



HAL
open science

Adaptability of software tutorials to user interfaces

Raphaël Perraud

► **To cite this version:**

Raphaël Perraud. Adaptability of software tutorials to user interfaces. Actes joints de la Conférence Internationale Francophone sur l'Interaction Humain-Machine (IHM'24) de l'AFIHM, Rencontres Doctorales, 2024, pp.1-7. hal-04490515

HAL Id: hal-04490515

<https://hal.science/hal-04490515v1>

Submitted on 5 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adaptability of software tutorials to user interfaces

Adaptation des tutoriels pour logiciels aux interfaces utilisateurs

RAPHAËL PERRAUD, INRIA - Centre de l'Université de Lille, France

Video tutorials have become one the main media to learn software literacy. However, customizable user interfaces introduces challenges for learners, as differences between the demonstrated interface in tutorials and the learner's interface may impede effective learning. These differences are exacerbated by software updates that alter features, commands, and visual representations between interfaces. This leads to discrepancies between tutorial demonstrations and learners' interfaces, that impede effective learning and causing obsolescence. This thesis addresses this issue by introducing "interface distance" and explores its impact on user learnability. To face these challenges, this research will develop the concept of adaptable tutorials to minimize these interface differences and cover the main cases of tutorial depreciation.

Les tutoriels vidéo sont devenus l'un des principaux supports d'apprentissage des logiciels. Cependant, les interfaces utilisateur personnalisables présentent des défis pour les apprenants, car les différences entre l'interface démontrée dans les didacticiels et l'interface de l'apprenant peuvent entraver un apprentissage efficace. Ces différences sont exacerbées par des mises à jour logicielles continues qui modifient les fonctionnalités et les représentations visuelles des commandes entre les interfaces. Cela conduit à des écarts entre l'interface démontrée dans un tutoriel et l'interface d'un apprenant, qui entravent un apprentissage efficace et provoquent l'obsolescence des tutoriels vidéos. Cette thèse aborde ce problème en introduisant la notion de « distance d'interface » et explore son impact sur l'apprentissage de l'utilisateur. Pour faire face à ces défis, cette recherche développera le concept de tutoriels adaptables pour minimiser ces différences d'interface et couvrir les principaux cas de dépréciation des tutoriels.

CCS Concepts: • **Human-centered computing** → **Human computer interaction (HCI)**; **Empirical studies in interaction design**; **Interaction design process and methods**.

Mots Clés et Phrases Supplémentaires: adaptability, transfer learning, software learning, empirical studies

Mots Clés et Phrases Supplémentaires: Adaptabilité, transfert d'apprentissages, apprentissage des logiciels, études empiriques

Reference:

Raphaël Perraud. 2024. Adaptability of software tutorials to user interfaces. *IHM '24 : Rencontres Doctorales de la 35^e conférence Francophone sur l'Interaction Humain-Machine, March 25–29, 2024, Paris, France*

1 CONTEXT

A plethora of tutorials are now available in various formats (text, video, mixed) to broadcast knowledge about digital skills and software literacy. Video tutorials have become one of the main media to learn new skills [26]. The growth of video-based platforms in our daily use also transformed the way we learn new skills, and social media platforms such as Youtube or Tiktok became major players in the dissemination of knowledge [2, 5, 8, 27]. Videos are now the main support resource used by new users to develop software skills [18]. Video tutorials are favored among learners because they can include different ways of resolving a challenge [24] and help newcomers to achieve a goal by mimicking the process demonstrated in the tutorial.

However, user interfaces are becoming increasingly modular and customizable, and many differences between the learner's interface and the one demonstrated in the tutorial may arise, limiting the user's ability to follow. Differences can also stem from tools exposing different interface components and application states than the ones shown in the tutorial [39]. The on-going evolution of software interfaces also brings its own share of complexity, bringing changes at different

levels of the interface and the application. For instance, a software update can depreciate certain features in favor of new ones, change the user action used to activate the command or its visual representation. Thus, video tutorials suffer from one major limitation which is that they are produced at a given instant, with a specific configuration/customization, with specific plugins and language. Therefore, virtually, all video tutorials are doomed to become outdated because they are not updated, even though the software they demonstrate or provide instructions for has kept evolving. As a result, users often have to worry about matching their version of the software with the one of the tutorial, because significant interface changes can occur between two versions [20].

