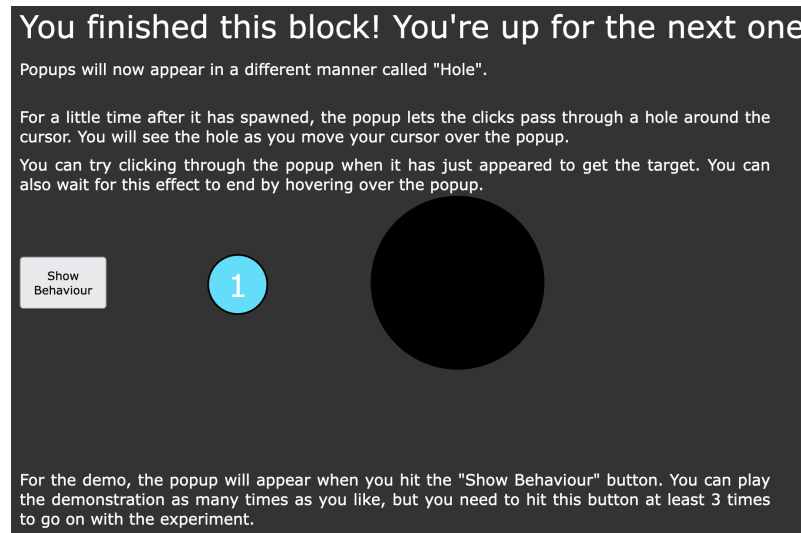


FIGURE 3.12: Screenshot of our experimental interface during the familiarization phase.

The rest of the study is split in five parts, one per tested BEHAVIOR (see Table 3.3 for a reminder), plus one “**DEFAULT**” behavior corresponding to standard appearing windows identical to the *Task practice phase* above. Each BEHAVIOR starts with a *familiarization phase* (Figure 3.12) wherein participants are shown the drop-zone, a single target, a concise explanation of the current BEHAVIOR, and a [Show Behaviour] button. Clicking the button reveals a pop-up window implementing the current BEHAVIOR, complete with a label and a [CLOSE] button. Pick-and-drop is active, because some aspects of our BEHAVIORS depend on user actions such as moving or clicking. With **DELAY_{POSTPONE}** and **HOLEOVER_{THROUGH}**, for instance, participants might observe by themselves what happens when the cursor moves, but they need a reason to click (here, picking the target) to observe the behavior-cancelling effect of clicking. Dropping the target in or out of the drop-zone does not end a trial or begin a new one. If the target is dropped in the drop-zone, it is simply relocated at a random location within the screen. The goal of this *familiarization phase* is to try out the BEHAVIOR’s response to timing, user input, or other. Participants can ‘spawn’ as many new pop-up windows as they want until they feel familiar with the BEHAVIOR. After three clicks on the [Show Behaviour] button, a new button labeled [Next] appears at the bottom-right of the screen; clicking it allows the participant to start the first block of this BEHAVIOR.

The next 5 consecutive blocks of 10 trials follow the **Tasks** descriptions in subsection 3.3.3. In particular, each block has a uniform distribution of target distance from the center of the screen (between $d_{\min} = \text{radius}_{\text{dropzone}} + \text{radius}_{\text{target}}$ and $d_{\max} = 0.9 \times \frac{\text{display_diagonal}}{2}$) and a mostly-uniform²² distribution of orientation from the drop-zone, and a 60%-20%-20% odds of respectively *Normal*, *Non-Interfering*, and *Interfering* trials.

²²Accounting for the screen’s aspect-ratio: not all angles are usable at distances larger than half the display’s smallest dimension.

Questions about the BEHAVIORS

I think I would like my popup windows to behave like this in general
 I found the popup behaviour unnecessarily complex
 I thought the popup windows were easy to interact with
 I think most people would understand this behaviour very quickly
 I understood this behaviour very quickly
 I found the popup behaviour cumbersome
 I felt very confident that I would not close a popup by accident
 I felt very confident closing the popups on purpose

Questions the pick-and-drop (primary) task

NA option

I felt in control of the situation in the task	
I felt I had time to change my course of action before clicking on a popup that just appeared	X
I felt surprised by popups	
I felt disturbed in my task by popups	
I was annoyed by popups	
If I could not read the popup content before closing it:	
- I was calm	X
- I was frustrated	X
- I felt responsible	X
- I blamed the system, the popup or my mouse	X
- I felt angry at myself	X
- I felt angry at the system, the popup or my mouse	X

TABLE 3.4: Questions asked after each BEHAVIOR. NA stands for “Non-Applicable”.

After the fifth block of each condition, participants were asked to answer several questions on 7-point Likert scales regarding their impression with this BEHAVIOR (listed Table 3.4). The questions included assessments of frustration, responsibility and blame assignment. These criteria showed up consistently during the analysis of the workshops and differ between different categories of errors in the questionnaire. We then moved on to the *familiarization* phase of the next BEHAVIOR.

At the end of the experiment, after the last BEHAVIOR, participants were asked to rank the different BEHAVIORS by order of preference. Finally, the experimenter invited the participants to comment on their ranking and on the experiment in general.

3.3.5 Design

Our study used a within-subject experiment design with pop-up window BEHAVIOR (**DEFAULT**, **FREEZE**_{BLOCK}, **HOLEOVER**_{THROUGH}, **AWAY**_{DISPLACE}, **DELAY**_{POSTPONE}) and TRIALTYPE (*Normal*, *Interfering*, or *Non-Interfering*) as independent variables. Order of presentation of BEHAVIOR was counterbalanced across participants using a balanced Latin square design. TRIALTYPE (*Non- /Interfering*) was randomized within each block following the proportions and rules described earlier. On-screen target location was pseudo-randomized in order to maintain a uniform distribution of directions and distance to the drop-zone, and to guarantee that all participants had the same target layout for the same BEHAVIOR and blocks.

In total, we collected $5 \text{ (BEHAVIORS)} \times 5 \text{ (blocks)} \times 10 \text{ (trials)} = 250 \text{ trials}$ at least²³ per participant. Calibration and task practice trials were not considered in the study analysis. The experiment lasted 57 minutes per participant on average.

²³Extra trials could be added at the end of blocks when a click occurred faster than a planned pop-up, see subsection 3.3.3.

3.4 Results

We collected 5043 trials in total. The initial study plan contained 5000 trials, 2000 of which were planned to feature pop-ups: 1000 aimed to create an interference, 1000 did not. In 192 (9.6%) of these 2000 planned pop-ups trials, the participant clicked the target fast enough to prevent the pop-up's appearance. Only 32 of these trials were aimed to create an interference, thus 32 new *Interfering* trials were added at the end of the same block, as described in subsection 3.3.3, in order to obtain the 1000 interfering pop-up trials that we initially aimed for. In situations where the block already ended with a *Pop-up* trial (11 out of these 32), and since we decided not to have consecutive pop-up trials, we also inserted *Normal* trials between the planned last trial of the block and the added pop-up trial. This adds up to $5000 + 32 + 11 = 5043$ trials.

The number of "missed" pop-up trials is strongly unbalanced between planned *Interfering* (32) and *Non-Interfering* (160) trials. We posit that this is because the window of opportunity to click the target before an interfering pop-up is much shorter (100 ms), making it less likely to occur.

3.4.1 Data Pre-processing and Analysis Approach

In what follows and unless specified otherwise, time measurements are aggregated as geometric means per participant and condition for statistical analysis or plotting purposes, as recommended for pointing studies and skewed datasets in general [161]. Other types of numerical data, such as numbers or proportions of correct/incorrect/unknown answers in the secondary tasks, are aggregated as arithmetic means per participant and condition.

Reported significant effects are obtained using Repeated-Measure Analysis Of Variance (RM ANOVA) performed on block aggregates. Sphericity is controlled using Mauchly's test, and degrees of freedom are corrected using the Greenhouse-Geisser correction when necessary. Post-hoc tests are pairwise t-test comparisons with Bonferroni p-value adjustment method and Tukey's Honestly Significant Difference [111] (HSD) tests when the independent variable had more than two levels. Error bars in bar charts and all Confidence Interval (CI)s represent the 95% CIs from bootstrap with 999 resamples over participant aggregates, using the adjusted Bias-Corrected and accelerated [183] (BCa) method of the `boot.ci`²⁴ statistical package.

Likert-scale answers and BEHAVIOR rankings are analyzed using Friedman tests with BEHAVIORS as groups and participants as blocks. We ran post-hoc pairwise comparisons using Wilcoxon rank sum test with continuity correction, using Bonferroni p-value adjustment.

Finally, qualitative feedback was coded by one of the authors in an iterative process of deductive thematic analysis [24], focusing on themes related to (1) the BEHAVIORS themselves, (2) emotional responses to the interferences and to the pop-up BEHAVIORS, and (3) recurrent themes that participants talked about such as notifications or the influence of the task on the appreciation of a pop-up BEHAVIOR. This feedback is reported with examples and descriptive statistics.

In what follows we explore our results by theme rather than by data type (e.g. time then errors then quantitative-subjective then...), applying the principle of triangulation [114] to strengthen our findings and conclusions.

3.4.2 Triggering Interferences

Our experimental protocol displayed *Interfering* pop-ups 100 ms after the mouse pointer entered the target, which differs from Schmid et al.'s procedure [162] that relied on a threshold value continuously updated from each participant's behavior throughout the experiment. Therefore, before concluding anything about the effectiveness of the tested BEHAVIORS, we need to ensure that our conditions reasonably triggered the desired phenomenon. In particular, we need to ensure that (1) trials scheduled to be *Interfering* (often) contained interferences, (2) trials scheduled to be *Non-Interfering* did not trigger (too many) interferences,

²⁴<https://www.rdocumentation.org/packages/boot/versions/1.3-28/topics/boot.ci>

and (3) participant behavior during the initial reaching phase of the primary task did not significantly change as a result of either experiment progression or pop-up BEHAVIOR.

From user interactions

Note that, as opposed to the countermanding protocols used in [162, 180] there is no definite way for us to label some trials as “successfully interfered with” when the click occurs, *e.g.* X ms and more after the window popped up. Indeed, some participants can be more alert, or faster readers than others. Our main guideline comes from Schmid et al.’s finding of 350 ms as the 50% mark for successful click inhibition in a standard 1-D pointing task [162].

Figure 3.13 shows the distribution of delays between pop-up appearance and the first mousedown event of the trial for each TRIALTYPE and for all BEHAVIORS (negative times thus correspond to a first click happening before the pop-up spawned). We observe that more than 60% of all intended *Interfering* trials had their first click *earlier* than 350 ms after the pop-up appearance, and more than 67% of all *Non-Interfering* trials had their first click *later* than 350 ms after the pop-up appearance. If we remove **DELAY_{POSTPONE}** from the set, since it alters the *timing* of the window appearance²⁵, we obtain an “interfered-with” ratio of 53 %. While a 53-60 % ratio of successfully triggering interferences may seem quite low, we stress that this phenomenon is probabilistic in nature. Schmid et al.’s results correlate the Stop-Signal-to-Click Delay (SSCD) to the chances of successfully inhibiting a planned click, so there is no “sure-fire” way to trigger it. Plus, precisely implementing an SSCD implies to have extremely accurate estimates of the user’s pointing performance and behavior, in all situations rather than on average, which is easier to achieve in a 1D, single-task protocol. Instead, we chose to use target entry as a reference frame because it is a mandatory step in selecting the target, and under the hypothesis that selection occurs shortly after it. This is of course an imperfect approximation: in case of *e.g.* fast-approaching ballistic movements that result in target overshoot, the window might pop up before the final phase of the pointing action, *i.e.* before the participant “sent the (motor) order” for the click. For reference, a 55-60 % success rate corresponds to a Stop-Signal-to-Click Delay of approximately 300-350 ms (Fig. 7a and 9 in [162]).

We also highlight that our 4-string labels were designed to be difficult to parse, being composed of similar-looking characters in an order that matters for the post-drop question. The average reading speed is estimated around 330 ms per word in English [27], but with known words. The combined durations of noticing the window, inhibiting one’s planned click, reading the precise sequence of characters, and committing it to memory (including distinctions such as “G6G6” vs. “6GG6”) likely took longer. We could therefore use a higher threshold—assuming that participants were dedicated to completing the tasks as well as possible—but we decided against it because instrumenting participants to measure precise reading phases would have added more time and discomfort to the study. We thus maintain our 350 ms as a conservative threshold, and its corresponding 53-60 % probability, while keeping open the possibility that a number of interferences might have occurred after longer delays.

From secondary task answers

The secondary task consisted in accurately reporting the label of the pop-up window, which is (by design) harder to do if that window was closed prematurely. Figure 3.14 shows the percentages of *Correct*, *Incorrect*, and *Unknown* answers gathered in the **DEFAULT** condition—the other BEHAVIORS are designed to facilitate the secondary task—and separated by TRIALTYPE. We observe a noticeable drop in correct answers in the *Interfering* trials, that matches an increase of *Unknown* answers. A RM ANOVA found a significant effect of TRIALTYPE on the percentage of correct answers in **DEFAULT** trials ($F_{1,19} = 15.985, p < .001$), with *Non-Interfering* trials resulting in significantly more correct answers (82.04%, $CI = [67.64, 89.16]$) than *Interfering* ones (60.5%, $CI = [48.39, 69.5]$). Conversely, a significant effect of trial type

²⁵All values below 0 in Fig. 3.13 are **DELAY_{POSTPONE}** since it is the only BEHAVIOR in which a first click occurring after the *scripted* pop-up time can still occur before the window appears.

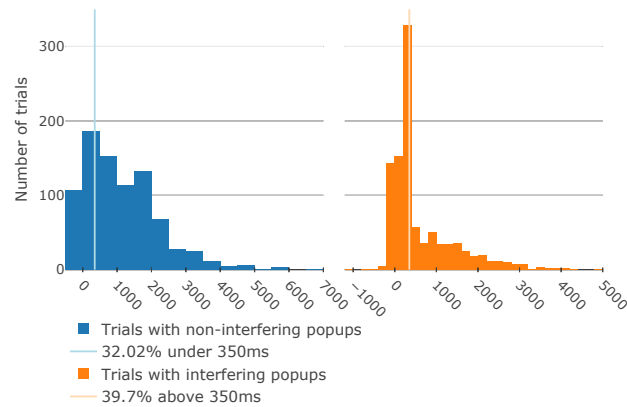


FIGURE 3.13: Distribution of the time between the appearance of a window and the first click of the trial, for all BEHAVIORS. Vertical lines indicate the 350 ms mark.

was also found on the percentage of *Unknown* answers ($F_{1,19} = 14.905, p < .01$) with *Non-Interfering* trials resulting in fewer “I don’t know” answers (13.68%, $CI = [7.44, 35.38]$) than *Interfering* ones (34%, $CI = [24.5, 46.95]$). The difference between the percentages of wrong answers with *Non-Interfering* (4.28%, $CI = [1.09, 9.75]$) and *Interfering* (5.5%, $CI = [2.5, 9]$) was not found significant. The amount of *Unknown* answers is a good indication that participants “played along” and did not try to guess at least some of the labels that they did not see. In the worst case, assuming a participant who purposefully selects a label without any idea of what was displayed on the pop-up window, we can expect a 75 % probability of selecting the wrong answer; considering that there are only about 4.9 % of *Incorrect* answers in our results, we can deduct that up to roughly 1.6 % of *Correct* answers could be caused by luck—if all participants intently ignored our instructions all the time. Participants in our baseline condition, **DEFAULT**, provided significantly fewer correct answers and more “I don’t know” answers in *Interfering* trials, which corroborates with our goal of triggering interaction interferences. These values will serve as comparison baselines later in this section.

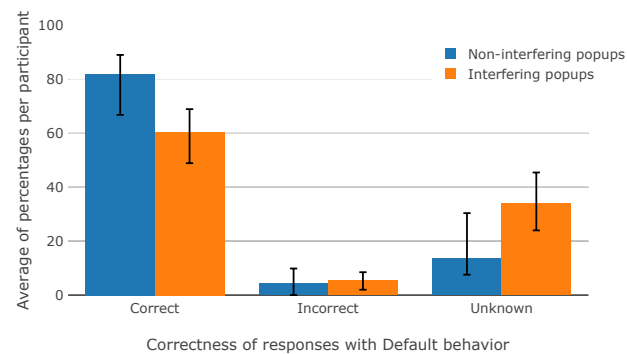


FIGURE 3.14: Proportions of *Correct*, *Incorrect*, and *Unknown* answers to the secondary task with the **DEFAULT** behavior, as a function of TRIALTYPE.

From subjective feedback

Participant feedback about the task in general, and about the **DEFAULT** condition in particular, expressed issues that we confidently assign to interaction interferences. 9 participants out of 20 reported frustrations referring to interferences, for instance complaining of being “*annoyed by [pop-ups] appearing*” [P8], that in daily life they “*sometimes appear suddenly like in the first exercise*” [P6, referring to task practice with **DEFAULT**], or that “*as soon as you click it is over*”²⁶ [P11].

2 participants explicitly referred to the unpredictability of pop-ups in itself, notably that they “*did not clearly understand the pattern of their appearance, and as a result it was more disturbing and frustrating*” [P6] and that it was “*very frustrating when you do not know when one might appear and disappear instantly*” [P17]. As a side-note on pop-up interferences “in the wild”, 7 participants volunteered that they try as hard as they can to forbid all pop-up windows in their daily lives, deactivating them everywhere possible and using pop-up blockers online; three of them even deactivated every notification on their smartphones, and one simply does not have internet on their phone to avoid having too many notifications.

3.4.3 Addressing Interferences Overall

We then investigated whether the proposed pop-up window **BEHAVIORS** successfully helped reduce the occurrences of interaction interferences or mitigate their effects.

From secondary task results

Part of the answer can be found by observing the proportion of *Interfering* pop-ups that were closed, as illustrated in Figure 3.15. We found a significant effect of **BEHAVIOR** on the percentage of popups closed in *Interfering* trials ($F_{1.49,28.29} = 0.81, p < .001$). Both **DEFAULT** and **FREEZE_{BLOCK}** (average pop-ups closed 100%, $CI = [100, 100]$) always required participants to close the pop-up windows before accessing the targets, and both were found significantly different from **AWAY_{DISPLACE}** (33.5%, $CI = [17.5, 53]$), **DELAY_{POSTPONE}** (36%, $CI = [19.5, 55]$) and **HOLEOVER_{THROUGH}** (42.5%, $CI = [26.61, 60]$). No significant difference was found between **DEFAULT** and **FREEZE_{BLOCK}**, nor between **HOLEOVER_{THROUGH}**, **DELAY_{POSTPONE}**, and **AWAY_{DISPLACE}**.

These results are consistent with the nature of each **BEHAVIORS**, highlighting two clear groups: a first group wherein every **BEHAVIOR** allows would-be interfered clicks to reach the target: by keeping the pop-up away (**DELAY_{POSTPONE}** and **AWAY_{DISPLACE}**) or by passing the click event through (**HOLEOVER_{THROUGH}**); and a second group that capture and interpret that click, either normally (**DEFAULT**) or with a tolerance for interferences (**FREEZE_{BLOCK}**). Note that while a **FREEZE_{BLOCK}** window does need to be closed to access the underlying target, the **FREEZE_{BLOCK}** behavior can effectively prevent it to be closed (or interacted with) by accident.

We did not analyze the *time* it took participants to close pop-up windows in *Interfering* trials, because whether a pop-up even needs to be closed is strongly dependent on the tested **BEHAVIOR**, as demonstrated above. Participants closed the pop-ups less often with **AWAY_{DISPLACE}**, **DELAY_{POSTPONE}**, and **HOLEOVER_{THROUGH}** than with **DEFAULT** and **FREEZE_{BLOCK}**, so the analysis would be run on strongly unbalanced population sizes. Plus, these data-points could be biased towards unusual or user-specific behaviors (for instance, some participants felt like closing pop-up windows before answering the four-choice question even when they knew the answer with confidence). We present in Subsection 3.4.4 an analysis of overall trial completion time.

From secondary task answers

Success in answering the secondary task, in comparison with the **DEFAULT** condition, is another indicator of interference-mitigation success.

We found significant effects of **BEHAVIOR** ($F_{1.69,32.04} = 13.81, p < .001$), **TRIALTYPE** ($F_{1,19} = 6.55, p < .05$), and **BEHAVIOR** $\times$ **TRIALTYPE** ($F_{1.92,36.55} = 11.51, p < .001$) on

²⁶Quotations with an asterisk were translated from French.

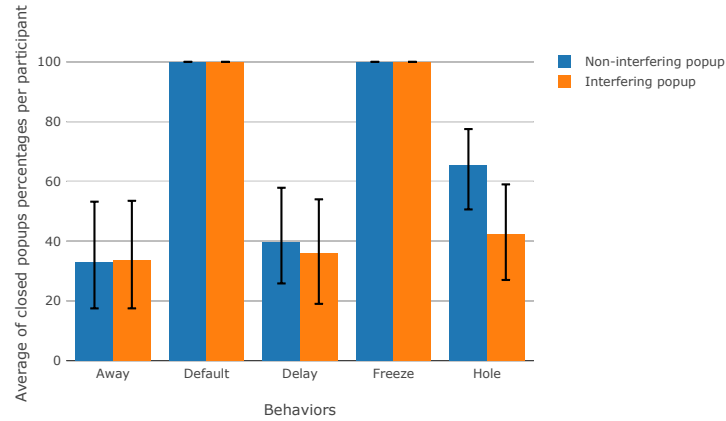


FIGURE 3.15: Percentages of closed pop-up windows as a function of BEHAVIOR and TRIALTYPE.

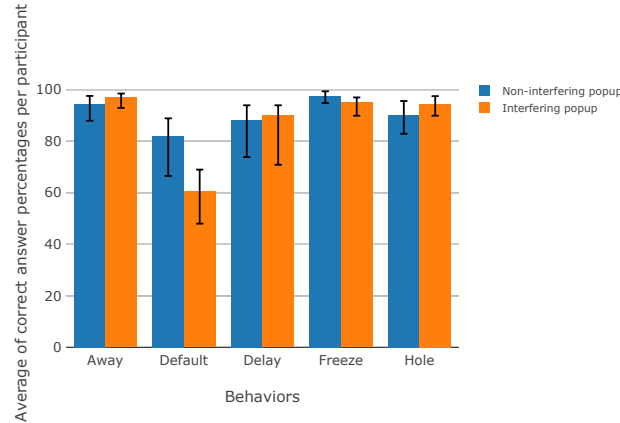


FIGURE 3.16: Percentages of correct answers to the secondary task as a function of BEHAVIOR and TRIALTYPE.

the average percentage of **correct answers**, illustrated in Figure 3.16. TRIALTYPE only affects that percentage with **DEFAULT** ($p < 0.001$), with *Interfering* trials (60.5%, $CI = [48.39, 69.5]$) having fewer correct answers on average than *Non-Interfering* (82.04%, $CI = [67.56, 89.03]$).

Regarding BEHAVIORS alone, post-hoc tests revealed that participants selected significantly fewer correct answers with **DEFAULT** (70.17%, $CI = [54.95, 76.16]$) than with **AWAY_{DISPLACE}** (95.97%, $CI = [90.67, 98.11]$), **FREEZE_{BLOCK}** (95.95%, $CI = [93.75, 97.7]$), **HOLEOVER_{THROUGH}** (92.69%, $CI = [87.78, 95.86]$), and **DELAY_{POSTPONE}** (89.26%, $CI = [74.34, 95.16]$), all $p < 0.05$.