In the end, many tutorial viewers work with different interfaces than the ones used for producing a video tutorial [39], and need to undergo an additional *user interface translation* step [37] in order to compensate for the differences that may exist between the two interfaces [21]. The most probable solution to overcome this issue currently consists in looking for another video tutorial produced with the correct software version, which takes significant time and might result in finding a video with significant irrelevant portions [18]. As users often tackle learning software by a task-oriented approach [14, 20, 22, 38], it becomes cumbersome to find a suitable tutorial due to the abundance of online tutorials and must interpret, if in addition they must adapt the content to suit to their goals [16] and make sure the version of the software used in the video and the configuration of the GUI correspond. Consequently, the objectives of this thesis will be thus to study the concept of *interface distance* and its impact on user's capabilities to reproduce a tutorial and her learnability.

Continuous interface updates and their customization ability lead users to face a multitude of different interfaces. The search for functionalities in these various interfaces can thus hinder their learning by generating friction that is not due to the difficulty of the tutorial itself, but rather to the availability of the interface to the user's needs. As various factors can cause minor or major depreciation of a part or the totality of a tutorial, it would be appropriate to enable the capacity of tutorials to adapt their content in order to reflect a meaningful interface for the learner. While many tools have been proposed to leverage software authoring and watching by creating new content, the question of the sustainability of the tutorials remains largely unexplored. Various solutions have been explored to overcome this problem, ranging from automatically detecting versioning issues [33] to relying on community-based sourcing to refine missing elements or leverage viewer's awareness following changes that need to be edited to old tutorials [19, 24, 41]. This thesis focuses on solving the problem of *interface distance* and *tutorial depreciation* with adaptable tutorials : tutorials whose content can be adapted to the learner's situation, considering her software knowledge, habits, materials and user interface state.

2 RELATED WORK

Video tutorials have inspired a large corpus of research in HCI, from the authors to the viewer's perspective. A first line of research can be found in the various tools that research proposed to assist tutorial authors and reduce the burden of producing their content. The approach of generating tutorials from user demonstration has become more important in research [4, 12, 24, 25] alongside the prevalence of interactive video tutorials as preferred medium for tutorials [18, 44]. While these solutions facilitate the production of tutorials, it cannot solve the issues of *interface distance* and *tutorial depreciation* since it would require tutorial authors to produce a tutorial for each interface configuration, which seems complicated with interfaces that can be personalized.

A second line of research aimed to facilitate the capabilities of viewers (or *learners*) to follow tutorials by integrating them directly into the user interface (which requires special authoring) [13, 28, 42]. Other tools offer to automate the analysis of tutorials to extract structuring information about the tutorial for the learner to help to select the relevant

tutorial [3, 29, 34, 35] and the sections of interest within a tutorial [17, 40, 44, 45]. However, if these tools offer to leverage tutorials by enabling new interactive properties, they do not have the capacity to update the video content itself, and the tutorial is still threatened by obsolescence. The concept of adaptive tutorials aims to reduce the burden of maintaining content across time, meaning and relevancy through interfaces and similar software and learner's context. These tutorials should thus be able to compute an *interface translation* to map similar features and workflows through updates, layout and shortcuts customizations, materials capabilities to preserve the relevance of the tutorial through regardless of this context.

In addressing the challenge of reconciling interface differences between distinct software interfaces, various projects aimed at mitigating potential frictions with parallel approaches. While *Where is that feature?* [1] delves into the problem of localization of features of a same application across different devices, and presents design strategies to guide users to find them from an interface to another. *Show-me-how* focuses on the transfer learning between similar software offering to perform the same means but each with a different interface [37]. Raissi et al. have extended the study of transfer learning by observing the influence of retroactive transfer when user alternate between different interfaces on user efficiency [36], observing that even small changes can produce significant drops in performance. *Blocks-to-CAD* uses this transfer learning principle to propose a cross-application bridge concept for learning new software [23] by gradually changing a familiar application into another one which uses similar interaction paradigms. Another approach would be to leverage users' awareness of available feature in the interface [9] or their vocabulary [7, 10] in order to be able to overgo by themselves the interface difference. However, very little has been done to understand the difficulties involved in differentiating between two interfaces. We argue that updates and customization can issue depreciation [33, 43] or lack of relevance of these due to the gap between the interfaces that can be fixed by enabling the capability of tutorial of being adaptable to user's context.