These significant differences are maintained when we only consider *Interfering* trials ($F_{2.04, 38.7} = 20.97$, $p < .001$), with **DEFAULT** having significantly fewer correct answers (60.5%, $CI = [48, 69]$) than **AWAY_{DISPLACE}** (97%, $CI = [93.06, 98.5]$), **FREEZE_{BLOCK}** (95%, $CI = [90.5, 97.5]$), **HOLEOVER_{THROUGH}** (94.5%, $CI = [88.5, 97]$), and **DELAY_{POSTPONE}** (90%, $CI = [70.5, 96]$), all $p < 0.05$.

Conversely, when only considering *Non-Interfering* trials, and even though the analysis of variance revealed a significant effect of BEHAVIOR ($F_{1.81, 34.4} = 3.898$, $p < .05$), the post-hoc Tukey tests found no significant difference between any of the individual BEHAVIORS. The opposite effects (**DEFAULT** $\gg$ other BEHAVIORS) were found for the proportions of “**I don’t know**” answers. No significant effect of BEHAVIOR nor TRIALTYPE was found on the

percentage of **wrong answers**.

Participants provided at least about 30 pp²⁷ more correct answers in *Interfering* trials with the candidate BEHAVIORS (see Figure 3.16), suggesting that they successfully mitigated the functional impact of interferences overall.

From subjective feedback

Friedman tests revealed no significant effect of BEHAVIOR on any of our Likert-scale questions. This is interesting in contrast to the results of the secondary task, which would tend to indicate that pop-up window BEHAVIORS helped participants avoid closing the *Interfering* pop-up windows too quickly. This can have several explanations: considering the relative rarity of *Interfering* pop-up trials, the Likert responses could reflect the overall tasks operationalization more than the BEHAVIORS themselves. Another possibility is that the interference-related advantage brought by the BEHAVIORS is counterbalanced by negative side-effects when they operate. We however take note that our BEHAVIORS do not seem, as a whole, to worsen user experience.

While we did not find a statistical difference between participants' responses to these statements, open comments suggest on the other hand that some participants still felt differences compared to DEFAULT. More precisely, 20 comments from 9 participants mentioned interference-like issues, 8 among which, written by 4 participants, compared specific BEHAVIORS to occurrences or consequences of interaction interferences.

For instance, regarding AWAY_{DISPLACE}, P11 reported that the pop-up window had “a lot less chances of being **closed by accident**. [They] have seen all the popups without being disturbed, except one that [they] noticed only once [they were] asked for the content.” P17 stated that “When popups appeared far from the mouse it was not that bad, but other times [they] would **close the popup too fast**.”

Regarding DELAY_{POSTPONE}, P4 reported they were “consistently able to read the content of the popups. [They] never **closed them by mistake**; rather, they were closed automatically by the system after telling what the popup said. [P4] could read the popup every time, because it was still open.” P6 stated that “having a delay is interesting, it feels less aggressive and [they] feel more comfortable dealing with [pop-up windows].”.

With regard to FREEZE_{BLOCK}, P6 reported that “Having the freeze saved [them] from **clicking too fast and not seeing the pop up**, but [they] would have [clicked it] without it.”.

P6 also concluded that, overall, “Except “DEFAULT” [they] did not have any interference.”.

3.4.4 Side-effects of BEHAVIORS

Interaction interferences can be extremely frustrating, but remain relatively infrequent [162]. While it is desirable to minimize their occurrences or consequences, it should not be at the expense of cumbersome or frustrating default GUI behavior. We therefore investigated their impact on *Non-Interfering* trials specifically, using *Normal* trials as a baseline for standard interaction.

From user inputs

Figure 3.17 illustrates overall trial duration in blue, and target picking duration (*i.e.* between the beginning of the trial and the first click on the primary target) in orange, per BEHAVIOR, for *Normal* and *Non-Interfering* trials.

With *Normal* trials, we found no significant effect of BEHAVIOR on either total trial time or target picking time (all $p < 0.05$). This is a sanity check that pop-up-less trials were not affected by pop-up behaviors.

In *Non-Interfering* trials, however, we found a significant effect of BEHAVIOR on total trial time ($F_{2,99,56.79} = 9.03, p < 0.001$), with significant differences between DELAY_{POSTPONE} (mean 2385 ms, $CI = [2088, 2710]$) and both DEFAULT (3179 ms, $CI = [2856, 3539]$) and FREEZE_{BLOCK} (3224 ms, $CI = [3014, 3450]$). We also found a significant effect of BEHAVIOR

²⁷“pp” is short for “Percentage Point”, *i.e.* the arithmetic difference of two percentages.

on target picking time ($F_{2.87,54.51} = 29.74$, $p < 0.001$), with significant differences between, on the one hand, **DEFAULT** (2385 ms, $CI = [2088, 2679]$) and **FREEZE_{BLOCK}** (2438 ms, $CI = [2242, 2597]$), and on the other hand, **AWAY_{DISPLACE}** (1627 ms, $CI = [1350, 2034]$), **DELAY_{POSTPONE}** (1252 ms, $CI = [1124, 1511]$), and **HOLEOVER_{THROUGH}** (1744 ms, $CI = [1533, 2002]$). These results indicate neutral to beneficial impacts of the candidate BEHAVIORS on the completion times of the primary task when compared to the **DEFAULT** condition in *Non-Interfering* trials, i.e. when pop-up windows had little chance to trigger an interference. Even though the candidate BEHAVIORS are not designed to improve *Non-Interfering* situations, **HOLEOVER_{THROUGH}**, **AWAY_{DISPLACE}**, and **DELAY_{POSTPONE}** made the primary task faster, probably because they eliminated the need to close the pop-up windows. This is likely a consequence of our task operationalization, however, and we do not conclude that these BEHAVIORS could improve overall time performance in real-life situations. Nevertheless, these results do indicate that our pop-window BEHAVIORS do not impair time performance.

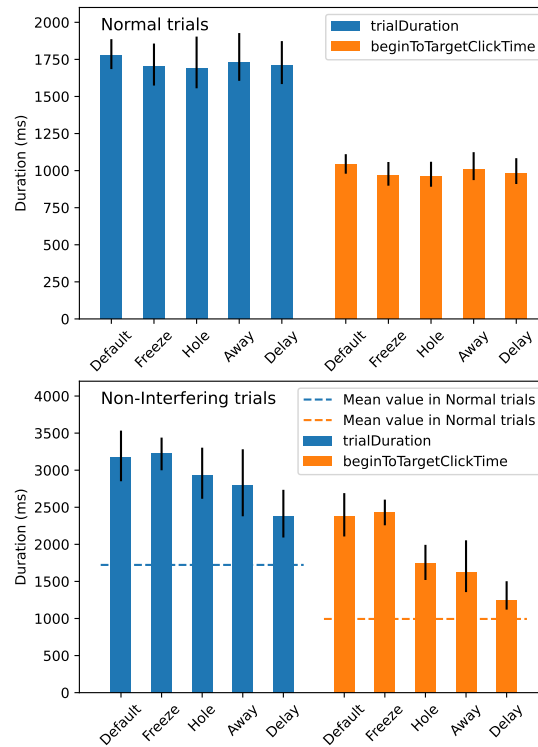


FIGURE 3.17: Total trial duration (in blue) and target acquisition duration (in orange) for each BEHAVIOR, in the *Normal* trials (top graph) and in the *Non-Interfering* trials (bottom).

From subjective feedback

Participants' preference ranking of the five BEHAVIORS are illustrated in Figure 3.18. A Friedman test found a significant effect of BEHAVIOR on these rankings ($\chi^2(4) = 15.56$, $p < 0.005$), and post-hoc tests identified significant differences between **DEFAULT** (median rank 4.5) and **AWAY_{DISPLACE}** (2), **DELAY_{POSTPONE}** (2.5), and **FREEZE_{BLOCK}** (2), as well as between **HOLEOVER_{THROUGH}** (4) and **AWAY_{DISPLACE}** and **FREEZE_{BLOCK}** (all $p < 0.05$ or less). The open comments that we collected draw a similar picture, with more detail about which aspect of each

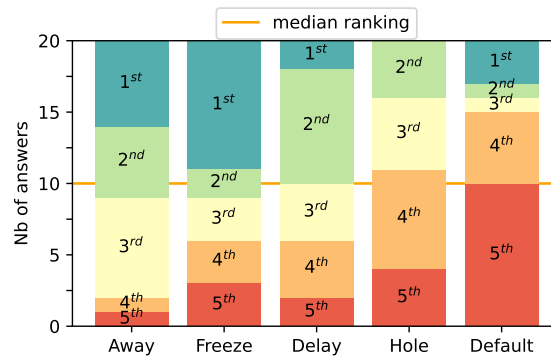


FIGURE 3.18: Stacked histogram of participant rankings per BEHAVIOR.

BEHAVIOR was appreciated, which one was disliked, and what can be done to improve some of them. Despite having found no clear effect of BEHAVIORS on any of our Likert questionnaires, we identified a number of usability and design insights from participants' numerous open comments regarding what they did or did not appreciate with each technique.

DELAY_{POSTPONE} **DELAY_{POSTPONE}** was ranked consistently across the range (first and last twice, median 2.5) and obtained mixed comments. Five participants expressed an appreciation for its working principle, for instance reporting *"never closing pop-ups by mistake"* [P4] or finding it *"more comfortable"* [P6]. However, we also identified two consistent caveats.

First, displaying the pop-up window immediately after the first click could give participants the impression that its appearance was somehow triggered by their action, or that they clicked the pop-up instead of the target by a few milliseconds, sometimes even prompting them to quickly re-click the pop-up window (e.g. *"DELAY_{POSTPONE}[was] closed a lot by accident"** [P14], *"it does not interrupt your task at the beginning, but it creates doubts if it is an error you did; maybe you did not [click] the target and left it behind, or it is the pop up showing up"* [P3]). As a consequence, some participants reported an overall sentiment of unpredictability [P16], or split-second impressions of having closed a pop-up by accident even though they had not. This was however not reflected in the amount of closed pop-up windows (Figure 3.15).

Second, a number of participants (3) explicitly specified that their positive opinion of **DELAY_{POSTPONE}** could be conditional to their activity, depending in particular on the relative importance of the task at hand vs. the information on the pop-up window (e.g. *"[DELAY_{POSTPONE}] becomes irritating when we have a task to do, but if we have to read any information, this is the best"* [P10]).

These comments indicate that users could benefit from additional delay between the click and the pop-up's appearance, for overall usability if not directly for performance purposes. That would however postpone the display of important information even further, which some participants hypothesized could become problematic depending on the task at hand. Another possibility would be to design **DELAY_{POSTPONE}**-specific appearance feedback (e.g. fading-in) that would indicate that a given pop-up was waiting to be displayed. These design adjustments need to be assessed in further studies.

AWAY_{DISPLACE} **AWAY_{DISPLACE}** received among the best rankings overall: more than half the participants ranked it first or second, and only two participants ranked it 4th or 5th (median 2). It also received a number of positive feedback (21 comments, 14 participants), praising e.g. that *"one can stay focused on what they are doing, while seeing what else is happening on the screen"** [P5], that **AWAY_{DISPLACE}** *"has a lot less chances of being closed by accident"* [P11], or that **AWAY_{DISPLACE}** was less/least annoying BEHAVIOR [P1, P5, P16]. As downsides, 3 participants found that the location of the pop-up felt random and unpredictable; and 3 participants mentioned that in a real setting it could become tedious to close them if they ended up too far (e.g. *"if I had to actually close each popup I would be annoyed"* [P17]).

Overall the open comments about **AWAY_{DISPLACE}** were primarily positive, the main issue being related to the unpredictable location of the pop-up window. This is very encouraging considering that our pop-up-locating algorithm was a first version and likely improvable, e.g. making the pop-up appear closer to the cursor (although at the risk of triggering more interferences) or at a designated location, as some participants suggested.

FREEZE_{BLOCK} Despite its longer additional delay (Figure 3.17) and the obligation to close the pop-ups, **FREEZE_{BLOCK}** received numerous positive comments (23, from 13 participants) and received among the best rankings (median 2) with the highest number of ‘first’ (9 participants). Among its appreciated features, it increased confidence and control (e.g. “*I knew I wasn’t going to miss the information*” [P19], “*Way more managable [than **DEFAULT**] [...] I could be fearless to try to do [the task] fast*” [P3]) and was “*very easy to follow*” [P10]. We collected 3 negative comments from 3 participants, two of them referring to the inherently blocking nature of **FREEZE_{BLOCK}**’s behavior (the last one did not specify what they disliked).

In terms of both positive and negative comments, **FREEZE_{BLOCK}** was a participants’ favorite even though its interrupting principle and some of its quantitative effects could have hinted otherwise. One participant even mentioned that they preferred **FREEZE_{BLOCK}** “*because it changes color*” [P8], which adds to our primary conclusions that richer visual feedback could improve our **BEHAVIORS**.

HOLEOVER_{THROUGH} Comments about **HOLEOVER_{THROUGH}** were mostly critical, in line with its ranking which was the worst among the candidate **BEHAVIORS** (median 4): more than half the participants ranked it 4th or 5th, and no-one ranked it first. Even its working principle was confusing to some (e.g. “*It felt slightly odd [...] to click through the popup*” [P17], “*I thought of my mom she wouldn’t have [understood]*” [P8], “*[it felt] unnatural to click through the windows*” [P11]) or even disturbing (“*This [behavior] seemed very disturbing to me, it took me a long time to get used to it*” [P5]). 2 participants felt that it was overly complicated and in particular required more focus than the other **BEHAVIORS** (e.g. “*Was hard at the beginning to deal with the pop up content, the number to grab and the hole animation at the same location on the screen*” [P6]). 2 participants specifically complained that they felt not in control of the **BEHAVIOR**’s duration, making them “*doubt in every click if [they were] clicking the target through the pop-up or clicking the pop-up*” [P2]. Finally, even its core goal was criticized, as 3 participants mentioned that **HOLEOVER_{THROUGH}** made them click pop-up windows when they did not intend to.

We intended the **HOLEOVER_{THROUGH}** technique as a post-hoc indication that a potentially interfered click had been sent to the target underneath, not as a “mode” during which users should interact while they can. However, during our post-experiment discussions we realized that some participants tried to actively aim at the target through the hole while the **BEHAVIOR** was still active, which would explain some comments about having insufficient control over its duration and about adding to the tasks’ complexity.

DEFAULT aside, **HOLEOVER_{THROUGH}** was the least appreciated **BEHAVIOR** evaluated in our study. There seems to have been a miscommunication when we explained what the behavior does and to what purpose. Similar to the previously discussed **BEHAVIORS**, we consider it possible that adjusting its visual feedback may better convey what its purpose and meaning are (and aren’t!).

DEFAULT Our experiment’s baseline condition received the worst ratings of all **BEHAVIORS** (median 4.5). It was ranked 5th by half the participants and 4th by five more. Being the condition “without any special behaviour” (Table 3.3), the comments about **DEFAULT** were more about pop-ups and interferences in general, or used as a reference to explain why a given candidate technique was preferred. Three participants felt like **DEFAULT** was the most understandable or predictable behavior. On the other hand, the simplicity of **DEFAULT** also has its upsides, as one participant noted that “[With **DEFAULT**] *you don’t feel guilty. You cannot undo what the system is doing*” [P18].

3.5 Discussion

In the following, we further discuss these results and elaborate on the limitations of our work.

3.5.1 Overall Utility of *Naïve* BEHAVIORS

The results of our experiment suggest that the candidate behaviors can already reduce the occurrences of (some) pointing-based interferences without impeding the user’s ability to perceive the content of the GUI change. Three of them, **FREEZE_{BLOCK}**, **AWAY_{DISPLACE}**, and **DELAY_{POSTPONE}**, were consistently preferred over the **DEFAULT** behavior (Figure 3.18), and the comments we gathered from our post-experiment questionnaires hint at a number of simple feedback refinements that could improve user satisfaction even further.

However, we acknowledge that a one-hour experiment can be a limited platform to study such an un-systematic phenomenon. First, despite our efforts to make them as infrequent as possible during the study, there is a risk that the unusually high rate of interferences affected the participants’ subjective answers—positively or negatively. Second, even though we tried to design *Naïve* behaviors that operate as innocuously as possible, there exists a risk that their systematic nature may cause more harm in day-to-day use than in the confines of an experiment clearly designed to characterize them²⁸. For these reason, despite rather positive results with *Naïve* behaviors, we still plan to design and evaluate more “intelligent” approaches that aim to *Detect* or even *Predict* interferences in order to make unusual behaviors even rarer.

3.5.2 Feasibility and Technical Requirements

The four candidate behaviors evaluated in our study have technical requirements that we consider among the easiest to fulfill. **FREEZE_{BLOCK}** and **HOLEOVER_{THROUGH}** require the ability to change the behavior of graphics elements, both in terms of visual appearance and of events handling. This can be implemented relatively easily, *e.g.* on webpages with the `MutationObserver`²⁹, or as an additional default behavior in most window managers. **FREEZE_{BLOCK}** can even benefit from existing window states such as Qt’s `enabled` property or JavaFX’s `disabled` property, which can be changed with simple method calls. The specific visual feedback of **HOLEOVER_{THROUGH}** could demand some extra work, but its value is anyway put to question by our participants’ feedback; a quieter implementation of the **MAPPING** behavior type where events are sent to the element below without visual feedback would be even simpler to implement, but at the possible cost of confusion as to whether the event was interpreted as intended.

AWAY_{DISPLACE} and **DELAY_{POSTPONE}** require the ability to monitor user events, which can already be done at system level with most operating systems via global mouse listeners, but also to control the default time or position at which the change occurs. The latter two can be more challenging to implement on top of an existing application since GUI toolkits rarely expose event handlers for GUI elements that are *about* to change, and waiting for a change to occur before immediately altering it can cause problematic visual flicker. Such solutions need to be investigated on a toolkit-by-toolkit basis for feasibility.

Connecting these requirements to our study results, we expected to find a correlation between the satisfaction with a BEHAVIOR and its *Technical Requirements*, *i.e.* the most appreciated behaviors would be the most technically demanding. We were therefore surprised that **FREEZE_{BLOCK}**, which we considered our most basic BEHAVIOR (and the only one unable to prevent an interference, only its worst consequences) ended up being ranked first the most. Even though our participant comments offer promising leads to improve **DELAY_{POSTPONE}**, **AWAY_{DISPLACE}** and **HOLEOVER_{THROUGH}**, this highlights the interesting possibility

²⁸This in fact belongs to a more general paradox when investigating methods to prevent unwanted GUI behaviors: a perfect solution might prove imperceptible to users, because it is difficult to notice the absence of an already rare, unwanted event.

²⁹<https://developer.mozilla.org/en-US/docs/Web/API/MutationObserver>

that a simple and predictable solution could be preferred even when its best-case consequences are worse. Then again, user feedback may differ in day-to-day use when users are not specifically directed to think about these behaviors, and where interfering pop-ups (and pop-ups in general) are less of a constant threat.

Design adjustments could improve the tested behaviors, but must however be done carefully. Too little feedback can disrupt the user – as with **DELAY_{POSTPONE}** – but too much feedback can also be overwhelming – as with **HOLEOVER_{THROUGH}**. To the user, realizing that an interface change has occurred at the very moment that they have generated an input is a mental effort in itself, whether the whole situation triggered an interference or not. When deploying interference countermeasures, we potentially add to this effort the need to understand *where* the click went, since the system’s interpretation no longer follows the chronological order of events. This may explain the success of **AWAY_{DISPLACE}**: by eliminating the co-location of events, the effect of the input is no longer temporally ambivalent.

In terms of user expectations, respecting the linearity of events can also lighten this mental effort of realizing what one’s input did in an interference. **HOLEOVER_{THROUGH}** creates a conflict in this matter: the user sees a popup appear, then their click happens, and when they expect the click to impact the popup, it actually impacts the layer underneath. This lack of causal logic can explain participants feeling confused or disturbed by the **BEHAVIOR**. **HOLEOVER_{THROUGH}** creates an error as defined in this thesis, by not respecting user’s linearity expectations. **FREEZE_{BLOCK}** also lets the interference happen (the user clicks where the popup just appeared), but blocking the click does not challenge the order in which events occurred.

More work is needed to single out a set of “best” behaviors, but our study demonstrated that simple and arguably achievable behaviors can already decrease the occurrences of pointing-based interferences without harming GUI readability or user experience. Future work should investigate the implementation of these **BEHAVIORS** in real systems, and its implications on existing window managers, notifications, and input events handlers.

As a concluding note, we had to define many of our **BEHAVIORS’** parameters by hand using pilot studies: their durations, their various triggers, the location of the appearing window with **AWAY_{DISPLACE}**, etc.; and we could only test *Naïve* behaviors, which leaves many interesting possibilities for all of these parameters. Our controlled experiment was designed primarily to establish the feasibility, performance, and limitations of the principles behind our behavior types; as such, there remains a lot to do before we can present a definitive set of techniques that can adapt to any task, context, device, etc. Finer tuning of these numerous relevant parameters would have generated too many conditions to test in this exploratory work, but it is a very promising next step in our investigation of this problem space.

3.5.3 Applicability to Other Types of Interferences

Schmid et al. reported a number of examples of real-life interferences provided by the participants of an online survey [162]. Most of these examples involved a pointing action as triggering event, followed in frequency by keyboard events, and a minority of gesture events. We devised our design space of solutions to be input-agnostic, but had to settle on a single type of interference-triggering GUI events in our experiment to keep its duration manageable. We chose pop-up windows as they are among the most common and recognizable interferences around [162]. We also expect that it is as good a stimulus as any other GUI event to trigger *pointing-based* interferences, since the physiological ability to inhibit a click likely does not depend on the nature of the change. However, that nature does affect the GUI behaviors that can be applied to avoid it or to alleviate its consequences, and some of the resulting **BEHAVIORS** would arguably only apply to that category of interaction interferences: what could it mean to **HOLEOVER** an app taking the focus, when the target element is not hidden as a result? or to apply **AWAY** to word suggestion updates of a gesture keyboard?

We however see value in many of the design space’s categories of behaviors and requirements for other types of interaction interferences. We will now discuss how our results and *Pop-up*-specific **BEHAVIORS** can generalize to other forms of pointing-based interferences, structured around the four examples given in the introduction. In them, the intended target of a pointing action either:

- became **occluded** (e.g. a notification appeared above the target right before the click),

- **moved** (e.g. trying to click a link when the webpage unexpectedly updates),
- **changed** (e.g. the word suggestion above a gesture keyboard changes just before the tap),
- or **disappeared** (e.g. a notification disappears right before the user clicks to close it).

DISPLACE

The DISPLACE behavior type, of which **AWAY_{DISPLACE}** is derived, can apply to any interference whose trigger is location-dependent. Its implementation needs to be case-specific, however, compared to *Pop-up*-style interferences.

The “**occluded target**” situation was operationalized as a pop-up window in our study, but other examples such as notification panels can also be displaced when an interference is expected to occur. In situations when whole applications take the foreground and focus, e.g. as a result of a planned prompt, it may be complicated to move a full-screen app out of the way of the cursor.

When the intended click target **moves**, it would seem strange to want to DISPLACE it some more. But it can make sense in situations where the viewport of the target can itself be manipulated.

Take the example of an emails list that is updated by the arrival of a new email, right as the user was about to click another email. The update typically shifts all emails down by one slot, resulting in the user clicking the email (previously) above their target. **AWAY** cannot be directly applied here, but a DISPLACE-type of behavior could consist in appending the new email to the top of the list without automatically shifting/scrolling its contents.

DISPLACE behaviors make less sense for situations where the intended target is **updated**, since the click will still likely occur *somewhere* unintended. The same goes for **disappearing** targets.

POSTPONE

The POSTPONE behavior type, of which **DELAY_{POSTPONE}** is derived, is rather universal in its application because interaction interferences themselves are essentially temporal phenomena. Any potentially interfering interface change (**occlusion, move, change, disappearance**) can be postponed to a safer time, assuming that interference risks can be accurately estimated—and that the initial timing of said change is not important. The real challenge here lies in estimating *when* and *how long* to postpone an update, as our study already showed issues with the simplest approach to apply the change immediately after we estimate the risk gone: participants felt like their click caused the window appearance. In the other extreme, a too-cautious approach can cause extended delays. This begs the question of how to provide visual feedback to the postponing behavior, or whether the system should provide visual feedback at all.

BLOCK

Similarly, the principle of the BLOCK behavior type (of which **FREEZE_{BLOCK}** is derived) can be applied in seemingly all situations (**occlusion, move, change, disappearance**), because purposefully ignoring input events that are possibly interfered with does not depend on the nature of the interference. It however requires to accurately assess the risks of imminent interferences lest GUI elements be too-frequently greyed out, which may appear random in the user’s perspective. Typically, applying the *Naive* **FREEZE_{BLOCK}** behavior to every updated element in an interface could prove disturbing if we maintain the greyed-out visual feedback. In the “**disappear**” case, **BLOCK**ing all risks of interference might mean freezing the entire interface unless a precise scene graph is exposed.

MAPPING

The MAPPING behavior type, which is the superset of THROUGH from which **HOLEOVER**_{THROUGH} was derived, can be seen as the equivalent of POSTPONE when the interfering GUI change could not be prevented: by sending the interfered-with event to its (estimated) original target, it simulates a situation in which the interface change had not yet occurred, despite what the user perceives. Like POSTPONE, its principle is not location-dependent and therefore can be applied to seemingly all categories of interferences (**occlusion**, **move**, **change**, **disappearance**)—provided that there exists a history of recent interface states, which is a significantly constraining requirement in today’s systems.

Its subset, THROUGH, simplifies that technical constraint by assuming that the interference was location-dependent but also in the foreground of the intended target. While simpler, it almost exclusively apply when the intended click target is **occluded** by another visual element.

It can however remain complex to implement, *e.g.* when multi-window apps take the foreground and focus, and the “hole” needs to reach down several layers. In all other cases, **move**, **change**, and **disappearance**, the **HOLE** technique cannot be applied as-is but might be extended as “worm-holes” showing the recent past around the cursor to indicate that the event was sent to its initial target.

Correct

Finally, every behavior type in the *Correct* category, *i.e.* AUTOMATIC, PROMPT and INFORM, are independent on the type of interference since they can only be applied after both the interface change and the user event (Figure 3.2). Their interpretative and technical constraints remain the same regardless of interference type.

Part of our future work will consist in investigating the existence of other behavior types that we could have missed in our design space.

3.5.4 Overriding the Behaviors

While we focused on minimizing the occurrences of pop-up interferences and mitigating their effects, there exist very specific situations where one may prefer the interference *not* to be handled. This can be the case when the user knows that a certain event will happen at a certain on-screen location, for instance that a notification window will appear, and barrage-click on that area in anticipation.

Our current point of view is that such situations remain exceptional – even compared to interferences themselves – and do not justify the current state of things where GUI toolkits implicitly assume that every user input is fully informed and faithfully reflects the user’s intentions. Moreover, it is interesting to note that the proposed BEHAVIORS are not all affected in the same way in this situation. Typically, while **DELAY**_{POSTPONE} might purely hold the notification spawn until the end of the click barrage, other like **FREEZE**_{BLOCK} might easily cope with these situations since the notification would be activated after the second click. In any case, the negative side-effects of anti-interference GUI behaviors will need to be thoroughly explored in real, long-term field studies in which participants can experience them realistically. This is part of our future works, using or adapting monitoring tools such as [83, 176].

3.6 Summary and Conclusion

In this chapter we investigate how Graphical User Interfaces, old and new, could minimize the occurrences and effects of interaction interferences, a phenomenon triggering errors in terms of user expectations.

We describe a design space of possible solutions that differ in terms of types of behaviors, *i.e.* whether they *Prevent*, *Avoid*, or *Correct* the interferences, and in terms of their interpretative and technical requirements. From this space, we designed and instantiated

four behaviors for appearing (or *Pop-up*-style) GUI element that we compared to the default window behavior in a controlled experiment with 20 participants. Our results indicate that simple, low-requirement behaviors can already prevent or mitigate most *Pop-up*-style interferences without harming the readability of the appearing element. Participants majoritarily rated the default condition (*i.e.* no mitigation behavior at all) the worst, and their feedback offered valuable hints on how to improve existing behaviors. They also hint at a correlation between user preference and the predictability of the GUI behavior, sometimes even at the cost of best-case outcomes. We then discuss how our design space and behaviors can generalize from appearing GUI elements to all types of interaction interferences.

Our next steps will involve tweaking the feedbacks of our existing behaviors, exploring methods to detect and predict interferences in order to only apply behaviors in “risky” situations, and exploring the technical feasibility of these solutions in real systems and GUI toolkits. In particular, we will propose updated models of common UI patterns and models (MVC, ECS, finite state machines, etc.) that can expose and allow the *Technical requirements* that we list as a dimension of our design space.

This study illustrates how an erroneous phenomenon can be solved by adapting the system to the user’s expectations. The user is surprised and frustrated, not because the system behaves in an unwanted manner, but because they are caught off guard by the the sequence of events, that they physiologically cannot prevent. Interaction interferences show that “user expectations” do not only include wanted behavior, but also an overall fluid use. This notion, and its impact on the definition of error in this thesis, is discussed in the next chapter.

Chapter 4

The stray from user expectations and its revealing possibilities

Classical approaches of error define the concept in regard to “what the system is supposed to do”. These specifications are decided on the system’s side, by its conceptors and designers. Moreover, because the subject has been thoroughly investigated in critical and industrial settings, there exists an impulse, even in HCI, to look for a root cause to fix. For Reason, Carthey, and Leval [154], there is also an existing blame bias among human beings (as described in subsection 1.2.3 of the first chapter), that encourages people to look for and perceive the existence of a guilty party.

This thesis changes the focal and centers error around user experiences, and more specifically around user expectations. This chapter presents the consequences of taking on such an approach, and discusses its implications in regard to the different studies presented in the preceding chapters.

4.1 Redefining errors as user expectation violations

An error in this thesis is an incident where user’s expectations and wants on the interaction are disregarded, transgressed, violated or in less strong terms where they are invalidated. Defining the phenomenon as such has direct consequences over which incidents will be considered errors, which will not, and how this discrimination will be decided. This section puts forward the characteristics of this new definition, and argues puts in perspective the classical one; even when defined from the system specifications, error is a subjective and negotiated concept.

4.1.1 What does a user-centered concept of error entail?

Usual approaches of error define the notion as a stray from specifications. In Safety Sciences, both the *Old View* and *New View*, if they differ in responsibility attribution, agree on this: the scope of what constitutes an error is determined by the system. In Engineering as well, Avizienis et al. [9] defines an error as the system not fulfilling its function *i.e.*, the specifications determined by the designers. This taxonomy influenced a lot of later works employing the same approach [132, 14]. Some studies in HCI shed a light on the possible gaps in the definition of error between different designers, or between designers and users [35]. Overall however, the power of determining the extent of that definition falls on the system’s side. There may be a negotiation, but mostly held by designers and developers and not by users.

One might think that the mere definition of error is not that impactful. If some phenomena like interaction interferences are not labeled as errors, but as “problems”, does it matter? If interruptions are designated as “inconveniences”? In computer science, the word error implies something that needs to be fixed. This may be due to the fact that we mostly interpret it from the point of view of the system. An error is, to put it bluntly, something binary, either present or not. It is obvious and non-negotiable. The system’s functions should provide the reference of what should be, from which an error diverges.

Vocabulary matters: As Chilana, Ko, and Wobbrock [35]’s study shows, the description of bugs impacts their potentiality to be fixed. One can imagine that a phenomenon described as a disruption would be less likely to be treated as a serious problem, whereas something accepted as a bug, however, will receive a high amount of attention and care. A problem or a nuisance will likely only be mitigated or managed, when a bug or an error will be fixed.

This is not to say that all problems should be considered of the same criticality. Of course, some issues are more serious and urgent than others: it makes sense to prioritize them. And in some cases, the system specifications take precedence in the definition of that priority, in critical systems in particular. However, I argue that in most everyday life contexts, criticality would be better determined by the users. They are, after all, the ones facing the consequences of negative incidents.

In this work, an error is defined as a stray from user expectation, and by extension, user experience define its criticality. This leads to 3 major developments in the definition of error.

First, *the consequences of a bad incident encompasses all of user experience*. They are not only material, but also subjective and emotional. One cannot assess such an error only looking at the system. An examination of the user gives access to potential information relating to context. Their lived experience is very telling of the stakes put in the interaction: the importance and difficulty of the task, their state of mind, their relationship with the tool etc. The workshops presented in section 2.2 are very telling in that aspect: even when they were not prompted to share those implications, participants still brought them up in their depictions of incidents.

This experience does not necessarily fit into the current evaluation standards of mental load or usability (see section 4.2.1 below).

Second, *the criticality of an incident is determined from the user perspective*. Their context, task, and personal history define the importance of the interaction. Specifically, this means that the definition of criticality can challenge the one that would be set forth by its designer. An otherwise “grave” incident happening to a device or a task that the user does not care about is not critical from their point of view. For instance, in the workshops, a participant shared not doing a backup of their pictures regularly, because the pictures themselves were disposable to them; the potential data loss would not be critical to them (“I take really useless photos that serve no purpose. I take 40 photos of my cat every day. I tell myself I’m not going to save them, there’s no point. I’ll take another 40 tomorrow.”* [P8_{W2}]). Conversely, a “minor” incident happening during an important activity that they have spent a lot of time on might be very critical to them.

Finally, since they are assessed according to the user, at a specific time and place, *errors are variable*. Every error is unique in its user response, and is therefore treated as such in this framework. Errors are personal and specific; they might vary from one person to the next, and from one instance to another. Some incidents will be problematic for some users, and not for others. The need for an error to be corrected is dependent, not on the phenomenon, but on its assessment by the person affected by it. This implies that criticality is not a function of error type, but is categorized independently. A specific phenomenon is best investigated in different contexts and with different users. The studies in chapter 2 show for instance that the intensity of blame can very much vary within the same type of incidents, as for FAIL-UNSAFE acts in the workshops.

Contrary to former approaches, an error as defined in this thesis does not take the system’s point of view into account, but only the user’s experience. As such, it is highly dependent on context and on the user. An error is not a system state gone wrong, but a user experience gone wrong. This allows some novel perspective on the study of what goes wrong in interaction, by allowing to define user-specific needs, or at least to consider them in the evaluation of a system.



FIGURE 4.1: User error drawn by Patte de Lou [133]

4.1.2 Errors and bugs as negotiated denominations

Errors and bugs are closely related to each other. A bug can be described as “an unexpected defect, fault, flaw, or imperfection” [120]. Bugs are specifically related to machines, even though the word can be used metaphorically when talking about people in difficulty.

Going back to Avizienis et al. [9], bugs would not be machine failures *i.e.* something that prevents the system from fulfilling its primary function. Instead, in this taxonomy, I would say they are machine faults or errors, depending on whether they are active or not.

Relating to the concept of error developed in this thesis, a bug is an error attributed to the system, that the user perceives as a bug.

For Vitali-Rosati [181], when a bug (or an error) happens, it presents an alternative. Instead of the expected result, something unexpected happens. The incident is admittedly unwanted, but does provide an idea of possibilities intersecting. From there, and upon user appreciation, the persistence of an error can become not only an expected event, but a wanted one.

This idea transpires very much in the expression “it’s not a bug, it’s a feature”. Sometimes, a bug becomes a proper functionality: using the Request library for Python for example, a bug makes it possible to skip certificate verification [85]. Most unexpectedly for its developers, it has become a popular by-default way of bypassing SSL³⁰ validation. The discussions on the issue, in Figure 4.2, show the discrepancy between “intended behavior” and what end users make of an unintentional feature. Generalizing from this, what a bug entails is dependent on our world view, on our *expectations* as users and designers. These expectations are not absolutely fixed however. To users in particular, they can change as possibilities open up through surprising behaviors of the system. Given the choice between the behavior they were waiting for and the actual response of the system, users get to determine what is the preferable behavior, *i.e.* what a bug is to them.

For developers, a bug is a stray from specifications [35]. As for users though, their definition of a bug can change if faced with alternatives that feel beneficial to them. Specifications are not set in stone as a whole. In Chilana, Ko, and Wobbrock [35]’s study on bug reports, bugs expressed as community expectation violations had greater chances to be fixed than if they were presented as strays from personal expectations. The description of a problem

³⁰for Secure Sockets Layer, a web security protocol based on encryption.

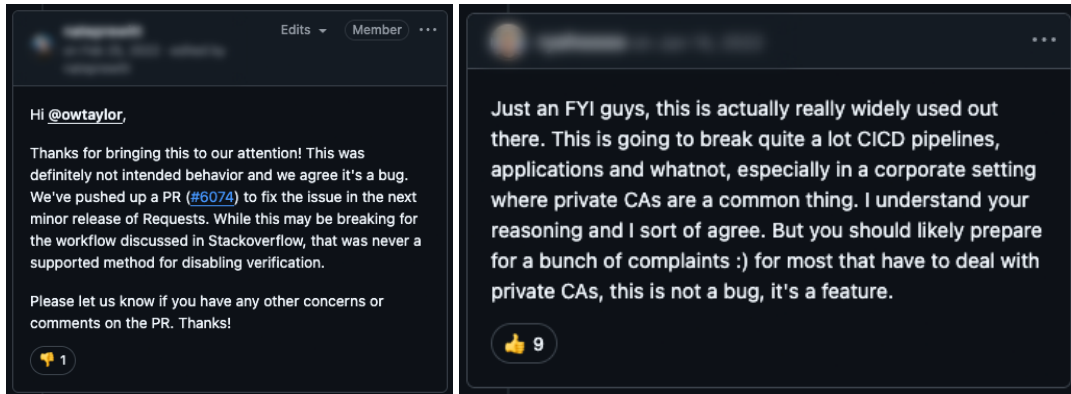


FIGURE 4.2: Responses to issue 6071 on the Request repository [85]

will more likely be considered as an error by developers if it concerns a group of people rather than an individual. By accepting new bug definitions, designers effectively change the specifications of the system to fit users' idea of an error.

What the system should be is dependent on the person making the assessment, but also on context [184]. This judgement of "what an error is" (and hence, "what the system should do") is always socially biased, be it from a user or a developer. Chilana, Ko, and Wobbrock [35] show that developers have in the end the power to make that call. They will not however necessarily be sensitive or preventive of matters like inclusivity. According to Widder and Goues [184], this has a consequence on the type of bugs that will be accepted as such by developers, and in turn corrected. To them, a system that is not inclusive of women or minorities is bugged, much like a system with a segmentation fault. For example, a robot interfering with wheelchair users' safe paths is an error, and requires fixing, even if it concerns very few people [50].

Widder and Goues [184] argue for a research that explicitly states a priori judgments about which bugs are worth fixing and why, so that inclusivity flaws can be noticed and explored later. Depending on who the designers and developers are, the problems to be fixed and their set priorities will be different. This is not a question of pointing fingers and devaluating their work, but simply pointing out their subjectivity, which we all inevitably have.

Even when defined as a stray from specification, the scope of the word "error" is not fixed. It is negotiated among peers: designers, developers reading bug reports and deciding which to fix and which to not, *i.e.* which are "real" errors. The line between error and functionality is traced by social factors and context, much like in the definition defended in this thesis.

When described as a stray from user expectation, the distinction of what an error is will vary from one user to another. Expectations, as for specifications, come from somewhere. They reveal specific interests, needs, not only from the people defining them (designers, users), but also from their environment and context.

4.1.3 The setting of wrong expectations

"C'est quoi par exemple un agent conversationnel? C'est un algorithme qui est capable de manipuler la langue naturelle et de la manipuler au même niveau qu'en parlant cette langue. Ça, c'est la définition. Donc, par exemple, ça me fait beaucoup rigoler tous les articles sur le fait que ChatGPT ne donne pas des informations justes, ou vraies. Bah, évidemment, il n'est pas fait pour. C'est comme si je disais, je ne sais pas, "cette voiture frit très mal les courgettes". Bah ouais, en effet."—Marcello Vitali-Rosati in [95]

"What is a conversational agent, for example? It's an algorithm that can manipulate natural language to the same level as speaking it. That's the definition. So, for example,

I get a big laugh out of all the articles about ChatGPT not giving accurate or true information. Well, obviously, it's not designed to. It's like saying, I don't know, "this car fries zucchini really badly". Well, yeah, indeed."—Translated quote

In La Porte and Cassar [95]'s podcast about bugs, the guest talks about critics made to LLMs not providing true informations. He explains that these critics do not make sense, as LLMs are not made to be reliable on that front. Their specifications confine them to be language manipulators, not to answer questions accurately. An LLM giving out false information is not an error, because it is just not part of their functions.

If we look at these AIs with the usual definition of error, Vitali-Rosati is right. If these systems are not designed to be factually accurate in the speech they provide, failing to do so is not an error. As language manipulators, an error from an LLM would be a spelling or grammatical mistake, or incoherent speech.

LLMs do not provide reliable information, as a car does not fry zucchini reliably; it is just not made to do so. I would argue however, that the difference is that a car does not lead people to think it can fry zucchini. The bodywork can get warm on a hot day, but otherwise nothing in its affordances or design cues points in that direction. ChatGPT on the other hand, generates speech that is confident and affirmative, encouraging users to trust its output. As a participant of the workshops (section 2.2) underlined, the LLM does not explicit where their proficiencies are: *"I asked it to do something it doesn't know how to do, but instead of telling me it doesn't know how to do it, it just did something stupid"** [P9_{W3}]. The interface also leans toward a mimic of human traits: the text appears bit by bit, as if written by a person and expresses emotions. This behavior of the LLM fosters anthropomorphization on the part of users, which has been shown to increase their confidence in the system [46]. Kran et al. [92] in fact categorize anthropomorphization as a dark pattern in LLMs.

The system purposely setting up wrong expectations is a dark pattern. Anthropomorphization is a possibility, but other kinds of dark patterns could fit this description. "Sneaking" for example, which is hiding, disguising or delaying information relevant to the user [68], can be used to create expectations on the part of users that will not be met.

If an error is defined as an incident that does not meet user expectations, then the setting of wrong expectations can itself be considered as a cause for errors.

Some of these errors can be categorized as dark patterns: these expectations susceptibly push users to act in a manner that is preferable to an organization's interests, and not to their own. However, other expectation errors are not malicious or intentional. For example, a participant in the workshops (see section 2.2) complained about a feature that they were expecting, but that was inexistant in the system. In this case, the assignment of responsibility is not so obvious. Were there design cues suggesting a functionality that did not exist? Is there an obvious feature missing in the tool? Or did the user misinterpret what the tool was for? In some cases, the expectation can come from similar softwares. For example, if Photoshop can afford to have a high-end feature, it will be expected everywhere, even in softwares like Gimp wich would not have the manpower to implement it. In any case however, this stray from expectation is an error, as defined here. The attribution of fault is not that important here in order to fix this error, it can very well be no one's fault. The investigation should instead start here: there is a problem in the representation the user makes about the system.

4.2 Methodological considerations around errors

The evaluation methods of error and its definition are co-defined: the shared idea of what an error encompasses is influenced by the methods used to measure it, and vice-versa. Consequently, the classical definition of error came with conceptions on how an event is correctly assessed. What does the approach of errors in this thesis implies in terms of methodologies? This section presents insights on potential changes of practices in order to better include user experience in the evaluation of problematic incidents.

4.2.1 Building on classical questionnaires

In the first chapter (subsection 1.3.2), I presented SUS and NASA TLX as the two most popular standardized questionnaires in HCI research. While they have their qualities in evaluating parts of user experience, this section points to the areas that they do not cover in the experience of negative incidents, as per the results of chapter 2 studies.

SUS and NASA TLX respectively evaluate the perceived usability of a system and the mental workload felt by experiments' participants. In the context in which they were developed, those focuses were indeed key. The SUS was created in an overall early 1980's movement to develop assessment methods to measure perceived usability, completing the already existing tools focusing on objective usability [108]. As such, this questionnaire was filling an existing gap in usability assessments specifically, not in the evaluation of system interfaces as a whole. NASA TLX on the other hand, was developed by the US space agency over three years, and assesses mental workload. The questionnaire was developed specifically to evaluate pilots' experiences in automated aircrafts [10].

Because of their success, however, both those methods have become go-to evaluations for HCI research. Their popularity sets a precedent, an insurance of their validity and accuracy. They are considered the most trusted questionnaires available for researchers to assess user experience. They have then become sort of the default evaluation, from the system (SUS) and the user (NASA TLX) point of view, when they were not designed with such ambition.

In this thesis, error is defined and studied as a negative user experience. The studies in chapter 2 interrogate users and extract the elements that define an error, as a stray from expectations. The criteria by which users assess a situation to be bad can arguably be elements to look for to evaluate user experiences.

Results of the studies in chapter 2 coincide with certain SUS and NASA TLX questions (see tables 1.13 and 1.14).

By asking to rate *"I found the system unnecessarily complex"* and *"I thought the system was easy to use"*, the SUS polls users on the impact of problems on the interface, and specifically the struggles they encounter. These questions very much connect to the **Impact on the user** theme uncovered in the workshops, and specifically to the code **Difficulty** within. This questionnaire also asks the user to evaluate the integration of the system's functions and the consistency of the system, coinciding with our **System's state and behavior** theme. The SUS evaluating usability, its scope overlaps the study of bad user experience.

NASA TLX investigates workload, and as such also finds commonalities with our studies. By featuring *"Was the task easy or demanding"*, the questionnaire also polls on the experience of **Difficulty**. Another question bears the ideas of the **Stress/Apprehension** and **Irritation** codes altogether (*"How discouraged, stressed, irritated and annoyed [...] did you feel?"*) – the latter was also highlighted as an importance consequence for errors in our survey. All those codes fit in the **Impact on the user** theme, that overlaps with NASA TLX's scope as well. The theme **User's state and behavior** also transpires in the questionnaire, and especially the code **Lack of time**, in *"How much time pressure did you feel?"*.

However, the workshops and survey presented in this thesis also uncover aspects of user experience that do not appear in standardized questionnaires. In the workshops, feelings of lacking competence or information, awareness of risks, acting out of habit or testing things out in trial an error, the assessment of a "bad design", feelings of surprise or the observation of a focus loss were all aspects that could discriminate types of error from one another. In the survey, the second most common consequence for errors was a loss of confidence in the tool. This consequence however might only transpire if there was time to build a confidence to begin with, *i.e.* with interfaces that the user is rather familiar with. Both SUS and NASA TLX are not vested with this type of consequences for a reason: they evaluate systems that users are discovering in their common use in HCI studies.

Particularly, blame attribution stood out as an important characteristic of error in both our studies. The questionnaire revealed that blame was generally unidirectional (towards

the system or the user) and could differentiate categories from one another. When “error” is not defined in relation to its attributed cause, this attribution can become a factor in evaluating a problematic phenomenon, as it has been in our survey.

SUS and NASA TLX do not evaluate user experience as a whole –and do not claim to do so. While such a method does not yet exist, it is possible to enrich them with other questions, taking into account the specificity of the context, adaptations that are rather common [10, 108]. An alternative would be to evaluate not user experience as a whole, but negative experiences in particular. The focus point is still the user perception, but targeting what went wrong, how, and the user interpretations. Future investigations meant to investigate those user experiences could augment the classical methodologies with evaluations of such aspects.