3 PHD PROGRESS

3.1 Main objectives

The objective of this research is to establish a series of works on interfaces differences in learning situations with video tutorials. To do so, the milestones of this work are to understand high-level issues with video tutorials, define interface differences and hierarchize them according to their causes and effects on learners' use of the tutorial, propose solutions to cover the most problematic cases of interface differences.

3.2 Progress

3.2.1 High-level issues with tutorials. A review of the literature allowed us to observe that video tutorials have already largely inspired research in HCI but very little, however, focuses on improving the maintainability of tutorial content over time and user interface customizations. Preliminary interviews allowed us to identify high-level issues which will define the roadmap for this thesis work. These interviews allowed us to establish limitations related to video tutorials, their obsolescence and current practices to circumvent this problem. From the analysis of these semi-structured interviews, we have identified six challenges that will define the roadmap for this research work :

- (1) *Workload of maintenance.* Even with strategies to intervene only on minimal portions when necessary, the format of the video tutorials makes their maintenance cumbersome.
- (2) *Customization.* Authors receive a lot of help requests from their communities for personalized cases which increase the support burden due to specific incompatibilities.

- (3) *Novice-oriented tutorial authoring*. Authors limit their usual setup to suit the likely conditions of use by novice users.
- (4) *Keeping the viewer aware*. Interface updates can change the spatial organization and visual representation of feature and cause difficulties in following a tutorial.
- (5) *Quality over compatibility*. Users tend to trust particular video authors because they know they produce quality content even if the tutorial content is getting outdated.

The main alterations that an interface can undergo concern the location of features within the interface, defined by their *Reachability* and its *Availability*. Changes affecting the representation of the command activating the feature, such as updates of its visual iconography, are considered infrequent. Consequently, our focus centers on modifications related to the *Reachability* and *Availability* within the GUI. In this aim, we deliberately omit considerations of *Reachability* modifications attributable to differences in input modalities such as shortcuts. Our emphasis remains directed towards examining frictions stemming from alterations that impact the visual interface, regardless of if it was initiated by the learner (customized interface) or resulted from a software update. We therefore defined the concepts of *Reachability* and *Availability* as follows :

Reachability. The *Reachability* of a feature is dependent of its application hierarchy and its interface hierarchy. This first refers to the containerization of commands into semantic entities (e.g. the command relative to align an object to the left border is contained into an "Alignment" parent, itself contained by "Text"). The interface hierarchy refers then to the relative disposal, size, positioning and layer level of the *Representation* of the command on the interface. The *Reachability* of a command is dependant of the input modalities allowed to invoke to the same feature through another way (such as shortcuts).

Availability. The *Availability* of a command in a software define if an enabled instance of the command exists in the current state of the application. The variance of the *Availability* of a feature will be mostly due to application versioning or third-party add-ons.

3.2.2 *Quantifying frictions related to Reachability and Availability issues.* We therefore chose to focus our first main study on the frictions resulting from interface differences between a tutorial interface and a learner interface related to the variations of the *Reachability* and the *Availability* of commands.

We have characterized the differences in *Reachability* according to several levels of magnitude of spatial displacement between the interfaces. We have therefore defined three categories of *Reachability* differences. Features DIRECTLY REACHABLE ON SCREEN, whose parent component is visible on both interfaces (within graphic software, a "Brush" tool in the toolbar, for example). Features that are not NOT DIRECTLY REACHABLE ON SCREEN, whose parent component is present on the screen but not active (e.g. the "TextBold" feature in the "Text" component, activated when the tab is clicked to reveal the tab). These two types of *Reachability* differences can be broken down into different levels of spatial distance between the localizations of the same command between two interfaces. Finally, features REQUIRING WORKSPACE CONFIGURATION TO BE REACHABLE, notably by activating the parent window from the "Windows" application menu.

Then there are the cases of interface distance due to the modification of the *Availability* of a feature. Here we find two levels : when the feature is but available in the application (think of features that depend on selection in the workspace, or hardware configuration), and when the feature is simply NOT AVAILABLE (due to the addition of a plugin or a version difference, for example).