4.2.2 Quantitative and qualitative

In the field of HCI, there exists a conflict between quantitative and qualitative methods. More specifically, qualitative research is often subjected to criticisms from a quantitative point of view [40]; they can be criticized on the grounds of not having enough participants, enough data, not applying Inter-Rater Reliability (IRR) etc. In other words, they can be expected to be generalizable based on numbers. According to Crabtree [40], this kind of criticism that happens sometimes in the scientific reviewing process is rooted in *positivism*³¹. This school of thought posits the existence of a single objectivity, accessible through the measurement of the real world.

Numerous researchers make however the argument that any scientific study is always situated, *i.e.*, comes from a positional perspective [72]. Haraway [72] argues that a “god-like” point of view is impossible; that an objective truth cannot be obtained from a single approach. In HCI, Suchman [174] defends a similar perspective: a “*shift from a view of objective knowledge as a single, asituated, master perspective that bases its claims to objectivity in the closure of controversy, to multiple, located, partial perspectives that find their objective character through ongoing processes of debate*”.

Research questions and scientific methods are not self-evident, they are chosen, often-time unconsciously, according to people’s context as scientists and their point of view as individuals. Error in particular, as a scientific subject, is permeated by a history of situated perspectives. Critical systems, industry, engineering, and HCI have different priorities and expertises when it comes to error, while also influencing each other. Looking at chapter 1’s overview of these approaches, one can see that error is not always represented by the same metrics, nor measured by the same methods. Each of these approaches does not hold an absolute truth of what error is, but rather a situated point of view, which produces actionable information locally, within that situated point of view.

Despite these ongoing debates, qualitative and quantitative methods can also co-exist and complete each other in the form of mixed methods [4]. In fact, in chapter 2, the thematic analysis of workshops is used to inform a quantitative investigation. Qualitative methods first provide new insights on what different errors entail for users, in the expressiveness of their description of the incidents. An online survey then polls a larger number of participants, its questions arising from the results of the workshops. Mixed methods do not necessarily imply one informing the other, but such an approach was relevant in this case to first broaden the scope of possibilities in the traditional measures for determining the severity of an error.

The results from each of these methods also highlighted the situational nature of the feedback collected. As the workshop was a group exercise, it probably encouraged participants to respond to each other and encourage one another in their narratives. The questionnaire, on the other hand, was an individual exercise in which people reported memories of errors individually. These different contexts may explain some of the differences in results between the two studies: less blame for the system in the survey, and certain emotions expressed more intensely in the workshop.

³¹A philosophy that we already encountered in subsection 1.3.1, debated by Widder and Goues [184]

While using mixed methods arguably strengthen our results, other perspectives and other methods have been and will be necessary to “*find [an] objective character through ongoing processes of debate*” [174].

Running the study of interaction interferences, I also noticed that some quantitative tools could possibly strengthen qualitative feedback. For this work, we elected to use quantitative methods to evaluate the efficiency of our BEHAVIORS. We however left room for qualitative feedback to complete the questionnaires administered regularly throughout the controlled experiment. This last free comment by participants was collected by a small interview at the very end of the experiment.

In questioning participants at the time, I noted that this phase seemed enriched by the participants having to rank the BEHAVIORS immediately before. By asking them to compare the different designs, they had to consider their value, and reason in their choice. These reflections were very useful for the interview phase: most participants clearly presented what they liked about each BEHAVIOR, what they disliked, and the importance they gave to each of these aspects, ultimately justifying their ranking.

Working on this thesis, I observed that the weaving of quantitative and qualitative methods can be very rich and deepen the scope of scientific studies in interesting ways. If I had to start it all over, knowing what I know now, I would however change the order in which I conducted the studies. The controlled experiment on interaction interferences was indeed conducted before the workshops and the questionnaire described in chapter 2. It could have been interesting to use the results from the latter to more precisely question participants on their perception of the phenomenon, and maybe for instance seek to understand the **Criticality** they attributed to those events, both in the experiment and in general.

4.3 The stakes of interaction put forth by errors

Regardless of the approach taken, error disrupts the “normal” functioning of things. In doing so, it has the potential to challenge this normality. Looking at the user experience, this perspective reveals aspects of our relationship with our interfaces, the intentions they embody, and what we expect from them.

4.3.1 The benefits of a distance with representation models

As we explained in section 4.1.2, an error provides perspective and distance. It creates a gap between the expected behavior and the observed behavior, enabling a reassessment of what is considered *desirable*. Here, I present arguments defending that the perspective provided by error can also reveal the tool itself, as a mediator of an interaction.

For Vitali-Rosati [181], the “*naturalization*”³² of a world view, an opinion or an interpretation, is this view becoming so omnipresent that the fact that it is only one view among other possibilities, is generally forgotten. This interpretation becomes “natural”. As a consequence, it is not only perceived as neutral, but alternatives appear aberrant. Furthermore, the media or tool endorsing that view is not perceived, it is transparent, and the relationships it mediates appear as immediate, not mediated³³.

The concept becomes interesting in the light of representation models [181]. Indeed, any software working with a concept relating to physical reality will make assumptions, approximations in order to represent that concept in a way that is manageable. For example, an application like Tinder creates a model of attraction and appeal. Google Search implements an idea of relevance. This simplification must happen for the system to be possible at all, because there is no absolute truth of what these concepts entail every time. However,

³²Simply translated from “*naturalisation*” in French

³³“Immediate” comes in fact from the same latin roots as “mediated” in Latin, “im” being “without” and “mediate” “in-between” [186]

this representational model is rarely explicit, often vague, presented as inevitable. There is in turn a naturalization of those models: relevance becomes its choices of implementation in Google Search.

It is also generally considered that a good interaction technique must be invisible and feel “natural”. Typically, the development of pointing transfer functions probably improved the lives of mouse users, while not necessarily being noticed. Pointing became more and more efficient and precise, but people tend to recognize good changes not as intensely as bad ones, because of the negativity bias. This development “under the radar” is widely considered a good thing in design. The user does not feel like they have to learn an interaction technique, or that they need to accommodate to it changing.

Those representations exist also in the choices made throughout the development of technologies. Sometimes, a technical apparatus persists even as the technical level develops over time. Lanier [99] calls this a “*lock-in*”. This apparatus most often responds to a need at the time of its creation and becomes a basis on which other technical solutions are created. One example is the use of files, now universally widespread in computing, whereas other possibilities existed before. Today, however, it seems difficult to imagine a world without files. The use of files is naturalized throughout computing devices, and has rendered alternatives alien.

For Vitali-Rosati [181], an “intuitive” relationship with things implies a non-mediated relationship, an immediacy. The user does not see the mediation. And because it is the simpler, most direct rapport, it appears as the only type of relationship possible, or at least desirable. For the author, this lack of transparency is problematic and the reason for his “praise of bugs”.

In particular, Vitali-Rosati [181] cites Heidegger’s philosophy on technology. If we simplify his theory, Heidegger states that the very nature of a tool is to be available for use, while making the vision of the world it implements invisible. When one writes with a pen, they cannot be thinking about the pen itself and how they are holding it. In order to be perceived, it has to become unavailable, it has to break. Only then can it be observed so that other ways of doing things can be considered.

A system must break in order to be perceived for itself. However, different types of mediations are naturalized in a device, that do not have the same implications.

First, some naturalizations happen between a tool and the idea it implements. Tinder or Google Search construct a representation model between the user and a concept, existing outside the scope of the application. The choices made to implement appeal or relevance could be made visible without breaking the application altogether. One could imagine such applications showing the criteria measured to a certain degree, or at least making public the algorithm calculating those criteria.

On the other hand, interaction techniques are also naturalized. They are made invisible to the user. Transfer functions do not however appear as representation models. They are arguably a mediation, but through which the user controls the interface itself. The mediation is local. As per Heidegger’s characterization of a tool, it has to be invisible in order for the device to be usable at all. To be perceived, a transfer function has to break.

An error can have value in both those cases, making visible something that was hidden from the user. This does not mean however that they are on the same level. While the invisibility of interaction techniques is necessary for the user to manipulate the system, the obscurity of the representation models implementing a vision of the world is not.

A *lock-in* can happen both with mediating concepts and mediating tool interactions. The implementation of a concept can become common enough that it becomes the concept itself. For instance, Google Search’s implementation of relevance determines what will be seen or not, and in turn, somewhat, what is considered “relevant” or not. From a design point of view, an interaction technique can be so widely used that alternatives become unthinkable, or be dismissed from the strangeness alone. Errors are interesting there, in a research perspective, in that they can open new possibilities, making visible the mediation present in the interaction. Errors hold the potentiality to prevent a *lock-in*, or at least reveal its existence.

4.3.2 Errors and art

Whether we define it as a non-compliance to system specifications or a breaking of user expectations, an error is a disruption from perceived normalcy. As exposed in the previous subsection, it creates distance and perspective. This change of perspective can be an inspiration for artists, using error for their creations and performances.

Simon Weckert is an artist working on and around technology. As presented in [181], he notably participated in the colloque “Interactions multimédiales par écran” in Lyon on July 7th, 2022 with an installation. Using antennas dispersed in the room, he sent a GPS signal to all the phones present, geolocating all the people present’s smartphones in China.

His most famous performance was the Google Map Hacks, where he disrupted the functioning of the application [182]. Placing 99 smartphones running Google Maps in a handcart, he walked them through the streets of Berlin. The amount of devices using the application created a virtual traffic jam, which was in turn transmitted to other, real drivers. While he walked, cars in Berlin were redirected away from him, on other alternative routes, to avoid being stuck in traffic. The pictures and videos taken from this performance show him walking on the streets with nairy a car.



FIGURE 4.3: The Google Maps Hacks by Simon Weckert (pictures from [182])

Let us underline that in both those performances, the smartphones are working as intended. A smartphone is supposed to use a GPS signal to geolocate itself, and Google Maps is supposed to take in the positions of its users to determine the best routes. The problems are not in the functioning of those tools. From a developer’s point of view, they are working as intended.

Those artistic works are however considered *hacks*, because they challenge the idea of what the system is supposed to do. One expects their smartphone to geolocate itself depending on their position. They expect an application like Google Maps to choose a route depending on the number of cars on the road. Here, the artist creates a distance between what the application does and how it does it. When everything works as intended, the what and the how are not visibly dissociable: the smartphone will locate itself on the user’s position, according to an accurate GPS signal, and Google Maps will choose a route according to the number of cars on the road, corresponding to the number of phones running the application. By hacking the source of information that those systems run on, Weckert makes 2 facts visible to users:

1. These systems do not function on information directly taken from the real world: it interprets signals and makes assumptions doing so. The hacks presented here disrupt the system by messing with these assumptions.

2. If those systems are based on signals and not “the objective reality”, those signals have the potential to be hacked or falsified.

Those conclusions might not seem groundbreaking. Users can easily realize that their systems do not work magically, that they must take information and data from somewhere. However, when they are put forth as bluntly as here, those realities can make people feel uncomfortable. Vitali-Rosati [181] reports in his book that Weckert received messages and threats from unhappy users after the Google Map Hacks. This may stem from the amount of trust and dependency that we put on those systems. Google Maps is not a route mapping system among others, it is *the* reference route map. In fact, the rerouting feature that Weckert exploits in his experiment only works if a lot of –if not most, people are using the application. And most people do: the platform went over 2 billion monthly active users in 2024 [163]. Weckert was walking on empty streets, which also shows that a vast majority of drivers followed Google Maps indications in that moment.

In the context of relative monopolies like Google Maps, creating a distance between the function and the functionality also opens perspectives. Nowadays, choosing a route can be a synonymous to using Google Maps. However, if one realizes with Weckert’s hack that the application works a certain way, they face the possibility that it could work in another. Other artists play with this idea, not necessarily creating bugs to do so. Monjour, for instance, dissociates the action of going through a text from the scroll functionality. They install a crank handle to navigate a text [181]. If the system is less efficient than the existing one, it however raises the possibility that the scroll, universal in today’s devices, could work differently.

Because errors make visible the invisible, they can be a strong source of inspiration for artists. They can even be art itself when made with intention: a disruption that make people question their behavior and lived reality.

4.3.3 Distancing a Dark Pattern

A dark pattern is an interface feature that nudges the user into acting a certain way, in the interest of the organization behind the design. Examples include, among other things, bait-and-switch (advertising a free or low-cost item that is in fact non-available, only to present other options), confirmshaming (wording a confirmation step in a blaming way, for instance when not subscribing to a newsletter) or misdirection (misdirecting the user through counter-intuitive design).

Going back to the schema theory, a dark pattern can be seen as activating the wrong schema on purpose. With that, Norman’s definition of slips [128], would describe a dark pattern as a design feature triggering a slip on purpose.

Dark patterns also include nudges that can be more subtle. Monge Roffarello and De Russis [122] investigate for instance attention-capture dark patterns, which are meant to keep the user on the platform they are using. They point out 5 categories of design tools encouraging that behavior:

- recommendations can be used to send the user from one content to the next endlessly
- autoplay, *i.e.* when videos are played without any input from the user, taking away a decision-making step
- pull-to-refresh, especially on social media is manually reloading the interface with a pull gesture. This gesture offers a variable reward (revealing new content or not), playing with the same cognitive weaknesses that foster gambling addictions
- infinite scrolling, that can also be associated with a variable reward system
- social investment, that uses social metrics such as likes as reinforcement processes.

Those dark patterns steer the user towards a mindless, aimless state, in order for them to keep seeing and seeking more content.

As argued before, an error or a bug has the capacity to create a distance between the user and the tool. This distance can potentially break attention-capture dark patterns. For example, if my internet connection is slow, I might stop scrolling aimlessly on a social platform. As the interface breaks or slows down, one is made more aware of what they were doing and why.

Dark patterns, if they are considered errors in our categorization, rely on the interface working in the primary sense. Designers of dark patterns do not want an error or a bug to happen, because it also breaks their patterns.

In the same way, the distance created by error can inspire immediate perspectives on the activity itself. An error can momentarily put space between a user and their task, allowing its questioning. Why was I doing that? Is it the most efficient way? What was the goal?

4.3.4 The discretion of interaction interferences

This chapter points out the revealing properties of error in different ways. An error can create perspective, and from there allows a reflexion on the application, on the user behavior or on the task. This step back is however not systematic.

Interaction interferences are relatively common phenomena [162], that are rarely flagged as problematic by users. If people disagreed on them being simple cases of bad luck in our survey, the way people expressed themselves in the workshops suggests that they nonetheless can be attributed to a mishap. They are not in fact restricted to digital devices, and can happen in any context. Two conflicting actions just have to happen in a timeframe that prevents or limits the possibility of stopping one's course. They reveal in some way that user actions such as clicks or taps, that may appear instantaneous, do take time and can be disrupted as a result.

They tend to happen more in settings with a lot of events, changes or actions. Someone in the workshops brought up the example of FPS games for the **INTERACTION INTERFERENCE** category. It is indeed an event-filled environment where the possibility of shooting an ally by accident, through an interaction interference, is very real. In this case however, we could argue that the phenomenon is part of the game mechanic, participating in the feeling of immersion.

Describing the phenomenon to other people during my PhD, they would often react with a surprised acknowledgement that it had happened to them. Not that the phenomenon went unnoticed, they simply never regarded it as an error, *i.e.* as something that could potentially be fixed. Only when it was pointed out were users made aware of that possibility.

In a nutshell, interaction interferences are perceived as “normal” occurrences. They provoke annoyance and frustration but, to the user, are not seen as problems to be solved.

Interaction interferences and their consequences are not characterized as errors because at the moment, there exists no alternatives other than them happening. Since the phenomenon is not restricted to digital interfaces, this makes a lot of sense. In such a context however, the system's design could take into account their existence, by preventing their occurring or mitigate their consequences. The study in chapter 3 shows that there is potential in developing such design behaviors.

If new designs mitigating interaction interferences were to become more common, we can then imagine that more users would qualify them as errors. For now, the very possibility that their management could be different is lacking. When an error is “the way it's always been”, it will not be necessarily recognized as fixable or even problematic by the user.

The existence of different designs in general creates opportunities for errors in that way, because it creates different expectations. They can feel frustrating to users, for instance with an application not respecting design standards. At the same time however, these errors can foster productive discussions about those expectations and those standards. The multiplicity of interface designs can shed a light on their potentialities, through the occurrences of errors as defined in this thesis, possibilities that would otherwise be invisible in a uniform digital environment.

Conclusion

This thesis proposes and investigates a new take on the notion of error in HCI, defining it in relations to the user's expectations. An error becomes an experience with an interface that surprises or displeases the person, potentially disturbing their task but also possibly their relationship with the tool.

The classical approach of error relies on a tradition of *finding the causes*. The subject was indeed most studied in critical environments, by experts in safety and engineering, where a problem holds the potential of prompting catastrophic consequences. These fields characterized error within a certain context, highlighting the importance of determining a cause, pinpointing an area that had to be corrected in order to avoid a recurrence. However, this approach carries the risk of overly focusing on human error, specifically that of the operator at the end of the line, which is (conveniently) easier to trace. This argument fueled debates within the scientific community, leading to the questioning of organizations and systems that produce errors.

Inheriting from these approaches, HCI research often characterizes error from an engineering point of view, categorizing their causes. Other angles exist within the field, but usually focus on a specific error phenomenon or context.

Reframing the subject as *an event, caused by or during an interaction with a digital device, that does not meet the user's intents, expectations and wants about that interaction*, the concept of error developed in this thesis encompasses a broader set of problematic phenomena and introduces new characteristics by which an event can be assessed. This work shows that errors can be explained and described without putting the emphasis on finding who or what to blame. Through this shift of perspective, the gravity of an event becomes that experienced by the user, be it critical for the system or not. The user's perception being central to this definition does not mean that they are always right, or that they might "decide" how an error is evaluated. The analysis of an error simply begins with the observation of a discrepancy between the user's expectations and the behavior of the system.

I propose a categorization of errors according to the phenomena perceived by users. This taxonomy was confronted to users through two different consultations, mixing qualitative and quantitative approaches. Workshops showed that conversations about errors can induce expressions of emotions like guilt, frustration or surprise, but also deeper commentaries on blame regarding the event, self-confidence or perception of the tool. These results helped constructing an online survey on the perception of error, confirming the coherence of the categorization and providing a user experience portrait of each type of error. These studies display that an error, as defined in this thesis, is both situated and subjective.

Among the types of incidents in the taxonomy, we studied in particular interaction interferences as a recently-described, atypical combination of consequences: they are annoying to the user, but would not have been considered heavily problematic by the former approaches of error. While they rarely cause material consequences, they are source of frustration. We studied solutions to the phenomenon, not by correcting the behavior of the system or the person, but by changing the terms of the interaction. Our design behaviors for pop-up style GUI elements prevent the input from being taken by a new element showing up. The preferred solutions also respect users' range of expectations and felt predictable to them.

This new definition of error brings about methodological considerations for research. The evaluation of error becomes the evaluation of the violation of user's expectations. Those

do not only stem from the user, but also from the context in which the error happened, and the potential setting of wrong expectations, intentional or not. Errors also provide users and researchers with a distance to the device and its design, allowing to consider how it could be different.

As a final note, this new theory of error does not claim to replace the previous ones, but simply provides a new perspective to complete and refine the assessments of problems arising from our interfaces. It does, however, have the advantage of putting the user at its center, the very person experiencing the consequences of these phenomena. To come back to the French expression “l’erreur est humaine”/“the error is human”, I could now interpret it differently than as a comment on human fallibility. Instead, errors are human because they are characterized as errors by users and designers, and because they elicit profoundly human reactions.



FIGURE 4.4: Couple's therapy drawn by *Patte de Lou* [133]

Glossary

dark pattern instances where designers use their knowledge of human behavior (e.g., psychology) and the desires of end users to implement deceptive functionality that is not in the user's best interest [68]. 64, 89, 95, 96, 101, 110, 149, 183, 189

error (as developed in this thesis) an event, caused by or during an interaction with a digital device, that does not meet the user's intents, expectations and wants about that interaction. 14, 57, 59, 64, 68, 71, 75, 78, 79, 103, 111, 115, 116, 117, 119, 121, 122, 123, 127, 143, 144, 145, 147, 174, 176, 177, 179, 180, 181, 183, 184, 186, 190, 191

interaction interference changes in the user interface occurring immediately before the user carries out a meaningful atomic input action, but too late for them to inhibit that action—even if they perceived the change—which as a consequence is interpreted differently than the user intended [110]. 14, 64, 71, 145, 147, 179, 186, 190, 191

interruption Cognitive disturbance provoked by a change in the interface, causing the user to loose focus. 70, 89, 179

mode error an error resulting from the user acting correctly, according to a certain state of the system, but that is not in fact the active state of the system.. 47, 54, 62, 67, 92

procedural error the violation of the (optimal) path to the current goal by a non-optimal action. This can either be the addition of an unnecessary or even hindering action, which is called an intrusion. Or a necessary step can be left out, constituting an omission [70]. 48, 58

S. PERERE Simulation of Performance in Error [17]. 62

schema a mental codification of experience that includes a particular organized way of perceiving cognitively and responding to a complex situation or set of stimuli [120]. 32, 34, 35, 41, 47, 48, 189

slip routine, skill-based actions deviate from the intended goal due to lapses in attention, habit interference, or environmental design [129]. 62, 64, 68, 79, 148, 189

Acronyms

AI Artificial Intelligence. 66, 77, 86, 183

AR Augmented Reality. 55, 56

ATC Air-Traffic Control. 31, 197

ATS Activation Trigger Schema [128]. 9, 32

BCa Bias-Corrected and accelerated [183]. 164

CI Confidence Interval. 164

COCOM Contextual Control Model [76]. 9, 38, 41

ECM Evaluated Classification Mismatch [21]. 52, 53, 197

EOC Error Of Commission. 31

FRAM Functional Resonance Analysis Method [77]. 9, 38, 41, 42, 45, 197

GEMS Generic Error Modeling System [149]. 9, 29, 32, 33, 34, 38, 51, 62, 197

GOMS Goals, Operators, Methods and Selection rules [31]. 49

GUI Graphical User Interface. 147, 148, 150, 151, 152, 154, 155, 156, 157, 159, 169, 173, 174, 175, 176, 177, 191, 198

HAMSTERS Human-centered Assessment and Modeling to Support Task Engineering for Resilient Systems [55]. 49, 50, 51, 197, 201

HCI Human-Computer Interaction. 3, 7, 13, 14, 17, 19, 20, 21, 25, 27, 31, 39, 41, 47, 49, 51, 52, 53, 54, 55, 58, 59, 60, 61, 63, 64, 76, 77, 78, 89, 93, 95, 96, 110, 148, 179, 184, 185, 191, 209

HCIL Human-Computer Interaction Loop. 52, 53, 55, 197

HEI Human Error Identification. 29, 30, 31, 33, 51

HEIST Human Error Identification in Systems Tools. 33

HMI Human-Machine Interaction. 41

HRA Human Reliability Assessment. 30, 32, 33

HSD Honestly Significant Difference [111]. 164

IRR Inter-Rater Reliability. 100, 185

JCS Joint Cognitive System [82]. 40, 41

LLM Large Language Model. 58, 66, 88, 92, 183

MFG Memory For Goals [5]. 48, 49, 58

MIODMIT Multiple Input and Output Devices and Multiple Interaction Techniques [41]. 53, 197

- NASA TLX** National Aeronautics and Space Association Task Load index. 60, 61, 184, 185, 201
- NAT** Normal Accident Theory [137]. 45
- NATO** North Atlantic Treaty Organization. 26
- OAM** Organizational Accident Model. 37
- OS** Operating System. 93
- OSH** Occupational Safety and Health. 43
- OSS** Open Source Software. 56
- PCM** Perceptual Cycle Model [127]. 9, 32, 34, 35, 36, 197
- POISE** People, Organization, Interactive System, Environment [131]. 53
- PPI** Positive Performance Indicator. 43
- PSA** Probabilistic Safety Assessment [127]. 30, 31
- RE** Resilience Engineering. 29
- RM ANOVA** Repeated-Measure Analysis Of Variance. 164, 165
- SCM** Swiss Cheese Model. 36, 197
- SPAR-H** Standardized Plant Analysis Risk-Human Reliability Analysis [61]. 32
- SRK** Skill, Rule and Knowledge [144]. 9, 29, 32, 33, 38, 45
- STAMP** Systems-Theoretic Accident Model and Processes [106]. 45
- SUS** System Usability Scale [26]. 59, 60, 61, 184, 185, 201
- SUX** Shitty User Experience. 67
- TAG** Task-Action Grammar [135]. 49
- VR** Virtual Reality. 62
- VSS** Vulnerable System Syndrome [154]. 37

List of Figures

1	Word search: 30 words from the Acknowledgement section, all directions allowed	8
2	Expectation clash drawn by <i>Patte de Lou</i> [133]	15
1.1	Venn diagram of the English synonyms codes	20
1.2	The chain of dependability according to [9]	22
1.3	Fault classes as presented by Avizienis et al. [9]	23
1.4	Fault classes tree as presented by Avizienis et al. [9]	24
1.5	Failure modes according to the four viewpoints as presented by Avizienis et al. [9]	24
1.6	Taxonomy of unsafe acts in Reason [149]	27
1.7	The user operability evaluation simplified schematic in Payne and Altman [134]	28
1.8	<i>“Control flow model of the ATC simulation operator”</i>	31
1.9	Influence of Performance Shaping Factors on human error probability [61]	32
1.10	Simplified diagram of the SRK model from Rasmussen [144]	33
1.11	GEMS as presented by Embrey [51]	34
1.12	Categories of errors from the GEMS presented by Hollnagel [80]	34
1.13	Neisser’s Perceptual Model Cycle as presented in [138]	35
1.14	PCM adapted and applied to the Tesla crash incident by Banks, Plant, and Stanton [16]	35
1.15	SCM to explain the cause of accidents from Reason, Carthey, and Leval [154]	36
1.16	Blame cycle from Reason, Carthey, and Leval [154]	38
1.17	Phenotype and genotype of errors in Hollnagel [80]	40
1.18	The cyclical model of human action by Hollnagel, applied to driving [53]	41
1.19	Example of a FRAM-built model from a batch process [60]	42
1.20	Tension between the <i>Old View</i> and the <i>New View</i>	44
1.21	Simplified flow chart of Halbrügge, Quade, and Engelbrecht [70]’s cognitive model. Dashed arrows represent a retrieval error, and the dotted ones represent a failure in visual search.	49
1.22	HAMSTERS representation for objects, information and knowledge	50
1.23	Checking the weather conditions HAMSTERS task representation	51
1.24	ECM classification process in [21]	53
1.25	MIODMIT simplified by Palanque, Cockburn, and Gutwin [132]	53
1.26	Palanque, Cockburn, and Gutwin [132]’s changes to Avizienis et al. [9]’s taxonomy	53
1.27	Müller, Oulasvirta, and Murray-Smith [123]’s control theoretic representation of the HCIL	55
1.28	Bahaei et al. [14]’s taxonomy of faults. The dotted faults are AR-related.	56
1.29	Computer frustration model from [75]	66
1.30	In the game <i>Stray</i> [173], the controls for directions are inverted when the character has a paper bag on their head (example and screenshot taken from [39])	67
1.31	Interaction interferences characteristics reported by survey participants in [162]	72
2.1	The steps of the CATEGORIES PHASE of the workshop. On the yellow canvas are written the tasks given to the participants at each step.	83
2.2	Percentages of tags within each workshop’s examples	102
2.3	Percentages of examples within each workshop tagged with each code	109

2.4	Schema of categories inter-relationships. Dotted arrows represent a partial overlap of categories ("Some INTERRUPTIONS are DARK PATTERNS₊ "), and full arrows an inclusion ("all INTERACTION INTERFERENCES are WRONG SYSTEM INTERPRETATION ").	112
2.5	Questionnaire process diagram	120
2.6	Participants' age groups (left) and last diploma (right).	127
2.7	Number of participants for each level of (left) confidence in using their usual tools, (middle) confidence in handling new tools, and (right) overall expertise in handling digital technology.	128
2.8	Participant count for every number of answered examples	128
2.9	Number of incidents for every consequence (participants could tick more than one per incident)	130
2.10	Number of responses for every response of "Do you consider that this incident was the fault of someone or something?"	130
2.11	Participants assessments of whether they thought they deserved the incident to happen (left), and whether they felt guilty (right).	131
2.12	Number of situations for every level of habit (left) and comfort (right) in the context	131
2.13	Types of tasks during which the reported incidents occurred.	132
2.14	Selected answers for PROBLEMS WITH THE EQUIPMENT	132
2.15	Selected answers for ERRORS IN PLANNING	133
2.16	Selected answers for WRONG USER INTERPRETATION	134
2.17	Selected answers for WRONG SYSTEM INTERPRETATION	135
2.18	Selected answers for INTERRUPTIONS	136
2.19	Selected answers for INTERACTION INTERFERENCE	137
2.20	Selected answers for AIMED ACTION	138
2.21	Selected answers for SLIP	140
2.22	Selected answers for FAIL-UNSAFE	141
2.23	Selected answers for OMISSION	142
2.24	Selected answers for DARK PATTERN₊	143
3.1	(a) Simplified diagram of human reaction to a stimulus. Interferences can start before movement onset. (b) The "horse race" model of movement inhibition. The user successfully suppresses his action if the stop-process ends before the main action process. Reproduced from [162] with the authors' permission. . .	148
3.2	Time windows for the three main categories of behavior (<i>Prevent, Avoid, Correct</i>) and their corresponding technical requirements (in italics).	150
3.3	The AWAY_{DISPLACE} BEHAVIOR makes the window appear at the same time as initially planned, but some distance away from the cursor's path. Clicking as planned selects the target.	154
3.4	AWAY displays the new GUI element so that its center is on the line perpendicular to the latest cursor displacement, and the distance between its closest edge and the last cursor location P_n is the same as between the last and 10 th last cursor location P_{n-9}	155
3.5	The DELAY BEHAVIOR postpones the appearance of the window until 250 ms have passed, the cursor stops moving, or until a click. Clicking as planned selects the target and immediately triggers the appearance of the window. Moving the cursor around postpones that appearance even further.	155
3.6	With the HOLEOVER BEHAVIOR the window appears at the planned time and place, but a hole in the pop-up follows the cursor and sends input events through for 1 second. Within that second, clicking as planned selects the target. Clicking after more than 1 second affects the window normally.	156
3.7	The FREEZE BEHAVIOR makes the appearing window unresponsive for 1 second or until a click. Clicking as planned deactivates the "freeze", after which any action on the window is treated normally.	156
3.8	Experiment task (a) before and (b) after a popup is displayed (to scale, cropped)	158

3.9	A trial in our study. (1-2) The participant moves their cursor towards the current target in yellow. (2a) A pop-up window may appear if the trial is <i>Non-Interfering</i> , or (3-3a) 100 ms after the user enters the target in <i>Interfering</i> trials. (4-6) In any case the participant completes the primary task. The dropped target turns green, the new current target turns yellow, and a new target appears in cyan. (6a-6b) In <i>Pop-up</i> trials, the participant is asked what is/was written on the pop-up window. Sizes and distances were adjusted for figure readability; the cursor did not actually turn red on clicks.	160
3.10	Main phases and blocks of our experiment protocol.	160
3.11	Fitts Law and time thresholds for Participant 17: the Fitts law regression (blue line) is calculated from the durations of the calibration trials (blue dots). The time range within which the <i>Non-Interfering</i> windows spawn (green area) spans from the beginning of the trial (0 ms) to 100 ms before the Fitts regression, depending on the Fitts Index of Difficulty of the target. The threshold to trigger the 4-second warning message also depends on the Fitts law regression and on the Index of Difficulty.	161
3.12	Screenshot of our experimental interface during the familiarization phase. . .	162
3.13	Distribution of the time between the appearance of a window and the first click of the trial, for all BEHAVIORS. Vertical lines indicate the 350 ms mark. .	166
3.14	Proportions of <i>Correct</i> , <i>Incorrect</i> , and <i>Unknown</i> answers to the secondary task with the DEFAULT behavior, as a function of TRIALTYPE.	166
3.15	Percentages of closed pop-up windows as a function of BEHAVIOR and TRIALTYPE.	168
3.16	Percentages of correct answers to the secondary task as a function of BEHAVIOR and TRIALTYPE.	168
3.17	Total trial duration (in blue) and target acquisition duration (in orange) for each BEHAVIOR, in the <i>Normal</i> trials (top graph) and in the <i>Non-Interfering</i> trials (bottom).	170
3.18	Stacked histogram of participant rankings per BEHAVIOR.	171
4.1	User error drawn by <i>Patte de Lou</i> [133]	181
4.2	Responses to issue 6071 on the Request repository [85]	182
4.3	The Google Maps Hacks by Simon Weckert (pictures from [182])	188
4.4	Couple's therapy drawn by <i>Patte de Lou</i> [133]	192

List of Tables

1.1	Most common synonyms in English according to Thesaurus [178]	18
1.2	Most common synonyms in French according to Synonymo.fr [175]	18
1.3	Themes of error coming from its synonyms	18
1.4	Definitions of “error”	25
1.5	Definitions of “mistake”	26
1.6	Definitions of “slip”	26
1.7	Tenets in system safety and their simplified definition according to Grant et al. [66]	46
1.8	Classification of slips <i>i.e. “error[s] in carrying out the intention”</i> by Norman [130]	48
1.9	HAMSTERS task types	50
1.10	HAMSTERS operators	50
1.11	Typing errors from [168]	54
1.12	Classification of expectation violation in bug reports by Chilana, Ko, and Wobbrock [35]	57
1.13	The standard SUS questions. Scores go from 1 (“Strongly disagree”) to 5 (“Strongly agree”)	59
1.14	The NASA TLX, table copied from [86]	60
1.15	Properties to look for for an error-insensitive system	62
1.16	Categories of frustrating events in [75], their percentage among all incidents reported and their frustration level median. Frustration levels ranked from 1 (not very frustrating) to 9 (very frustrating).	65
2.1	The 10 categories of errors before the workshops and their origins	75
2.2	Categories taken from the literature about error, their description (in italic) and examples (in small text)	77
2.3	FAIL-UNSAFE category, its description (in italic) and examples (in small text)	78
2.4	“Borderline cases” categories, their description (in italic) and examples (in small text)	79
2.5	Instructions sent by email to the participants before the workshops (translated from French)	81
2.6	Instructions given to the participants at the beginning of each workshop (translated from French)	81
2.7	Order in which the workshop participants went through the CATEGORIES	82
2.8	Questions given to the workshop participants during the DISCUSSION PHASE (translated from French)	84
2.9	Questions for each participant after the workshop. The American equivalents of the French diplomas are given in italic	85
2.10	Data drawn from the <i>Incidents_W</i> given by participants during the workshops.	86
2.11	Number and nature of correct <i>Incidents_W</i> given by CATEGORY and by workshop. Percentages represent the proportion of examples in the category in question	95
2.12	After the workshops, 12 categories of errors. Parentheses indicate our own label before the workshops suggested name changes. Changes made to the definitions are in italics. Plus signs indicate categories added as a result of the workshops.	97
2.13	Codes and code labels from the workshops	99

2.14	Number of <i>Incidents_W</i> tagged with each code per CATEGORY. The shades of yellow highlight the biggest values in each lines, <i>i.e.</i> the categories for which each code has the most <i>Incidents_W</i>	104
2.15	Percentage of <i>Incidents_W</i> of each CATEGORY tagged with each code. The shades of yellow highlight the biggest values in each lines, <i>i.e.</i> the categories for which the percentage of <i>Incidents_W</i> tagged with each code is the highest.	105
2.16	Workshop codes grouped by themes	111
2.17	Questionnaire <i>Sub-categories_Q</i> (in bold) and <i>Examples_Q</i> (A, B, C for each <i>Sub-category_Q</i>). The <i>Examples_Q</i> taken from the workshops are marked with a W, others were filled out by the researchers.	119
2.18	Demographic questions at the beginning of the survey. The English equivalents to the French diplomas are in italics.	122
2.19	Words given to participants of the questionnaire to characterize each <i>Sub-category_Q</i> . The codes are described in table 1.3. It is of note that the words in English are not always equivalent to the words in French.	123
2.20	Questions about the type of incident. The questions marked with SPEC only appear when the participant selected a specific example to describe, the questions marked with GEN only appear when they elected to rate the type of incident in general.	125
2.21	Questions about a specific occurrence of the selected <i>Example_Q</i> . The questions marked with SPEC only appear when the participant selected a specific example to rate, the questions marked with GEN only appear when they elected to describe the type of incident in general (see table 2.20).	126
2.22	Interpretations of the ϵ^2 effect sizes, as prescribed in [145].	129
3.1	Possible combinations (in blue) of behavior types (rows) ordered by categories (1 st column), and interpretative requirements (last three columns) in our design space. Asterisks (*) indicate behavior types that are location-dependent. Cells marked with 'X' indicate behaviors evaluated in the following controlled study. In our study of pop-up interferences below, the condition implementing THROUGH (<i>Naive</i>) will also serve as a demonstrator for MAPPING (<i>Naive</i>) ($\uparrow$).	152
3.2	Types of interference-solving behaviors grouped by their technical requirements (green cells). Cells marked with 'X' indicate behaviors evaluated in the following controlled study.	153
3.3	Interference-avoiding behaviors as they are implemented and explained to the participants in the experiment. As opposed to the rest of this study, the behavior names presented to the participants did not include the category they belong to, <i>e.g.</i> AWAY instead of AWAY _{DISPLACE}	157
3.4	Questions asked after each BEHAVIOR. NA stands for "Non-Applicable".	163

Appendix A

Definitions of synonyms of error

A.1 English definitions

These definitions come from the Cambridge Dictionary website [30].

blunder: a (bad) mistake.

failure: 1. the state of not having achieved a goal 2. sb or sth that does not achieve a goal 3. the act of not doing what you are supposed to 4. a situation in which sth stops working or progressing properly 5. the state or act of failing 6. an unsuccessful person or thing 7. inability, refusal etc to do something

fault: 1. responsibility for sth bad. 2. at fault: responsible 3. a weakness in sb's character 4. a problem or negative quality 5. a mistake; something for which one is to blame 6. an imperfection; something wrong

flaw: 1. sth that is not perfect 2. a fault; something which makes something not perfect

glitch: a small problem

inaccuracy: when sth is inaccurate, or sth that is inaccurate

lapse: 1. when sth fails or is not done properly 2. a mistake or failure (in behaviour/behavior, memory etc)

miscalculation: 1. a calculation that has not been done correctly, or the act of calculating something wrongly 2. a wrong judgment about a situation 3. the act of calculating an amount wrongly 4. the act of judging a situation badly

miscue: 1. an occasion when a sports player makes a mistake when hitting, kicking, or catching the ball 2. a mistake

misdeed: an act that is criminal or bad

mismanagement: 1. the process of organizing or controlling something badly 2. a situation in which something such as a company or an economy is organized or controlled badly

misstep: 1. an act of placing your foot somewhere in an awkward way 2. a mistake

misunderstanding: 1. an occasion when someone does not understand something correctly 2. a disagreement, argument, or fight 3. a failure to understand, or an argument resulting from the failure of two people or two sides to understand each other

offense: 1. the condition of having your feelings hurt esp. because someone has been rude or showed a lack of respect 2. an illegal act; crime

omission: the act of not including something or someone that should have been included, or something or someone that has not been included that should have been

sin: 1. the offence of breaking, or the breaking of, a religious or moral law 2. anything considered wrong

transgression: 1. the act or process of breaking a law or moral rule, or an example of this 2. an action that breaks a law or rule

wrongdoing: a bad or an illegal action

X: written on an answer to a question to show that the answer is not correct

absurdity: 1. the quality of being stupid and unreasonable, or silly in a humorous way 2. something that is stupid or unreasonable

boner: a mistake that causes embarrassment to the person who makes it

boo-boo: 1. a mistake 2. a slight injury

delinquency: 1. behaviour, especially of a young person, that is illegal or not acceptable to most people 2. an action that is illegal or not acceptable

delusion: 1. belief in something that is not true 2. something a person believes and wants to be true, when it is actually not true

deviation: 1. the action of being or doing something that is different from the usual or common way of behaving 2. a difference from what is usual or expected 3. the difference between a particular number and the average or normal number

erratum: a mistake in a printed or written document

fall: 1. the fact of the size, amount, or strength of something getting lower 2. the act of falling down to the ground, usually without intending to or by accident 3. the fact of being defeated or losing your power 4. a reduction in the amount or level of something 5. a situation in which someone or something that has been successful fails

fallacy: 1. an idea that a lot of people think is true but is in fact false 2. a false belief

falsehood: 1. a lie 2. something that is not true

falsity: 1. the state of not being correct 2. the state of not being sincere or of not expressing real emotions

goof: 1. a silly mistake 2. a silly or stupid person

howler: a stupid and obvious mistake, especially in something that someone says or writes

misapprehension: a failure to understand something, or an understanding or belief about something that is not correct

misconception: an idea that is wrong because it has been based on a failure to understand a situation

miss: 1. an occasion when something or someone fails to hit something or avoids hitting something 2. a failure to hit, catch, do, see, or experience something

oversight: a mistake made because of a failure to notice something

screamer: 1. a device that makes a very loud, unpleasant sound 2. an internet video that pretends to be an image, game, test, etc. but is intended to frighten or shock you unexpectedly

slight: an action that insults a person because someone fails to pay attention to them or to treat them with the respect that they deserve

slip: 1. a small mistake 2. an act of falling or sliding 3. an occasion when something becomes worse 4. a mistake that someone makes when not being careful 5. a reduction in the level or amount of something

solecism: 1. behaviour that is a social mistake or is not polite 2. a grammatical mistake

stumble

trespass: 1. the crime of going onto someone's land or entering someone's building without permission 2. an offence in which you break a moral or religious law 3. the act of going onto land or entering a building without the owner's permission

untruth: 1. a statement that is not true 2. the fact that something is not true

bad job: do a bad job: to do something badly

faux pas: 1. words or behaviour that are a social mistake or not polite 2. a remark or action in a social situation that is a mistake and causes embarrassment or offense

misbelief

misjudgment: 1. an unfair or wrong opinion about someone or something 2. a wrong guess about an amount or distance

screw-up: a situation in which you do something badly or make a big mistake

slip-up: 1. a mistake or something that goes wrong 2. a mistake that someone makes by not giving something enough attention 3. a careless mistake

A.2 French definitions and their traduction

These definitions come from the Synonymo.fr website [175].

aberration : Accès de folie, égarement, grave erreur de jugement ; absurdité : C'est une aberration de laisser les enfants jouer près de la route.

écart : Action de se détourner, de s'éloigner de sa ligne de conduite : Faire quelques écarts de régime.

égarement : Action de s'écarter des voies de la raison ; folie passagère ; aberration, dérèglement : Un moment d'égarement.

ânerie : Parole ou acte stupide d'un ignorant ou d'un sot.

étourderie : 1. Caractère d'une personne étourdie, défaut de prévoyance, de réflexion, d'attention ; distraction, inattention : Agir par étourderie. 2. Action, faute commise par inattention ; imprudence : Commettre des étourderies.

aveuglement : Manque de discernement par passion, obstination : Un aveuglement coupable.

bêtise : 1) Manque d'intelligence ou de jugement ; stupidité : J'ai eu la bêtise de ne pas vérifier aussitôt. 2) Propos stupide, irréfléchi, ou futile et plaisant ; sottise : J'ai peut-être dit une bêtise ? 3) Action irréfléchie, imprudente ou répréhensible : Vous avez fait une bêtise en refusant cette proposition.

bévue : Méprise grossière due à l'ignorance, à l'étourderie.

bavure : 1. Tache, macule dans une impression, dans une épreuve d'imprimerie, dans une empreinte, etc. 2. Erreur regrettable, faute lourde, commise notamment à l'occasion d'une opération policière ou militaire, ou conséquence fâcheuse d'une action : Une bavure des forces de l'ordre.

blague : 1. Histoire plaisante imaginée pour amuser ou pour tromper : Dire des blagues. 2. Farce, plaisanterie faite aux dépens de quelqu'un : Faire une sale blague à un camarade. 3. Action irréfléchie, maladroite ; erreur : Cette blague dans son travail lui a coûté très cher.

boulette : Faire une boulette, commettre une bévue, une erreur.

bourde : 1. Sottise, erreur : Commettre une bourde. 2. Vieux. Histoire inventée pour abuser quelqu'un ; mensonge : Tu nous racontes des bourdes.

confusion : 1. Action de confondre, de prendre quelque chose ou quelqu'un, pour quelque chose, quelqu'un d'autre ; erreur, méprise : Ce malentendu était dû à une confusion de noms. 2. État de ce qui est confus, indistinct, désordonné ; désordre : Ce plan reste d'une grande confusion. 3. Situation d'un groupe où tout le monde s'agite de manière désordonnée : La réunion s'est terminée dans la confusion générale. 4. Embarras, trouble causé par le sentiment d'une maladresse, d'une faute ; gêne, honte : Rougir de confusion.

contresens : 1. Sens contraire à celui dans lequel on doit prendre quelque chose : Prendre le contresens d'une étoffe. 2. Interprétation opposée à la véritable signification ; sens erroné : Une traduction pleine de contresens. 3. Action ou attitude absurde, situation opposée à ce qui devrait être : Cette politique est un contresens.

coquille

défaut : 1. Absence, manque ou insuffisance de ce qui serait nécessaire : Défaut de mémoire, d'attention. 2. Imperfection matérielle de quelque chose : Diamant qui a un défaut. 3. Imperfection physique de quelqu'un, d'un animal : Il a un léger défaut de prononciation. 4. Imperfection morale ; travers : Le mensonge est un vilain défaut. 5. Manquement aux règles de l'art, du goût : Les défauts d'un film.

dérèglement : 1. État de ce qui est déréglé, irrégulier, désordonné, qui ne fonctionne plus régulièrement ; détraquement : Dérèglement du pouls. 2. Action de s'écarter des règles de la morale ; désordres dans la manière de se conduire : Le dérèglement des mœurs.

déviation : Écart hors d'une ligne morale, philosophique ou politique donnée ; variation dans la conduite : Légère déviation de principes.

distracted : 1. Disposition habituelle à l'étourderie ou état passager de quelqu'un qui manque de concentration : La distraction des savants est légendaire. 2. Acte qui traduit une sorte d'inattention : Excusez ma distraction : je ne vous ai pas prié de vous asseoir.

doublon

duperie : Littéraire. Action de duper, de flouer ; tromperie ; fait d'être trompé, mystifié : Cette promesse est une duperie.

entorse : Atteinte portée à quelque chose ; manquement : Entorse au règlement.

erratum : Faute d'impression répertoriée dans l'errata.

errements : Agissements habituels et erronés de caractère blâmable ; erreurs : Persister dans ses errements.

faiblesse : 1. Manque de solidité, de résistance : La faiblesse d'une voûte. 2. Manque de puissance, de ressources : La faiblesse d'une économie. 3. Manque d'énergie, d'intensité, de mordant : Faiblesse de la voix. 4. Manque d'étendue, de grandeur : Faiblesse d'une différence. 5. Manque d'efficacité ; défaut dû à cette insuffisance : La faiblesse d'un système. 6. Manque de vigueur, d'énergie ; incapacité à résister, à se défendre : Être d'une faiblesse coupable avec quelqu'un. 7. Littéraire. Impuissance à résister aux tentations, aux passions, et en particulier à l'amour ; manque d'emprise sur soi-même : La faiblesse humaine. Sa faiblesse pour elle le poussait à des imprudences. 8. Point faible : C'est là la grande faiblesse du film. 9. Manque de capacité intellectuelle, de talent, déficience dans les connaissances : Faiblesse d'esprit.