An initial study was carried out to assess the frictions imposed by these levels of interface difference, and to study viewer behavior in locating, revealing, activating and applying a feature that is not presented identically between interfaces. Our results suggests that the alterations of the *Availability* of a feature and modifications REQUIRING WORKSPACE CONFIGURATION are unsurprisingly the differences that create the most frictions, leading to confusing attempts to reveal the feature on the interface, or activate it in order to apply its effect. A number of users are unaware of the configurable capacity of certain applications, and only focus on a localized region close to the position of the feature in the tutorial shown on their interface when looking for it. Locating a feature seems to be contextual, with participants paying more visual attention to components with semantics close to the feature they are looking for (e.g. a "Text styles" window when locating the "Paragraph" window). Floating interface elements are not considered as salient visual elements, contrary to what we had hypothesized. They are rarely, if ever, taken into account in visual searches, and several participants reported having difficulty recognizing them as interface elements, since they were not integrated into the structure of a panel, or an element grouping together several other similar interface components. Interface alterations where feature are DIRECTLY REACHABLE seems to make it easier for users to locate them, but spatial proximity has no significant impact on their ability to do so. The study has been submitted at DIS'24 in early February.

3.2.3 Minimize interface differences between tutorials and learner's interface. The previous study helped us to better understand how differences in terms of *Reachability* and *Availability* impact user experience with video tutorials. In the following steps, we will investigate the notion of interface distance to be able to measure the distance between two interfaces from the user's point of view, at several levels : application concepts, operating procedures, interaction styles, input capabilities, spatial organization and visual representation of features [15]. We will propose then a systemic analysis of the envisioned solutions of adaptable software tutorials, by implementing new systems supporting the diffusion of adaptive tutorials for different devices and contexts of use.

A promising approach would be compilable tutorials that could be run on the learner's machine as an executable, process context informations before generating the tutorial. To do this, we would rely on the work already done on capturing, recording and analysing the interaction history [11, 13, 30–32]. An interaction description-approach [6] allows to generate a tutorial fitted for the learner's context and envision that it would be capable to process an *interface translation* for differences due to spatial reorganization, feature depreciation, language and user preference settings, graphics update.

A more global reflection will be conducted on the levers offered by the adaptability of video tutorials for improving the learnability of the interface as well as promoting serendipitous discoveries of its features. We will then explore the cost of different adaptations on the interface comparison process, the retention of its learning as well as the possibilities of learning new workflows within the same application.

RÉFÉRENCES

- [1] Jessalyn Alvina, Andrea Bunt, Parmit K. Chilana, Sylvain Malacria, and Joanna McGrenere. 2020. Where Is That Feature? : Designing for Cross-Device Software Learnability. In *Proceedings of the 2020 ACM Designing Interactive Systems Conference*. ACM, Eindhoven Netherlands, 1103–1115. <https://doi.org/10.1145/3357236.3395506>
- [2] Sara Atske. 2018. Many Turn to YouTube for Children's Content, News, How-To Lessons.
- [3] Nikola Banovic, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2012. Waken : Reverse Engineering Usage Information and Interface Structure from Software Videos. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. ACM, Cambridge Massachusetts USA, 83–92. <https://doi.org/10.1145/2380116.2380129>
- [4] Lingfeng Bao, Zhenchang Xing, Xin Xia, and David Lo. 2019. VT-Revolution : Interactive Programming Video Tutorial Authoring and Watching System. *IEEE Transactions on Software Engineering* 45, 8 (Aug. 2019), 823–838. <https://doi.org/10.1109/TSE.2018.2802916>