faute : 1. Manquement à la règle morale, à une prescription religieuse : Faute avouée est à moitié pardonnée. 2. Manquement à une règle, aux devoirs qui découlent d'un contrat de travail ou d'une activité réglementée : Une faute professionnelle grave. 3. Manquement à un règlement, à une règle de jeu : Faute de service au tennis. 4. Manière d'agir qui manifeste un manque d'habileté, de prudence : Mon emportement m'a fait commettre quelques fautes. 5. Manquement à une norme, à un principe, à une procédure : Faute de frappe. Faute de français. 6. Responsabilité de quelqu'un ou de quelque chose dans un acte coupable, une erreur, un manquement, ou dans une quelconque situation : L'accident est arrivé par sa faute.

faux : 1. Ce qui est contraire à la vérité ; erreur, mensonge : Plaider le faux pour savoir le vrai. 2. Altération frauduleuse de la vérité susceptible de causer un préjudice, par la fabrication ou l'usage d'une pièce, d'un objet, etc. : Le testament est un faux. 3. Copie frauduleuse d'une œuvre originale.

faux pas

folie : 1. Manque de jugement : Vous n'aurez pas la folie de démissionner maintenant. 2. Idée, parole, action déraisonnables, insensées : Des folies de jeunesse. Dire des folies.

fourvoiement : Littéraire. Action de fourvoyer quelqu'un ou fait de se fourvoyer ; erreur.

gaffe : Acte, parole particulièrement inadaptés à une situation ; maladresse, impair.

hérésie : 1. Idée, opinion, pratique qui s'oppose aux idées, aux opinions généralement admises : Une hérésie scientifique. 2. Au figuré. Manière d'agir jugée aberrante, contraire au bon sens et aux usages : Boire de l'orangeade avec un poulet rôti, c'est une hérésie !

illusion : 1. Interprétation erronée d'une donnée sensorielle : Illusion auditive. Illusion d'optique. 2. Effet obtenu par le moyen de l'art, de l'artifice, du truquage et qui crée le sentiment du réel ou du vrai : L'illusion de la vie donnée par un automate. 3. Appréciation conforme à ce que quelqu'un souhaite croire, mais fausse par rapport à la réalité : Perdre ses dernières illusions.

impair : Maladresse choquante ou préjudiciable ; manque de tact, de goût.

inadvertance : Faute commise par inattention, étourderie.

inexactitude : 1. Caractère de ce qui est inexact, erroné : L'inexactitude d'un renseignement, d'un calcul. 2. Faute, erreur commise : Récit plein d'inexactitudes.

infidélité : 1. Littéraire. Manque de fidélité, de respect à un engagement : L'infidélité à la parole donnée. 2. Littéraire. Déloyauté, trahison de la part de quelqu'un qui est subordonné à quelqu'un d'autre : L'infidélité des sujets à leur roi. 3. Manque de fidélité envers un ami, un être aimé : Cet ami m'a fait une infidélité. 4. Violation du devoir de fidélité entre époux, qui peut constituer une cause de divorce ou de séparation de corps. 5. Fait d'avoir recours à quelqu'un d'autre que celui auquel on a habituellement recours : Faire une infidélité à son boucher. 6. Manque d'exactitude, de sûreté, de vérité : L'infidélité de la mémoire.

injustice : 1. Caractère de quelqu'un, de quelque chose qui est injuste : Une société fondée sur l'injustice. 2. Acte, décision contraire à la justice : Être victime d'une injustice.

irrégularité : 1. Manque de régularité, de symétrie, d'uniformité de quelque chose, de ponctualité chez quelqu'un : Avoir une certaine irrégularité dans ses horaires. 2. Action irrégulière, contraire à la loi, au règlement : On a relevé certaines irrégularités dans les comptes.

mécompte : Littéraire. Espérance déçue, désillusion : S'exposer à de graves mécomptes.

mégarde : Par mégarde, par inadvertance, sans le vouloir : Marcher par mégarde sur le pied du voisin.

maladresse : 1. Caractère d'une personne maladroite, de ses gestes ou de ses réalisations : Remarquer la maladresse d'un dessin. 2. Défaut d'habileté, de savoir-faire dans la conduite, dans les actions : La maladresse a fait échouer les négociations. 3. Acte maladroit : Accumuler les maladresses.

malentendu : Fait de se méprendre sur quelque chose, en particulier sur le sens d'une parole, d'un mot : Faire cesser un malentendu. 2. Désaccord entre des personnes, né d'une divergence d'interprétation, en particulier d'une mésentente sentimentale : Un douloureux malentendu.

manquement : Action de manquer à un devoir, à une loi, à une règle : De graves manquements au règlement.

méprise : Erreur de celui qui se trompe sur quelqu'un, sur quelque chose : Être victime d'une méprise.

mensonge : 1. Action de mentir, de déguiser, d'altérer la vérité : Le mensonge sert de fondement à sa politique. 2. Assertion contraire à la vérité : Ce que vous dites est un mensonge. 3. Littéraire. Ce qui est faux, illusoire, trompeur : Leur bonheur est un mensonge.

non-sens : 1. Ce qui manque de sens ; phrase ou parole dépourvue de sens : Faire des non-sens dans une version latine. 2. Défi au bon sens, chose absurde : Cette entreprise est un non-sens économique.

paralogisme : Raisonnement faux dans sa forme.

plantage : Familier. Fait de se planter.

préjugé : 1. Jugement sur quelqu'un, quelque chose, qui est formé à l'avance selon certains critères personnels et qui oriente en bien ou en mal les dispositions d'esprit à l'égard de cette personne, de cette chose : Avoir un préjugé contre quelqu'un. 2. Opinion adoptée sans examen, souvent imposée par le milieu, l'éducation : Avoir les préjugés de sa caste.

quiproquo : Méprise par laquelle une personne, une chose est prise pour une autre.

sophisme : 1. Argument qui, partant de prémisses vraies, ou jugées telles, aboutit à une conclusion absurde et difficile à réfuter. 2. Raisonnement vicié à la base reposant sur un jeu de mots, un argument séduisant mais faux, destiné à induire l'interlocuteur en erreur.

tromperie : 1. Ce qui trompe ou vise à tromper. 2. Délit consistant à induire une personne en erreur sur les marchandises qu'on lui livre ou la prestation de services qu'on effectue.

Appendix B

Workshop Reports

B.1 Workshop 1 (HCI researchers)

For the HCI research group, the **PROBLEM WITH THE EQUIPMENT** category was too broad and poorly named. A **PROBLEM WITH THE EQUIPMENT** could be hardware or low-level software to them, even though a participant pointed out that it is never completely one or the other. Also, the problems are not the same when talking about slowness or complete crashes. In the **DISCUSSION PHASE**, they proposed splitting this category into two: *“Design Fault”** and *“Accident/Breakdown”**. A participant also raised the issue of planned obsolescence.

The **ERROR IN PLANNING** category raised questions for them (do we really plan or do we always react to the present? Is the time scale for planning too long?), even though they brought up many examples.

YOUR INTERPRETATION seemed clear to them, and they wanted to rename it by focusing on a feedback flaw and removing the word “your” from the name. A participant pointed out that there is a spectrum between a user interpretation error and a design error.

Regarding **WRONG SYSTEM INTERPRETATION**, they found it strange to talk about interpretation, because *“the system just does what it’s told”**. They found that this category was the counterpart to **WRONG USER INTERPRETATION**; it depends on where you place the blame. For these last two categories of **WRONG SYSTEM INTERPRETATION** and **WRONG USER INTERPRETATION**, during the **DISCUSSION PHASE**, they raised that it depends on whose fault it is.

For **INTERRUPTION**, they provided examples that opened up other dimensions: 1- an interruption may fail to distract us when we wanted to deal with its content 2- Is getting lost on the internet an interruption? In any case, they did not like the name, which they felt does not correspond to the phenomenon: *“distraction”** is better. The separation into two categories of interruption and distraction was mentioned by a participant in the **DISCUSSION PHASE**.

INTERACTION INTERFERENCE was already mentioned in **WRONG SYSTEM INTERPRETATION**, where they discussed the link with “race condition”. They propose the names *“Asynchrony”** (*“Asynchrone”*), *“Interception”** (*“Interception”*), and *“Cross fire mishap”** (*“Balle perdue”*). They found examples in **ACTION** but did not like the name, preferring *“motor error”**, *“context error”** or *“execution inaccuracy”**.

The same applied to **SLIP** (namely *“Bad reflex”**) where they said that it should not say “bad” in the title and blame the user when it is actually a problem of interface consistency. A participant suggested *“Contextual action”** and another *“Interface inconsistency”**.

For **FAIL-UNSAFE**, we had examples that raised questions: what type of error occurs when the backup system crashes or freezes? What if the backups exist but are too old? They distinguished between conscious (laziness) and unconscious (unaware) lack of prevention. They suggested the names *“Non-prevention”**, *“Procrastination”**, and *“Laziness”**. In the **DISCUSSION PHASE**, they proposed splitting this category into two based on this difference.

Finally, for **OMISSION**, they found examples. They distinguished between forgetting an action and forgetting knowledge. The division of forgetfulness into two categories was quickly mentioned by a participant in the **DISCUSSION PHASE**.

In the **DISCUSSION PHASE**, they gave a few additional examples of errors. A participant mentioned that there is no category for “The system does not do what it is supposed to do”, or that creating a category for *“lack of interface clarity”** would be another way of looking at

things. Another person raised the case where “you don’t know if it’s something that’s there but you can’t find it, or something that doesn’t exist”, and they found examples of this.

B.2 Workshop 2 (baseline)

For the group of regular users, examples of **PROBLEM WITH THE EQUIPMENT** were easy to find. However, one example was unclear because they did not know what happened; they did not know what caused a bug when making an online payment. The name of the category raised questions for two participants: for them, **PROBLEM WITH THE EQUIPMENT** must be “*something physical that you can touch*”^{*}; an application is not hardware. One participant suggested adding to the description that the tool can “*lose a functionality*”^{*}, and not necessarily completely cease to function. They suggested “*Hardware and network*”^{*} and “*Hardware and system*”^{*} as other names. During the **DISCUSSION PHASE**, they decided to place accessibility issues in the **PROBLEM WITH THE EQUIPMENT** category. They also wanted to divide **PROBLEM WITH THE EQUIPMENT** into two categories: system and design.

For **ERROR IN PLANNING**, one person wondered whether accessibility issues would fall into this category (website inaccessible to the visually impaired) and put the post-it note aside. Another participant talked about a planning error in refusing to watch a tutorial before using software, but they wanted to do the task, not learn how to use the software. Two of them talked about feeling guilty before and after a task when they failed, feeling guilty for not being able to use the equipment because they are “*bad at computers*”^{*}. One participant anticipated that a situation would make her angry and in turn delegated certain tasks. They found that this category was very user-oriented, that it was clear that it was the user’s fault. One of them pointed out that there is a blur between a feature that should exist but is missing, and a user who does not know that the feature exists. They would have liked to add the nuance of user or system error to the description.

They found several relevant examples for **WRONG USER INTERPRETATION**. They quickly found themselves discussing Dark Patterns (overly long texts, cookies, changing the position of the cross to close ads). One participant also mentioned the lack of simplification of system messages, which she sometimes did not understand.

WRONG SYSTEM INTERPRETATION was met with an immediate collective “Oula”³⁴ in response to the title, but they found several relevant examples. Two participants disagreed that the system could misinterpret a click. For one of them, the system just does what it is told, but they pointed out that you would have to be an expert to never make a mistake. For another person, the notion of interpretation was problematic: it is not a machine thing, but a human thing. Finally, they would have liked to add a reference to the system’s language skills in the description to distinguish them from the interpretation of a click: they are not the same thing.

They also found examples of **INTERRUPTION** easily. One of them talked about ADHD and spontaneous distractions that were not caused by the computer. Another person talked about a download that took too long and caused her to stray from her task. Someone gave the example of a slow computer, which can also disrupt concentration. They suggested “*Attention diversion*”^{*} or “*Disturbance*”^{*} as another category name, because “we’re not really talking about something that blocks us”.

They also had no trouble finding examples of **INTERACTION INTERFERENCES**, thanks to those already given. One of them talked about a dark pattern with interference to get you to buy a mobile game. Two people talked about killing an ally or NPC in a video game. A participant also mentioned that the phenomenon happens more often on phones, where there are more micro-decisions to make and the interactivity is faster. A person gave the example of a patient list that can update just as you are about to click on it. For the name, they would have liked to add the notion of surprise and speed, for example through “*Untimeliness*”^{*}, “*Interruption*”^{*}, or “*Irruption*”^{*}.

For **ACTION**, they found examples related to text selection and nested scrolls. For one person, screen calibration is a hardware error. They found the name of **ACTION** too general.

³⁴A French interjection that is difficult to translate, expresses a sentiment similar to “oh dear”

One of them indicated that the category was based on user error. They suggested the names “*Incorrect Action*”, “*Unexpected Action*”, “*Incorrect Calibration*”, and “*Clumsiness*”.

For **SLIP**, they found many relevant examples. They mentioned keyboard shortcuts, alert sounds (music/sounds that resembled a ringtone or notification), skipping stories, deleting notifications, and closing apps. They had no issues with the name and description.

For **FAIL-UNSAFE**, they had many examples of significant consequences of a lack or insufficiency of prevention. Two of them said that “it has to happen once” to finally put the necessary preventive actions in place and stop “being lazy”. One person talked about the fact that sometimes it’s not worth it (the effort of sorting/saving 40 photos of your cat). They had no comments on the name and description.

For **OMISSION**, they talked about forgetting attachments, alarms, and **FAIL-UNSAFE**. They questioned this category because it overlaps with **FAIL-UNSAFE**, **ERROR IN PLANNING**, and **INTERRUPTION**. During the **DISCUSSION PHASE**, they considered removing the category. One of them proposed counting it in **ACTION** as a “non-action”, but they were unable to reach a decision.

In the **DISCUSSION PHASE**, they proposed an additional category on respecting user consent.

B.3 Workshop 3 (IT professionals)

For the group of computer scientists, the **Hardware** category prompted them to write rather critical examples, where the use of a computer, laptop, or Wi-Fi network becomes impossible or severely limited. Based on these examples, they suggested the name “*Mechanical Bug*” or “*Breakdown*” and added the notion of unexpected or sudden to the description.

For **ERROR IN PLANNING**, they talked about differences between operating systems that are not taken into account in the action and hasty action (particularly regarding code). They discussed the category, focusing on the fact that understanding the tool is essential for good planning, and that the task itself is often simple. Also, the plan is made as they go along, “instinctively”.

For **YOUR INTERPRETATION**, participants gave examples of interfaces that deviated from the standard, and commands that are used because they “work in 90% of cases”. They suggested the name “*User interpretation*” and criticized the category: sometimes the user does not even try to interpret, and sometimes these “errors” are part of a learning process.

For **WRONG SYSTEM INTERPRETATION**, the examples focused on touchscreen issues (cold fingers, latency), voice recognition, and ChatGPT. They found the category particularly relevant/clear and easily found examples for it. They suggested “*Interpretation by the system*” as a name to avoid any ambiguity.

For **INTERRUPTION**, they discussed autonomous distractions and notifications that more or less require immediate action. They discussed who is to blame for the phenomenon and proposed the term “*cognitive disruption*”.

For the **INTERACTION INTERFERENCE** category, they offered quite a few examples, some of which were related to the phenomenon but were not actually interference (nested scrolls, overly complex interface). Latency can cause interference with the user: you click, you see no effect, you click again, you reverse the effect through interference. They wanted to change the name of the category and suggested “*volatile interface*”, “*latency*”, and “*decorrelation*”. In the **DISCUSSION PHASE**, they talked about the overlap between **INTERRUPTION** and **INTERACTION INTERFERENCE**.

For **ACTION**, their examples were numerous and relevant, focusing on accidental clicks. They found the category close to **INTERACTION INTERFERENCE**, and agreed to blame the user. They suggested the names “*Failed reaction*”, “*Automatic response*”, “*User error*”, “*Careless user*”, and “*Lack of user precision*”.

SLIP spoke to them, they gave examples related to disrupting a habit, and appreciated the name of the category. In the **DISCUSSION PHASE**, they discussed the overlaps between **ACTION** and **SLIP**.

For **FAIL-UNSAFE**, they discussed instances where not updating is an act of prevention in itself, because one does not want to leave a situation that works for them, even knowing that it is not permanent. Once again, according to them, the problem lies entirely with the user.

They suggested the names *"Forgetfulness"*, *"Refusal/Denial of Prevention"*, and *"Ignorance of Risks"*. They likened this category to **SLIP**.

Examples of **OMISSION** ranged from forgetting to recompile before retesting code to forgetting to send a message to which a response is expected. The category was clear to them. In the **DISCUSSION PHASE**, they discussed the overlap between **ERROR IN PLANNING** and **OMISSION**.

They proposed adding the categories *"Lack of knowledge of the tool"* and *"Poor ergonomic choices"*.

B.4 Workshop 4 (baseline)

For the group of experienced office workers, the category **PROBLEM WITH THE EQUIPMENT** inspired examples related to latency and freezing, internet box problems, and failure to start up. They would have liked to see the word "bug" mentioned somewhere, and wondered where to place connection problems between the hardware and the system.

For **ERROR IN PLANNING**, they emphasized that these are problems of user incompetence or ignorance, while giving their examples. No comments on the name or description.

For **YOUR INTERPRETATION**, they mentioned commands in English or German, similar icons that are unclear or change position in the interface, and keyboard shortcuts that do not always perform the same actions depending on the context. They suggested adding *"incorrect"* or *"misunderstood"* to the description.

To illustrate **WRONG SYSTEM INTERPRETATION**, they gave examples of interaction interference, automatic correction, and confusion between regaining focus on the application and clicking on a widget. They suggested adding *"by"* the system to the name. For them, the category concerns the system's inability to predict the user.

For **INTERRUPTION**, they gave a few examples and emphasized that it was often their fault, due to non-digital concentration problems. They suggested the name *"disruption"*.

INTERACTION INTERFERENCE led to confusing examples about the nature of the phenomenon. For half of the participants, interferences are only of the pop-up type. They would have added the notion of latency or delay to the description, or even separated this category into two: *"pop-up type"* and *"latency"*. In the **DISCUSSION PHASE**, some suggested combining **INTERACTION INTERFERENCE** and **INTERRUPTION**, since both phenomena stem from the same causes (pop-ups) and have the same consequences (loss of concentration).

The examples on **ACTION** focused quite a bit on touch recognition (cold, dirty, wet fingers) and changes from usual habits (screen size, keyboard layout). They suggested the title *"Beside"* and adding the concept of *"aiming"* to the description.

On **SLIP**, they gave examples of forgetting attachments, different shortcuts depending on the app, and learned reflexes that are no longer relevant (accepting all options in office tools creates bad habits with cookies on the web).

For **FAIL-UNSAFE**, the examples related to technical problems with backup tools for their work tools, vandalism of internet cables, and the lack of backups. They would have liked to see the consequences of inaction specified in the description.

For **OMISSION**, many examples revolved around emails (forgetting attachments, subject lines, or closing phrases), but they also mentioned forgetting to save or recharge a laptop. Once again, they wanted to clarify the consequences of forgetfulness in the description. In the **DISCUSSION PHASE**, they considered a link between prevention and forgetfulness, since *"one cannot exist without the other"*.

For other possible types of problems, they suggested the issue of fake news and software design problems for a professional tool that is not adapted to the reality of the work.

Bibliography

- [1] ACM Library. <https://dl.acm.org>.
- [2] Piotr D. Adamczyk and Brian P. Bailey. "If Not Now, When? The Effects of Interruption at Different Moments within Task Execution". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '04. Vienna, Austria: Association for Computing Machinery, 2004, pp. 271–278. ISBN: 1581137028. DOI: 10.1145/985692.985727. URL: <https://doi.org/10.1145/985692.985727>.
- [3] Shamel Addas and Alain Pinsonneault. "The Many Faces of Information Technology Interruptions: A Taxonomy and Preliminary Investigation of Their Performance Effects". In: *Info. Sys. Jour.* 25.3 (May 2015), pp. 231–273. ISSN: 1350-1917. DOI: 10.1111/isj.12064. URL: <https://doi.org/10.1111/isj.12064>.
- [4] Amina Ahmed, Lucas Pereira, and Kimberly Jane. "Mixed Methods Research: Combining both qualitative and quantitative approaches". In: (Sept. 2024).
- [5] Erik M Altmann and J.Gregory Trafton. "Memory for goals: an activation-based model". In: *Cognitive Science* 26.1 (2002), pp. 39–83. ISSN: 0364-0213. DOI: [https://doi.org/10.1016/S0364-0213\(01\)00058-1](https://doi.org/10.1016/S0364-0213(01)00058-1). URL: <https://www.sciencedirect.com/science/article/pii/S0364021301000581>.
- [6] Maartje Ga Ament et al. "Making a Task Difficult: Evidence That Device-Oriented Steps Are Effortful and Error-Prone". In: *Journal of Experimental Psychology: Applied* 19.3 (2013), p. 195. DOI: 10.1037/a0034397.
- [7] Caroline Appert, Olivier Chapuis, and Emmanuel Pietriga. "Dwell-and-Spring: Undo for Direct Manipulation". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: Association for Computing Machinery, 2012, pp. 1957–1966. ISBN: 9781450310154. DOI: 10.1145/2207676.2208339. URL: <https://doi.org/10.1145/2207676.2208339>.
- [8] Caroline Appert et al. "Reciprocal Drag-and-Drop". In: *ACM Trans. Comput.-Hum. Interact.* 22.6 (Sept. 2015). ISSN: 1073-0516. DOI: 10.1145/2785670. URL: <https://doi.org/10.1145/2785670>.
- [9] Algirdas Avizienis et al. "Basic Concepts and Taxonomy of Dependable and Secure Computing". In: *IEEE Trans. Dependable Secur. Comput.* 1.1 (Jan. 2004), pp. 11–33. ISSN: 1545-5971. DOI: 10.1109/TDSC.2004.2. URL: <https://doi.org/10.1109/TDSC.2004.2>.
- [10] Ebrahim Babaei et al. "Should we use the NASA-TLX in HCI? A review of theoretical and methodological issues around Mental Workload Measurement". In: *International Journal of Human-Computer Studies* 201 (2025), p. 103515. ISSN: 1071-5819. DOI: <https://doi.org/10.1016/j.ijhcs.2025.103515>. URL: <https://www.sciencedirect.com/science/article/pii/S1071581925000722>.