- [5] Cynthia J. Brame. 2016. Effective Educational Videos : Principles and Guidelines for Maximizing Student Learning from Video Content. *CBE—Life Sciences Education* 15, 4 (Dec. 2016), es6. <https://doi.org/10.1187/cbe.16-03-0125>
- [6] Peggy Chi, Nathan Frey, Katrina Panovich, and Irfan Essa. 2021. Automatic Instructional Video Creation from a Markdown-Formatted Tutorial. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 677–690. <https://doi.org/10.1145/3472749.3474778>
- [7] Parmit K. Chilana, Amy J. Ko, and Jacob O. Wobbrock. 2012. LemonAid : Selection-Based Crowdsourced Contextual Help for Web Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Austin Texas USA, 1549–1558. <https://doi.org/10.1145/2207676.2208620>
- [8] Pedro Cuesta-Valiño, Pablo Gutiérrez-Rodríguez, and Patricia Durán-Álamo. 2022. Why Do People Return to Video Platforms? Millennials and Centennials on TikTok. *Media and Communication* 10, 1 (Feb. 2022), 198–207.
- [9] Leah Findlater and Joanna McGrenere. 2010. Beyond Performance : Feature Awareness in Personalized Interfaces. *International Journal of Human-Computer Studies* 68, 3 (March 2010), 121–137. <https://doi.org/10.1016/j.ijhcs.2009.10.002>
- [10] Adam Fourney, Ben Lafreniere, Richard Mann, and Michael Terry. 2012. "Then Click Ok!" : Extracting References to Interface Elements in Online Documentation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Austin Texas USA, 35–38. <https://doi.org/10.1145/2207676.2207682>
- [11] Christian Frisson, Sylvain Malacria, Gilles Bailly, and Thierry Dutoit. 2016. InspectorWidget : A System to Analyze Users Behaviors in Their Applications. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, San Jose California USA, 1548–1554. <https://doi.org/10.1145/2851581.2892388>
- [12] Floraine Grabler, Maneesh Agrawala, Wilmot Li, Mira Dontcheva, and Takeo Igarashi. 2009. Generating Photo Manipulation Tutorials by Demonstration. In *ACM SIGGRAPH 2009 Papers on - SIGGRAPH '09*. ACM Press, New Orleans, Louisiana, 1. <https://doi.org/10.1145/1576246.1531372>
- [13] Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2010. Chronicle : Capture, Exploration, and Playback of Document Workflow Histories. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*. ACM, New York New York USA, 143–152. <https://doi.org/10.1145/1866029.1866054>
- [14] Nathaniel Hudson, Benjamin Lafreniere, Parmit K. Chilana, and Tovi Grossman. 2018. Investigating How Online Help and Learning Resources Support Children's Use of 3D Design Software. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, Montreal QC Canada, 1–14. <https://doi.org/10.1145/3173574.3173831>
- [15] Robin H. Kay. 2007. Learning Performance and Computer Software : An Exploration of Knowledge Transfer. *Computers in Human Behavior* 23, 1 (Jan. 2007), 333–352. <https://doi.org/10.1016/j.chb.2004.10.029>
- [16] Caitlin Kelleher and Randy Pausch. 2005. Stencils-Based Tutorials : Design and Evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Portland Oregon USA, 541–550. <https://doi.org/10.1145/1054972.1055047>
- [17] Kandarp Khandwala and Philip J. Guo. 2018. Codemotion : Expanding the Design Space of Learner Interactions with Computer Programming Tutorial Videos. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*. ACM, London United Kingdom, 1–10. <https://doi.org/10.1145/3231644.3231652>
- [18] Kimia Kiani, George Cui, Andrea Bunt, Joanna McGrenere, and Parmit K. Chilana. 2019. Beyond "One-Size-Fits-All" : Understanding the Diversity in How Software Newcomers Discover and Make Use of Help Resources. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–14. <https://doi.org/10.1145/3290605.3300570>
- [19] Juho Kim. 2015. *Learnersourcing : Improving Learning with Collective Learner Activity*. Thesis. Massachusetts Institute of Technology.
- [20] Ben Lafreniere, Andrea Bunt, Matthew Lount, and Michael Terry. 2013. Understanding the Roles and Uses of Web Tutorials. *Proceedings of the International AAAI Conference on Web and Social Media* 7, 1 (2013), 303–310. <https://doi.org/10.1609/icwsm.v7i1.14413>
- [21] Ben Lafreniere, Andrea Bunt, Matthew Lount, Michael Terry, and Donald Cowan. 2012. *Looks Cool, I'll Try This Later!* : Understanding the Faces and Uses of Online Tutorials. Technical Report 2012-01. University of Waterloo, Waterloo, CA.
- [22] Benjamin Lafreniere, Andrea Bunt, and Michael Terry. 2014. Task-Centric Interfaces for Feature-Rich Software. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures : The Future of Design*. ACM, Sydney New South Wales Australia, 49–58. <https://doi.org/10.1145/2686612.2686620>
- [23] Ben Lafreniere and Tovi Grossman. 2018. Blocks-to-CAD : A Cross-Application Bridge from Minecraft to 3D Modeling. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. ACM, Berlin Germany, 637–648. <https://doi.org/10.1145/3242587.3242602>
- [24] Benjamin Lafreniere, Tovi Grossman, and George Fitzmaurice. 2013. Community Enhanced Tutorials : Improving Tutorials with Multiple Demonstrations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Paris France, 1779–1788. <https://doi.org/10.1145/2470654.2466235>
- [25] Ben Lafreniere, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2014. Investigating the Feasibility of Extracting Tool Demonstrations from In-Situ Video Content. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Toronto Ontario Canada, 4007–4016. <https://doi.org/10.1145/2556288.2557142>
- [26] Patricia G. Lange. 2018. Informal Learning on YouTube. In *The International Encyclopedia of Media Literacy* (1 ed.), Renee Hobbs and Paul Mihailidis (Eds.). Wiley, 1–11. <https://doi.org/10.1002/9781118978238.ieml0090>
- [27] Gao-fu Liu, Peng-chao Gao, Yu-chun Li, and Zhuo-ping Zhang. 2019. Research on the Influence of Social Media Short Video Marketing on Consumer Brand Attitude. In *Proceedings of the 2019 5th International Conference on Social Science and Higher Education (ICSSHE 2019)*. Atlantis Press, Xiamen,