- [11] Jonathan Back, Ann Blandford, and Paul Curzon. "Slip Errors and Cue Salience". In: *Proceedings of the 14th European Conference on Cognitive Ergonomics: Invent! Explore! ECCE '07*. London, United Kingdom: Association for Computing Machinery, 2007, pp. 221–224. ISBN: 9781847998491. DOI: 10.1145/1362550.1362595. URL: <https://doi.org/10.1145/1362550.1362595>.
- [12] Jonathan Back, Duncan P. Brumby, and Anna L. Cox. "Locked-out: investigating the effectiveness of system lockouts to reduce errors in routine tasks". In: *CHI '10 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '10. Atlanta, Georgia, USA: Association for Computing Machinery, 2010, pp. 3775–3780. ISBN: 9781605589305. DOI: 10.1145/1753846.1754054. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/1753846.1754054>.
- [13] Jonathan Back, Anna Cox, and Duncan Brumby. "Choosing to interleave: human error and information access cost". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: Association for Computing Machinery, 2012, pp. 1651–1654. ISBN: 9781450310154. DOI: 10.1145/2207676.2208289. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/2207676.2208289>.
- [14] Soheila Sheikh Bahaei et al. "Effect of Augmented Reality on Faults Leading to Human Failures in Socio-technical Systems". In: *2019 4th International Conference on System Reliability and Safety (ICSRS)*. Nov. 2019, pp. 236–245. DOI: 10.1109/ICSRS48664.2019.8987586.
- [15] Brian P. Bailey and Joseph A. Konstan. "On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state". In: *Computers in Human Behavior* 22.4 (2006). Attention aware systems, pp. 685–708. ISSN: 0747-5632. DOI: <https://doi.org/10.1016/j.chb.2005.12.009>. URL: <https://www.sciencedirect.com/science/article/pii/S074756320500107X>.
- [16] Victoria A. Banks, Katherine L. Plant, and Neville A. Stanton. "Driver error or designer error: Using the Perceptual Cycle Model to explore the circumstances surrounding the fatal Tesla crash on 7th May 2016". In: *Safety Science* 108 (2018), pp. 278–285. ISSN: 0925-7535. DOI: <https://doi.org/10.1016/j.ssci.2017.12.023>. URL: <https://www.sciencedirect.com/science/article/pii/S0925753517314212>.
- [17] Luiz Carlos Begosso and Lucia Vilela Leite Filgueiras. "Human Error Simulation as an Aid to HCI Design for Critical Systems". In: *Proceedings of VII Brazilian Symposium on Human Factors in Computing Systems*. IHC '06. Natal, RN, Brazil: Association for Computing Machinery, 2006, pp. 120–127. ISBN: 1595934324. DOI: 10.1145/1298023.1298040. URL: <https://doi.org/10.1145/1298023.1298040>.
- [18] Hortense Blazsin and Frank Guldenmund. "The social construction of safety: Comparing three realities". In: *Safety Science* 71 (2015). Editors' corner 2013, pp. 16–27. ISSN: 0925-7535. DOI: <https://doi.org/10.1016/j.ssci.2014.06.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0925753514001349>.
- [19] Matthias Böhmer et al. "Interrupted by a phone call: exploring designs for lowering the impact of call notifications for smartphone users". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI

- '14. Toronto, Ontario, Canada: Association for Computing Machinery, 2014, pp. 3045–3054. ISBN: 9781450324731. DOI: 10.1145/2556288.2557066. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/2556288.2557066>.
- [20] Paul A. Booth. "Errors and theory in human-computer interaction". In: *Acta Psychologica* 78.1 (1991), pp. 69–96. ISSN: 0001-6918. DOI: [https://doi.org/10.1016/0001-6918\(91\)90005-K](https://doi.org/10.1016/0001-6918(91)90005-K). URL: <https://www.sciencedirect.com/science/article/pii/000169189190005K>.
- [21] Paul A. Booth. "Identifying and interpreting design errors". In: *International Journal of Human-Computer Interaction* 2.4 (1990), pp. 307–332. DOI: 10.1080/10447319009525988. eprint: <https://doi.org/10.1080/10447319009525988>. URL: <https://doi.org/10.1080/10447319009525988>.
- [22] Judith Borghouts et al. "Visualizing Magnitude: Graphical Number Representations Help Users Detect Large Number Entry Errors". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 59.1 (2015), pp. 591–595. DOI: 10.1177/1541931215591130. eprint: <https://doi.org/10.1177/1541931215591130>. URL: <https://doi.org/10.1177/1541931215591130>.
- [23] Virginia Braun and Victoria Clarke. *Thematic Analysis - A Practical Guide*. London EC1Y 1SP: SAGE Publications Ltd, 2022. ISBN: 978-1-4739-5323-9.
- [24] Virginia Braun and Victoria Clarke. "Using thematic analysis in psychology". In: *Qualitative Research in Psychology* 3.2 (2006), pp. 77–101. DOI: 10.1191/1478088706qp063oa. URL: <https://www.tandfonline.com/doi/abs/10.1191/1478088706qp063oa>.
- [25] H Brignull et al. *Deceptive patterns – user interfaces designed to trick you*. Apr. 2023. URL: <https://www.deceptive.design/>.
- [26] John Brooke. "SUS-A Quick and Dirty Usability Scale". In: *Usability evaluation in industry* 189.194 (1996).
- [27] Marc Brysbaert. "How many words do we read per minute? A review and meta-analysis of reading rate". In: *Journal of Memory and Language* 109 (2019), p. 104047. ISSN: 0749-596X. DOI: <https://doi.org/10.1016/j.jml.2019.104047>. URL: <https://www.sciencedirect.com/science/article/pii/S0749596X19300786>.
- [28] Wolfgang Büschel et al. "Investigating Smartphone-Based Pan and Zoom in 3D Data Spaces in Augmented Reality". In: *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '19. Taipei, Taiwan: Association for Computing Machinery, 2019. ISBN: 9781450368254. DOI: 10.1145/3338286.3340113. URL: <https://doi.org/10.1145/3338286.3340113>.
- [29] Michael D Byrne and Elizabeth M Davis. "Task structure and postcompletion error in the execution of a routine procedure". In: *Human Factors* 48.4 (2006), pp. 627–638.
- [30] *Cambridge Dictionary*. <https://dictionary.cambridge.org>.
- [31] Stuart K. Card, Allen Newell, and Thomas P. Moran. *The Psychology of Human-Computer Interaction*. USA: L. Erlbaum Associates Inc., 1983. ISBN: 0898592437.

- [32] John S. Carroll et al. "The variety of beliefs about the causes of safety among safety practitioners". In: *Safety Science* 148 (2022), p. 105641. ISSN: 0925-7535. DOI: <https://doi.org/10.1016/j.ssci.2021.105641>. URL: <https://www.sciencedirect.com/science/article/pii/S0925753521004811>.
- [33] Géry Casiez, Nicolas Roussel, and Daniel Vogel. "1 € filter: a simple speed-based low-pass filter for noisy input in interactive systems". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: Association for Computing Machinery, 2012, pp. 2527–2530. ISBN: 9781450310154. DOI: 10.1145/2207676.2208639. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/2207676.2208639>.
- [34] Irina Ceaparu et al. "Determining causes and severity of end-user frustration". In: *International journal of human-computer interaction* 17.3 (2004), pp. 333–356.
- [35] Parmit K. Chilana, Amy J. Ko, and Jacob O. Wobbrock. "Understanding Expressions of Unwanted Behaviors in Open Bug Reporting". In: *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*. 2010, pp. 203–206. DOI: 10.1109/VLHCC.2010.35.
- [36] Martin Constable. "Deconstructing an old master painting using photoshop's advanced toolset". In: *ACM SIGGRAPH ASIA 2008 Educators Programme*. SIGGRAPH Asia '08. Singapore: Association for Computing Machinery, 2008. ISBN: 9781605583884. DOI: 10.1145/1507713.1507731. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/1507713.1507731>.
- [37] Alan Cooper. *The Myth of the Metaphor*. <https://medium.com/@MrAlanCooper/the-myth-of-metaphor-de16bfa1cbbe>. Visual Basic Programmers Journal. 1995. URL: <https://medium.com/@MrAlanCooper/the-myth-of-metaphor-de16bfa1cbbe>.
- [38] M.D. Cooper. "The Emperor has no clothes: A critique of Safety-II". In: *Safety Science* 152 (2022), p. 105047. ISSN: 0925-7535. DOI: <https://doi.org/10.1016/j.ssci.2020.105047>. URL: <https://www.sciencedirect.com/science/article/pii/S0925753520304446>.
- [39] Michelle V Cormier et al. "This Game SUX: Why & How to Design Sh@*!y User Experiences". In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. CHI '25. Association for Computing Machinery, 2025. ISBN: 9798400713941. DOI: 10.1145/3706598.3713246. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3706598.3713246>.
- [40] Andy Crabtree. "H is for human and how (not) to evaluate qualitative research in HCI". In: *Human-Computer Interaction* 0.0 (2025), pp. 1–24. DOI: 10.1080/07370024.2025.2475743. eprint: <https://doi.org/10.1080/07370024.2025.2475743>. URL: <https://doi.org/10.1080/07370024.2025.2475743>.
- [41] Martin Cronel et al. "MIODMIT: A Generic Architecture for Dynamic Multimodal Interactive Systems". In: *Human-Centered Software Engineering*. Ed. by Cristian Bogdan et al. Cham: Springer International Publishing, 2019, pp. 109–129. ISBN: 978-3-030-05909-5.

- [42] Sidney Dekker. *Drift into Failure: From Hunting Broken Components to Understanding Complex Systems*. Jan. 2011, pp. 1–220. ISBN: 9781315257396. DOI: 10.1201/9781315257396.
- [43] Sidney Dekker. *Foundations of Safety Science : A Century of Understanding Accidents and Disasters*. USA: CRC Press, Taylor & Francis Group, 2019. DOI: 10.4324/9781351059794.
- [44] Sidney Dekker. *The Field Guide to Understanding Human Error*. Aldershot: Ashgate, 2006.
- [45] Sidney Dekker, Daniel Siegenthaler, and Tom Laursen. “Six stages to the new view of human error”. In: *Safety Science Monitor* 11 (Jan. 2007).
- [46] Ameet Deshpande et al. “Anthropomorphization of AI: Opportunities and Risks”. In: *Proceedings of the Natural Legal Language Processing Workshop 2023*. Ed. by Daniel Preoțiuc-Pietro et al. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 1–7. DOI: 10.18653/v1/2023.nllp-1.1. URL: <https://aclanthology.org/2023.nllp-1.1/>.
- [47] Vivek Dhakal et al. “Observations on Typing from 136 Million Keystrokes”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, pp. 1–12. ISBN: 9781450356206. DOI: 10.1145/3173574.3174220. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3173574.3174220>.
- [48] Linda Di Geronimo et al. “UI Dark Patterns and Where to Find Them: A Study on Mobile Applications and User Perception”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–14. ISBN: 9781450367080. DOI: 10.1145/3313831.3376600. URL: <https://doi.org/10.1145/3313831.3376600>.
- [49] “Quality Concepts and Terminology” part 1: Geberic Terms and Definitions. Standard. International Organization for Standardization, Feb. 1992.
- [50] Lavallee Elise. “Pitt pauses testing on food delivery robots following reports of impeded accessibility”. In: *Pittsburg City Paper* (Oct. 2019). URL: <https://www.pghcitypaper.com/news/pitt-pauses-testing-on-food-delivery-robots-following-reports-of-impeded-accessibility-16050658>.
- [51] David Embrey. “Human Error Understanding Human Behaviour and Error”. In: *Hum. Reliab. Associates* 1 (Jan. 2005).
- [52] Will Epperson et al. “Interactive Debugging and Steering of Multi-Agent AI Systems”. In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. CHI '25. Association for Computing Machinery, 2025. ISBN: 9798400713941. DOI: 10.1145/3706598.3713581. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3706598.3713581>.
- [53] Erik Hollnagel’s website. <https://erikhollnagel.com/>. URL: <https://erikhollnagel.com/>.

- [54] Abigail Evans and Jacob Wobbrock. "Taming wild behavior: the input observer for obtaining text entry and mouse pointing measures from everyday computer use". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: Association for Computing Machinery, 2012, pp. 1947–1956. ISBN: 9781450310154. DOI: 10.1145/2207676.2208338. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/2207676.2208338>.
- [55] Racim Fahssi, Célia Martinie, and Philippe Palanque. "Enhanced Task Modelling for Systematic Identification and Explicit Representation of Human Errors". In: *Human-Computer Interaction – INTERACT 2015*. Ed. by Julio Abascal et al. Cham: Springer International Publishing, 2015, pp. 192–212. ISBN: 978-3-319-22723-8.
- [56] Guillaume Faure, Olivier Chapuis, and Michel Beaudouin-Lafon. "Acquisition of Animated and Pop-Up Targets". In: *Human-Computer Interaction – INTERACT 2009*. Ed. by Tom Gross et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 372–385. ISBN: 978-3-642-03658-3.
- [57] Dan Fitton and Janet C. Read. "Creating a Framework to Support the Critical Consideration of Dark Design Aspects in Free-to-Play Apps". In: *Proceedings of the 18th ACM International Conference on Interaction Design and Children*. IDC '19. Boise, ID, USA: Association for Computing Machinery, 2019, pp. 407–418. ISBN: 9781450366908. DOI: 10.1145/3311927.3323136. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3311927.3323136>.
- [58] Paul M Fitts. "The information capacity of the human motor system in controlling the amplitude of movement." In: *Journal of experimental psychology* 47.6 (1954), p. 381. DOI: <https://doi.org/10.1037/h0055392>.
- [59] Paul Morris Fitts and Richard E Jones. *Analysis of factors contributing to 460" pilot-error" experiences in operating aircraft controls*. Aero Medical Laboratory, 1947.
- [60] FRAM. <https://functionalresonance.com/>. URL: <https://functionalresonance.com/>.
- [61] David Gertman et al. "The SPAR-H human reliability analysis method". In: Jan. 2004.
- [62] Silvia Gherardi, Davide Nicolini, and Francesca Odella. "What Do You Mean By Safety? Conflicting Perspectives on Accident Causation and Safety Management in a Construction Firm". In: *Journal of Contingencies and Crisis Management* 6.4 (1998), pp. 202–213. DOI: <https://doi.org/10.1111/1468-5973.00089>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/1468-5973.00089>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-5973.00089>.
- [63] Tony Gillie and Donald Broadbent. "What makes interruptions disruptive? A study of length, similarity, and complexity". In: *Psychological research* 50.4 (1989), pp. 243–250.
- [64] Donald Golden. "A plea for friendly software". In: *SIGSOFT Softw. Eng. Notes* 5.4 (Oct. 1980), pp. 4–5. ISSN: 0163-5948. DOI: 10.1145/1010884.1010885. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/1010884.1010885>.

- [65] Julien Gori, Olivier Rioul, and Yves Guiard. "Speed-Accuracy Tradeoff: A Formal Information-Theoretic Transmission Scheme (FITTS)". In: *ACM Trans. Comput.-Hum. Interact.* 25.5 (Sept. 2018). ISSN: 1073-0516. DOI: 10.1145/3231595. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3231595>.
- [66] Eryn Grant et al. "Back to the future: What do accident causation models tell us about accident prediction?" In: *Safety Science* 104 (2018), pp. 99–109. ISSN: 0925-7535. DOI: <https://doi.org/10.1016/j.ssci.2017.12.018>. URL: <https://www.sciencedirect.com/science/article/pii/S0925753517306720>.
- [67] Colin M. Gray, Cristiana Santos, and Nataliia Bielova. "Towards a Preliminary Ontology of Dark Patterns Knowledge". In: *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI EA '23. Hamburg, Germany: Association for Computing Machinery, 2023. ISBN: 9781450394222. DOI: 10.1145/3544549.3585676. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3544549.3585676>.
- [68] Colin M. Gray et al. "The Dark (Patterns) Side of UX Design". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, pp. 1–14. ISBN: 9781450356206. DOI: 10.1145/3173574.3174108. URL: <https://doi.org/10.1145/3173574.3174108>.
- [69] Marc Halbrügge, Michael Quade, and Klaus-Peter Engelbrecht. "A Predictive Model of Human Error based on User Interface Development Models and a Cognitive Architecture". In: Jan. 2015, pp. 238–243. ISBN: 9789036777636.
- [70] Marc Halbrügge, Michael Quade, and Klaus-Peter Engelbrecht. "Cognitive Strategies in HCI and Their Implications on User Error". In: Jan. 2016, pp. 2549–2554. ISBN: 9780991196739.
- [71] Maeda F Hanafi et al. "Diagnosing and Prioritizing Issues in Automated Order-Taking Systems: A Machine-Assisted Error Discovery Approach". In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. CHI '25. Association for Computing Machinery, 2025. ISBN: 9798400713941. DOI: 10.1145/3706598.3713312. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3706598.3713312>.
- [72] Donna Haraway. "Situated knowledges: The science question in feminism and the privilege of partial perspective 1". In: *Women, science, and technology*. Routledge, 2013, pp. 455–472.
- [73] S. G. Hart and L. E. Staveland. "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research". In: *Human Mental Workload*. Ed. by P. A. Hancock and N. Meshkati. Amsterdam: North Holland Press, 1988.
- [74] Keith C. Hendy. "A Tool for Human Factors Accident Investigation, Classification and Risk Management". In: 2003. URL: <https://api.semanticscholar.org/CorpusID:111504832>.
- [75] Morten Hertzum and Kasper Hornbæk. "Frustration: Still a Common User Experience". In: *ACM Trans. Comput.-Hum. Interact.* 30.3 (June 2023). ISSN: 1073-0516. DOI: 10.1145/3582432. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3582432>.

- [76] Erik Hollnagel. "Context, cognition, and control". In: *Co-Operation in Process Management-Cognition and Information Technology* (Jan. 1998).
- [77] Erik Hollnagel. *FRAM: The Functional Resonance Analysis Method: Modelling Complex Socio-Technical Systems*. Oct. 2012. ISBN: 978-1-4094-4551-7. DOI: 10.1201/9781315255071.
- [78] Erik Hollnagel. "Position Paper, NATO Conference on Human Error". In: OECD Halden Reactor Project Norway. Bellagio Italy, Sept. 1983.
- [79] Erik Hollnagel. *Safety-I and Safety-II: The Past and Future of Safety Management*. Jan. 2014. ISBN: 978-1-4724-2305-4. DOI: 10.1201/9781315607511.
- [80] Erik Hollnagel. "The phenotype of erroneous actions". In: *International Journal of Man-Machine Studies* 39.1 (1993), pp. 1–32. ISSN: 0020-7373. DOI: <https://doi.org/10.1006/imms.1993.1051>. URL: <https://www.sciencedirect.com/science/article/pii/S0020737383710515>.
- [81] Erik Hollnagel. "The phenotype of erroneous actions: Implications for HCI design". In: *Human-computer Interaction and complex systems* (1991).
- [82] Erik Hollnagel and David Woods. *Joint cognitive systems: Foundations of cognitive systems engineering*. Jan. 2005, pp. 1–223.
- [83] Dugald Ralph Hutchings et al. "Display Space Usage and Window Management Operation Comparisons between Single Monitor and Multiple Monitor Users". In: *Proceedings of the Working Conference on Advanced Visual Interfaces*. AVI '04. Gallipoli, Italy: Association for Computing Machinery, 2004, pp. 32–39. ISBN: 1581138679. DOI: 10.1145/989863.989867. URL: <https://doi.org/10.1145/989863.989867>.
- [84] Mustafa Ibrahim et al. "VERSATILE: Very Fast Partial Reconfiguration Controller". In: *ACM Trans. Reconfigurable Technol. Syst.* 18.3 (Sept. 2025). ISSN: 1936-7406. DOI: 10.1145/3748728. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3748728>.
- [85] *Issue 6071*. Requests repository. URL: <https://github.com/psf/requests/issues/6071>.
- [86] Pichamol Jirapinyo et al. "Preclinical endoscopic training using a part-task simulator: learning curve assessment and determination of threshold score for advancement to clinical endoscopy". In: *Surgical endoscopy* 31 (Oct. 2017). DOI: 10.1007/s00464-017-5436-x.
- [87] Anjali Khurana et al. "Do It For Me vs. Do It With Me: Investigating User Perceptions of Different Paradigms of Automation in Copilots for Feature-Rich Software". In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. CHI '25. Association for Computing Machinery, 2025. ISBN: 9798400713941. DOI: 10.1145/3706598.3713431. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3706598.3713431>.
- [88] Barry Kirwan. "Human error identification techniques for risk assessment of high risk systems—Part 2: towards a framework approach". In: *Applied Ergonomics* 29.5 (1998), pp. 299–318. ISSN: 0003-6870. DOI: [https://doi.org/10.1016/S0003-6870\(98\)00011-8](https://doi.org/10.1016/S0003-6870(98)00011-8). URL: <https://www.sciencedirect.com/science/article/pii/S0003687098000118>.

- [89] Muneo Kitajima and Peter G. Polson. "A Model-Based Analysis of Errors in Display-Based HCI". In: *Conference Companion on Human Factors in Computing Systems*. CHI '94. Boston Mass. USA: Association for Computing Machinery, 1994, pp. 301–302. ISBN: 0897916514. DOI: 10 . 1145 / 259963 . 260508. URL: <https://doi.org/10.1145/259963.260508>.
- [90] Thomas Kosch et al. "A Survey on Measuring Cognitive Workload in Human-Computer Interaction". In: *ACM Comput. Surv.* 55.13s (July 2023). ISSN: 0360-0300. DOI: 10 . 1145 / 3582272. URL: [https : / / doi - org . ressources-electroniques.univ-lille.fr/10.1145/3582272](https://doi-rg.ressources-electroniques.univ-lille.fr/10.1145/3582272).
- [91] Nantia Koulidou et al. "The Development and (Mis)appropriation of a Digital Kit for Jewellers". In: *ACM Trans. Comput.-Hum. Interact.* 32.1 (Apr. 2025). ISSN: 1073-0516. DOI: 10 . 1145 / 3702316. URL: [https : / / doi - org . ressources-electroniques.univ-lille.fr/10.1145/3702316](https://doi-rg.ressources-electroniques.univ-lille.fr/10.1145/3702316).
- [92] Esben Kran et al. "DarkBench: Benchmarking Dark Patterns in Large Language Models". In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: <https://openreview.net/forum?id=odjMSBSWRt>.
- [93] John G Kreifeldt and Mary E McCarthy. "Interruption as a test of the user-computer interface". In: *JPL Proc. of the 17th Ann. Conf. on Manual Control*. 1981.
- [94] Korbinian Kuhn, Verena Kersken, and Gottfried Zimmermann. "Evaluating ASR Confidence Scores for Automated Error Detection in User-Assisted Correction Interfaces". In: *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. CHI EA '25. Association for Computing Machinery, 2025. ISBN: 9798400713958. DOI: 10 . 1145 / 3706599 . 3720038. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3706599.3720038>.
- [95] Xavier de La Porte and Séverine Cassar. *Eloge du bug*. France Inter. Le code a changé, Sept. 2024. URL: [https : / / www . radiofrance . fr / franceinter / podcasts/le-code-a-change/le-code-a-change-9373535](https://www.radiofrance.fr/franceinter/podcasts/le-code-a-change/le-code-a-change-9373535).
- [96] Ben Lafreniere et al. "False Positives vs. False Negatives: The Effects of Recovery Time and Cognitive Costs on Input Error Preference". In: *UIST '21* (2021), pp. 54–68. DOI: 10 . 1145 / 3472749 . 3474735. URL: <https://doi.org/10.1145/3472749.3474735>.
- [97] Kevin C. Lam et al. "More Errors vs. Longer Commands: The Effects of Repetition and Reduced Expressiveness on Input Interpretation Error, Learning, and Effort". In: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI '22. New Orleans, LA, USA: Association for Computing Machinery, 2022. ISBN: 9781450391573. DOI: 10 . 1145 / 3491102 . 3502079. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3491102.3502079>.
- [98] Kevin C. Lam et al. "The Effects of System Interpretation Errors on Learning New Input Mechanisms". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. Yokohama, Japan: Association for Computing Machinery, 2021. ISBN: 9781450380966. DOI: 10 . 1145 / 3411764 . 3445366. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3411764.3445366>.
- [99] Jaron Lanier. *You are not a gadget*. Vintage, 2010.

- [100] Justin Larouzee and Jean-Christophe Le Coze. "Good and bad reasons: The Swiss cheese model and its critics". In: *Safety Science* 126 (2020), p. 104660. ISSN: 0925-7535. DOI: <https://doi.org/10.1016/j.ssci.2020.104660>. URL: <https://www.sciencedirect.com/science/article/pii/S0925753520300576>.
- [101] Justin Larouzee, Franck Guarnieri, and Denis Besnard. "Le modèle de l'erreur humaine de James Reason". In: 2014.
- [102] Jonathan Lazar, Adam Jones, and Ben Shneiderman. "Workplace user frustration with computers: An exploratory investigation of the causes and severity". In: *Behaviour & Information Technology* 25.03 (2006), pp. 239–251.
- [103] Jean Christophe Le Coze. "The 'new view' of human error. Origins, ambiguities, successes and critiques". In: *Safety Science* 154 (2022), p. 105853. ISSN: 0925-7535. DOI: <https://doi.org/10.1016/j.ssci.2022.105853>. URL: <https://www.sciencedirect.com/science/article/pii/S0925753522001928>.
- [104] David Leadbetter et al. "Towards Model Based Prediction of Human Error Rates in Interactive Systems". In: *Proceedings of the 2nd Australasian Conference on User Interface*. AUIC '01. Queensland, Australia: IEEE Computer Society, 2001, pp. 42–49. ISBN: 076950969X.
- [105] Byungjoo Lee et al. "AutoGain: Gain Function Adaptation with Submovement Efficiency Optimization". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1–12. ISBN: 9781450367080. URL: <https://doi.org/10.1145/3313831.3376244>.
- [106] Nancy Leveson. "A new accident model for engineering safer systems". In: *Safety Science* 42.4 (2004), pp. 237–270. ISSN: 0925-7535. DOI: [https://doi.org/10.1016/S0925-7535\(03\)00047-X](https://doi.org/10.1016/S0925-7535(03)00047-X). URL: <https://www.sciencedirect.com/science/article/pii/S092575350300047X>.
- [107] Aaleyah Lewis et al. "Inaccessible and Deceptive: Examining Experiences of Deceptive Design with People Who Use Visual Accessibility Technology". In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. CHI '25. Association for Computing Machinery, 2025. ISBN: 9798400713941. DOI: 10.1145/3706598.3713784. URL: <https://doi-ressources-electroniques.univ-lille.fr/10.1145/3706598.3713784>.
- [108] James R. Lewis. "The System Usability Scale: Past, Present, and Future". In: *International Journal of Human-Computer Interaction* 34 (2018), pp. 577–590. URL: <https://api.semanticscholar.org/CorpusID:51869946>.
- [109] Gordon Logan et al. "On the Ability to Inhibit Thought and Action: General and Special Theories of an Act of Control". In: *Psychological review* 121 (Feb. 2014), pp. 66–95. DOI: 10.1037/a0035230.
- [110] Alice Loizeau, Sylvain Malacria, and Mathieu Nancel. "GUI Behaviors to Minimize Pointing-Based Interaction Interferences". In: *ACM Transactions on Computer-Human Interaction* 31 (2024), pp. 1–34. URL: https://inria.hal.science/hal-04460441/file/popups_hal.pdf.
- [111] Richard Lowry. *One-Way Analysis of Variance for Independent Samples*. URL: <https://web.archive.org/web/20081017161620/http://faculty.vassar.edu/lowry/ch14pt2.html>. Archived from the original (<http://faculty.vassar.edu/lowry/ch14pt2.html>) on 17 October 2008.

- [112] Jamie Luguri and Lior Jacob Strahilevitz. "Shining a Light on Dark Patterns". In: *Journal of Legal Analysis* 13 (1 2021), pp. 43–109. DOI: 10 . 1093 / jla / laaa006. URL: <https://doi.org/10.1093/jla/laaa006>.
- [113] Aditi M. Bhoot, Mayuri A. Shinde, and Wricha P. Mishra. "Towards the Identification of Dark Patterns: An Analysis Based on End-User Reactions". In: *Proceedings of the 11th Indian Conference on Human-Computer Interaction*. IndiahCI '20. Online, India: Association for Computing Machinery, 2021, pp. 24–33. ISBN: 9781450389440. DOI: 10 . 1145 / 3429290 . 3429293. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3429290.3429293>.
- [114] Wendy E. Mackay and Anne-Laure Fayard. "HCI, Natural Science and Design: A Framework for Triangulation across Disciplines". In: *Proceedings of the 2nd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*. DIS '97. Amsterdam, The Netherlands: Association for Computing Machinery, 1997, pp. 223–234. ISBN: 0897918630. DOI: 10 . 1145 / 263552 . 263612. URL: <https://doi.org/10.1145/263552.263612>.
- [115] I. Scott MacKenzie. "Fitts' Law as a Research and Design Tool in Human-Computer Interaction". In: *Human-Computer Interaction* 7.1 (1992), pp. 91–139. DOI: 10 . 1207 / s15327051hci0701 \ _3. eprint: https://doi.org/10.1207/s15327051hci0701_3. URL: https://doi.org/10.1207/s15327051hci0701%5C%5C_3.
- [116] Daniela Markov-Vetter, Eckard Moll, and Oliver Staadt. "Evaluation of 3D selection tasks in parabolic flight conditions: pointing task in augmented reality user interfaces". In: *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*. VR-CAI '12. Singapore, Singapore: Association for Computing Machinery, 2012, pp. 287–294. ISBN: 9781450318259. DOI: 10 . 1145 / 2407516 . 2407583. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/2407516.2407583>.
- [117] Arunesh Mathur, Mihir Kshirsagar, and Jonathan Mayer. "What Makes a Dark Pattern... Dark? Design Attributes, Normative Considerations, and Measurement Methods". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI '21. Yokohama, Japan: Association for Computing Machinery, 2021. ISBN: 9781450380966. DOI: 10 . 1145 / 3411764 . 3445610. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3411764.3445610>.
- [118] Ryan D. McKendrick and Erin Cherry. "A Deeper Look at the NASA TLX and Where It Falls Short". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 62.1 (2018), pp. 44–48. DOI: 10 . 1177 / 1541931218621010. eprint: <https://doi.org/10.1177/1541931218621010>. URL: <https://doi.org/10.1177/1541931218621010>.
- [119] Philippe Meirieu. *Faire l'école, faire la classe*. ESF éditeur, 2004.
- [120] Merriam-Webster, *Encyclopaedia Britannica company*. <https://www.merriam-webster.com/>.
- [121] Giovanni Mirabella et al. "Inhibitory control of reaching movements in humans." In: *Experimental Brain Research* 174.2 (Sept. 2006), pp. 240–255. DOI: 10 . 1007 / s00221 - 006 - 0456 - 0.

- [122] Alberto Monge Roffarello and Luigi De Russis. "Towards Understanding the Dark Patterns That Steal Our Attention". In: *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*. CHI EA '22. New Orleans, LA, USA: Association for Computing Machinery, 2022. ISBN: 9781450391566. DOI: 10.1145/3491101.3519829. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3491101.3519829>.
- [123] Jörg Müller, Antti Oulasvirta, and Roderick Murray-Smith. "Control Theoretic Models of Pointing". In: *ACM Trans. Comput.-Hum. Interact.* 24.4 (Aug. 2017). ISSN: 1073-0516. DOI: 10.1145/3121431. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3121431>.
- [124] Mathieu Nancel and Andy Cockburn. "Causality: A Conceptual Model of Interaction History". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: Association for Computing Machinery, 2014, pp. 1777–1786. ISBN: 9781450324731. DOI: 10.1145/2556288.2556990. URL: <https://doi.org/10.1145/2556288.2556990>.
- [125] Mathieu Nancel et al. "Mid-air pan-and-zoom on wall-sized displays". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: Association for Computing Machinery, 2011, pp. 177–186. ISBN: 9781450302289. DOI: 10.1145/1978942.1978969. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/1978942.1978969>.
- [126] Mathieu Nancel et al. "Mid-Air Pointing on Ultra-Walls". In: *ACM Trans. Comput.-Hum. Interact.* 22.5 (Aug. 2015). ISSN: 1073-0516. DOI: 10.1145/2766448. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/2766448>.
- [127] Ulric Neisser. *Cognition and reality*. San Francisco: W.H. Freeman, 1987.
- [128] Donald A Norman. "Categorization of action slips." In: *Psychological review* 88.1 (1981), p. 1.
- [129] Donald A Norman. *The psychology of everyday things*. Basic books, 1988.
- [130] Donald A. Norman. "Design Rules Based on Analyses of Human Error". In: *Commun. ACM* 26.4 (Apr. 1983), pp. 254–258. ISSN: 0001-0782. DOI: 10.1145/2163.358092. URL: <https://doi.org/10.1145/2163.358092>.
- [131] Philippe Palanque. "POISE: A Framework for Designing Perfect Interactive Systems with and for Imperfect People". In: *Human-Computer Interaction – INTERACT 2021*. Ed. by Carmelo Ardito et al. Cham: Springer International Publishing, 2021, pp. 39–59. ISBN: 978-3-030-85623-6.
- [132] Philippe Palanque, Andy Cockburn, and Carl Gutwin. "A Classification of Faults Covering the Human-Computer Interaction Loop". In: *Computer Safety, Reliability, and Security*. Ed. by António Casimiro et al. Cham: Springer International Publishing, 2020, pp. 434–448. ISBN: 978-3-030-54549-9.
- [133] *Patte de Lou*. URL: https://www.webtoons.com/fr/canvas/moule-sentimentale/list?title_no=815279.
- [134] Dan Payne and James W Altman. *An index of electronic equipment operability: Report of development*. Tech. rep. AMERICAN INSTITUTES FOR RESEARCH PITTSBURGH PA, 1962.

- [135] Stephen J. Payne and T.R.G. Green. "Task-Action Grammars: A Model of the Mental Representation of Task Languages". In: *Human-Computer Interaction* 2.2 (1986), pp. 93–133. DOI: 10.1207/s15327051hci0202_1. eprint: https://www.tandfonline.com/doi/pdf/10.1207/s15327051hci0202_1. URL: https://www.tandfonline.com/doi/abs/10.1207/s15327051hci0202_1.
- [136] Eyal Peer et al. "'Protect Me Tomorrow': Commitment Nudges to Remedy Compromised Passwords". In: *ACM Trans. Comput.-Hum. Interact.* 31.5 (Nov. 2024). ISSN: 1073-0516. DOI: 10.1145/3689038. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3689038>.
- [137] Charles Perrow. "Organizing to Reduce the Vulnerabilities of Complexity". In: *Journal of Contingencies and Crisis Management* 7 (Dec. 2002), pp. 150–155. DOI: 10.1111/1468-5973.00108.
- [138] Katherine Plant and Neville Stanton. "The process of processing: exploring the validity of Neisser's perceptual cycle model with accounts from critical decision-making in the cockpit". In: *Ergonomics* 58 (Dec. 2014), pp. 1–15. DOI: 10.1080/00140139.2014.991765.
- [139] *Industrial-Process Measurement and Control—Evaluation of System Properties for the Purpose of System Assessment, Part 5: Assessment of System Dependability, Draft*. Standard. Int'l Electro-technical Commission (IEC) Secretariat, Feb. 1992.
- [140] Ying Qin, Wanhui April Zhou, and Bu Zhong. "Blame games: Who is to blame when things go wrong, human or AI?" In: *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. CHI EA '25. Association for Computing Machinery, 2025. ISBN: 9798400713958. DOI: 10.1145/3706599.3719949. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3706599.3719949>.
- [141] Jens Rasmussen. "Approaches to the control of the effects of human error on chemical plant safety". In: *International Symposium on Preventing Major Chemical Accidents*. American Institute of Chemical Engineers. 1987.
- [142] Jens Rasmussen. "Human errors. A taxonomy for describing human malfunction in industrial installations". In: *IEEE Transactions on Systems, Man & Cybernetics* 4.2-4 (1982), pp. 311–333.
- [143] Jens Rasmussen. "Risk management in a dynamic society: a modelling problem". In: *Safety Science* 27.2 (1997), pp. 183–213. ISSN: 0925-7535. DOI: [https://doi.org/10.1016/S0925-7535\(97\)00052-0](https://doi.org/10.1016/S0925-7535(97)00052-0). URL: <https://www.sciencedirect.com/science/article/pii/S0925753597000520>.
- [144] Jens Rasmussen. "Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models". In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-13.3 (1983), pp. 257–266. DOI: 10.1109/TSMC.1983.6313160.
- [145] L.M. Rea and R.A. Parker. *Designing and Conducting Survey Research: A Comprehensive Guide*. EBL-Schweitzer. Wiley, 2014. ISBN: 9781118767030. URL: <https://books.google.fr/books?id=U8BBAAQBAJ>.
- [146] Gemma J. M. Read et al. "State of science: evolving perspectives on 'human error'". In: *Ergonomics* 64.9 (2021). PMID: 34243698, pp. 1091–1114. URL: <https://doi.org/10.1080/00140139.2021.1953615>.
- [147] J Reason et al. "TRIPOD, A Principled Basis for Safer Operations". In: *The Hague: Shell Internationale Petroleum Maatschappij* (1989).

- [148] James Reason. *A Life in Error - From Little Slips to Big Disasters*. 1. London: CRC Press, 2013, p. 150. DOI: <https://doi.org/10.1201/9781315263830>.
- [149] James Reason. *Human error*. 1. Cambridge university press, 1990.
- [150] James Reason. "Human error: models and management". In: *BMJ (Clinical research ed.)* 320,7237 (2000), pp. 768–70. DOI: doi : 10 . 1136/bmj . 320 . 7237 . 768.
- [151] James Reason. *Managing the Risks of Organizational Accidents*. Routledge, 1997. DOI: <https://doi.org/10.4324/9781315543543>.
- [152] James Reason. "Safety paradoxes and safety culture". In: *Injury Control and Safety Promotion* 7.1 (2000), pp. 3–14. DOI: 10 . 1076 / 1566 - 0974 (200003) 7 : 1 ; 1 - V ; FT003. eprint: [https://doi.org/10.1076/1566-0974\(200003\)7:1;1-V;FT003](https://doi.org/10.1076/1566-0974(200003)7:1;1-V;FT003). URL: [https://doi.org/10.1076/1566-0974\(200003\)7:1;1-V;FT003](https://doi.org/10.1076/1566-0974(200003)7:1;1-V;FT003).
- [153] James Reason. "The Identification of Latent Organizational Failures in Complex Systems". In: *Verification and Validation of Complex Systems: Human Factors Issues*. Ed. by John A. Wise, V. David Hopkin, and Paul Stager. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 223–237. ISBN: 978-3-662-02933-6.
- [154] JT Reason, J Carthey, and MR de Leval. "Diagnosing "vulnerable system syndrome": an essential prerequisite to effective risk management". In: *Quality in Health Care* 10, Supp 2 (2001), pp. ii21–5. DOI: <https://doi.org/10.1136/qhc.0100021>.
- [155] Jun Rekimoto. "Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments". In: *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology*. UIST '97. Banff, Alberta, Canada: Association for Computing Machinery, 1997, pp. 31–39. ISBN: 0897918819. DOI: 10.1145/263407.263505. URL: <https://doi.org/10.1145/263407.263505>.
- [156] Kirsten M.A. Revell et al. "Breaking the cycle of frustration: Applying Neisser's Perceptual Cycle Model to drivers of semi-autonomous vehicles". In: *Applied Ergonomics* 85 (2020), p. 103037. ISSN: 0003-6870. DOI: <https://doi.org/10.1016/j.apergo.2019.103037>. URL: <https://www.sciencedirect.com/science/article/pii/S0003687019302431>.
- [157] Grégoire Richard et al. "Multivibes: What if your vr controller had 10 times more vibrotactile actuators?" In: *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2023, pp. 703–712.
- [158] Gene I Rochlin. "Safe operation as a social construct". In: *Ergonomics* 42.11 (1999), pp. 1549–1560.
- [159] Quentin Roy, Futian Zhang, and Daniel Vogel. "Automation Accuracy Is Good, but High Controllability May Be Better". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–8. ISBN: 9781450359702. DOI: 10 . 1145 / 3290605 . 3300750. URL: <https://doi.org/10.1145/3290605.3300750>.
- [160] Nicolas Salzmann. *Méthodologie & analyse de la valeur – DI05 – Support d'enseignement*. Université de Technologie de Compiègne, 2023. URL: https://di05.uv.utc.fr/resources/Supports-de-cours/di05_poly_nov2023_v1.7.pdf.

- [161] Jeff Sauro and James R. Lewis. "Average Task Times in Usability Tests: What to Report?" In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '10. Atlanta, Georgia, USA: Association for Computing Machinery, 2010, pp. 2347–2350. ISBN: 9781605589299. DOI: 10.1145/1753326.1753679. URL: <https://doi.org/10.1145/1753326.1753679>.
- [162] Philippe Schmid et al. "Interaction Interferences: Implications of Last-Instant System State Changes". In: *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 516–528. ISBN: 9781450375146. URL: <https://doi.org/10.1145/3379337.3415883>.
- [163] *Sci-Tech Today*. *Google Maps Statistics By Region, Demographics And Facts* (2025). URL: <https://www.sci-tech-today.com/stats/google-maps-statistics-updated/>.
- [164] Abigail J. Sellen and Donald A. Norman. "The Psychology of Slips". In: *Experimental Slips and Human Error: Exploring the Architecture of Volition*. Ed. by Bernard J. Baars. Boston, MA: Springer US, 1992, pp. 317–339. ISBN: 978-1-4899-1164-3. DOI: 10.1007/978-1-4899-1164-3_13. URL: https://doi.org/10.1007/978-1-4899-1164-3_13.
- [165] Naveen Sendhilnathan et al. "A Multimodal Approach for Targeting Error Detection in Virtual Reality Using Implicit User Behavior". In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. CHI '25. Association for Computing Machinery, 2025. ISBN: 9798400713941. DOI: 10.1145/3706598.3713777. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3706598.3713777>.
- [166] Naveen Sendhilnathan et al. "Detecting Input Recognition Errors and User Errors using Gaze Dynamics in Virtual Reality". In: *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. UIST '22. Bend, OR, USA: Association for Computing Machinery, 2022. ISBN: 9781450393201. DOI: 10.1145/3526113.3545628. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3526113.3545628>.
- [167] Kelly G Shaver. "Defensive attribution: Effects of severity and relevance on the responsibility assigned for an accident." In: *Journal of personality and social psychology* 14.2 (1970), p. 101.
- [168] Danqing Shi et al. "Simulating Errors in Touchscreen Typing". In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. CHI '25. Association for Computing Machinery, 2025. ISBN: 9798400713941. DOI: 10.1145/3706598.3713153. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3706598.3713153>.
- [169] Shaishav Siddhpuria et al. "Pointing at a Distance with Everyday Smart Devices". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, pp. 1–11. ISBN: 9781450356206. DOI: 10.1145/3173574.3173747. URL: <https://doi.org/10.1145/3173574.3173747>.
- [170] Herbert A Simon. "Models of man; social and rational." In: (1957).
- [171] Herbert A Simon. "Rational choice and the structure of the environment." In: *Psychological review* 63.2 (1956).

- [172] Neville A. Stanton and Paul M. Salmon. "Human error taxonomies applied to driving: A generic driver error taxonomy and its implications for intelligent transport systems". In: *Safety Science* 47.2 (2009), pp. 227–237. ISSN: 0925-7535. DOI: <https://doi.org/10.1016/j.ssci.2008.03.006>. URL: <https://www.sciencedirect.com/science/article/pii/S0925753508000441>.
- [173] BlueTwelve Studio. *Stray*. Game [Windows]. Annapurna Interactive, California, United States. 2022.
- [174] Lucy Suchman. "Located accountabilities in technology production". In: *Scandinavian journal of information systems* 14.2 (2002), p. 7.
- [175] *Synonymo.fr*. <https://www.synonymo.fr/synonyme/> or its public archive <https://web.archive.org/web/2/http://synonymo.fr/>.
- [176] Susanne Tak and Andy Cockburn. "Window Watcher: A Visualisation Tool for Understanding Windowing Activities". In: *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*. OZCHI '09. Melbourne, Australia: Association for Computing Machinery, 2009, pp. 241–248. ISBN: 9781605588544. DOI: 10.1145/1738826.1738865. URL: <https://doi.org/10.1145/1738826.1738865>.
- [177] *The Free Dictionnary for "the emperor's new clothes"*. <https://idioms.thefreedictionary.com/the+emperor's+new+clothes>.
- [178] *Thesaurus*. <https://www.thesaurus.com/> or its public archive <https://web.archive.org/web/20250107064218/https://www.thesaurus.com/>.
- [179] Rajashekhar V S et al. "To Show or Hide - Investigating the Effect of Cursor Visibility in Gaze-controlled Interfaces". In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. CHI EA '24. Honolulu, HI, USA: Association for Computing Machinery, 2024. ISBN: 9798400703317. DOI: 10.1145/3613905.3651005. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3613905.3651005>.
- [180] Frederick Verbruggen et al. "A consensus guide to capturing the ability to inhibit actions and impulsive behaviors in the stop-signal task." In: *eLife* 8 (Apr. 2019). DOI: 10.7554/eLife.46323.
- [181] Marcello Vitali-Rosati. *Eloge du Bug – Être libre à l'époque du numérique*. La Découverte, May 2024. ISBN: 978235522238.
- [182] Simon Weckert. *Google Maps Hacks*. <https://simonweckert.com/googlemaphacks.html>.
- [183] Rick Wicklin. *The bias-corrected and accelerated (BCa) bootstrap interval*. URL: <https://blogs.sas.com/content/iml/2017/07/12/bootstrap-bca-interval.html>.
- [184] David Gray Widder and Claire Le Goues. "What is a "bug"? On subjectivity, epistemic power, and implications for software research". In: *ArXiv abs/2402.08165* (2024). URL: <https://api.semanticscholar.org/CorpusID:267637082>.
- [185] Douglas A. Wiegmann and Scott A. Shappell. "The human factors analysis and classification system (HFACS)". In: *A human error approach to aviation accident analysis: the human factors analysis and classification system*. Routledge, Dec. 2017, pp. 45–71. DOI: 10.4324/9781315263878-3.
- [186] *Wiktionary*. <https://fr.wiktionary.org/wiki/error> and <https://fr.wiktionary.org/wiki/erreur>.

- [187] Max L. Wilson. *Mental workload vs cognitive load vs everything else in HCI*. <https://medium.com/@cogpi/mental-workload-vs-cognitive-load-vs-everything-else-in-hci-575722d14572>.
- [188] Sarah Wiseman et al. "Using Checksums to Detect Number Entry Error". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. Paris, France: Association for Computing Machinery, 2013, pp. 2403–2406. ISBN: 9781450318990. DOI: 10.1145/2470654.2481332. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/2470654.2481332>.
- [189] Shota Yamanaka and Hiroki Usuba. "Tuning Endpoint-variability Parameters by Observed Error Rates to Obtain Better Prediction Accuracy of Pointing Misses". In: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. CHI '23. Hamburg, Germany: Association for Computing Machinery, 2023. ISBN: 9781450394215. DOI: 10.1145/3544548.3580746. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3544548.3580746>.
- [190] Mingyuan Zhong et al. "ScreenAudit: Detecting Screen Reader Accessibility Errors in Mobile Apps Using Large Language Models". In: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. CHI '25. Association for Computing Machinery, 2025. ISBN: 9798400713941. DOI: 10.1145/3706598.3713797. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3706598.3713797>.
- [191] Yixin Zou et al. "Encouraging Users to Change Breached Passwords Using the Protection Motivation Theory". In: *ACM Trans. Comput.-Hum. Interact.* 31.5 (Nov. 2024). ISSN: 1073-0516. DOI: 10.1145/3689432. URL: <https://doi-org.ressources-electroniques.univ-lille.fr/10.1145/3689432>.