- China. <https://doi.org/10.2991/icsshe-19.2019.192>
- [28] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2013. Patina : Dynamic Heatmaps for Visualizing Application Usage. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Paris France, 3227–3236. <https://doi.org/10.1145/2470654.2466442>
- [29] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2014. Video Lens : Rapid Playback and Exploration of Large Video Collections and Associated Metadata. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. ACM, Honolulu Hawaii USA, 541–550. <https://doi.org/10.1145/2642918.2647366>
- [30] Alok Mysore and Philip J. Guo. 2017. Torta : Generating Mixed-Media GUI and Command-Line App Tutorials Using Operating-System-Wide Activity Tracing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, Québec City QC Canada, 703–714. <https://doi.org/10.1145/3126594.3126628>
- [31] Toshio Nakamura and Takeo Igarashi. 2008. An Application-Independent System for Visualizing User Operation History. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology - UIST '08*. ACM Press, Monterey, CA, USA, 23. <https://doi.org/10.1145/1449715.1449721>
- [32] Mathieu Nancel and Andy Cockburn. 2014. Causality : A Conceptual Model of Interaction History. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Toronto Ontario Canada, 1777–1786. <https://doi.org/10.1145/2556288.2556990>
- [33] Manziba Akanda Nishi and Kostadin Damevski. 2020. Automatically Identifying Valid API Versions for Software Development Tutorials on the Web. *Journal of Software : Evolution and Process* 32, 4 (April 2020). <https://doi.org/10.1002/smr.2227>
- [34] Amy Pavel, Colorado Reed, Björn Hartmann, and Maneesh Agrawala. 2014. Video Digests : A Browsable, Skimmable Format for Informational Lecture Videos. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. ACM, Honolulu Hawaii USA, 573–582. <https://doi.org/10.1145/2642918.2647400>
- [35] Luca Ponzanelli, Gabriele Bavota, Andrea Mocchi, Massimiliano Di Penta, Rocco Oliveto, Mir Hasan, Barbara Russo, Sonia Haiduc, and Michele Lanza. 2016. Too Long ; Didn't Watch! : Extracting Relevant Fragments from Software Development Video Tutorials. In *Proceedings of the 38th International Conference on Software Engineering*. ACM, Austin Texas, 261–272. <https://doi.org/10.1145/2884781.2884824>
- [36] Reyhaneh Raissi, Evanthia Dimara, Jacquelyn H. Berry, Wayne D. Gray, and Gilles Bailly. 2020. Retroactive Transfer Phenomena in Alternating User Interfaces. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–14. <https://doi.org/10.1145/3313831.3376538>
- [37] Vidya Ramesh, Charlie Hsu, Maneesh Agrawala, and Björn Hartmann. 2011. ShowMeHow : Translating User Interface Instructions between Applications. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. ACM, Santa Barbara California USA, 127–134. <https://doi.org/10.1145/2047196.2047212>
- [38] John Rieman. 1996. A Field Study of Exploratory Learning Strategies. *ACM Transactions on Computer-Human Interaction* 3, 3 (Sept. 1996), 189–218. <https://doi.org/10.1145/234526.234527>
- [39] Cristen Torrey, Elizabeth F. Churchill, and David W. McDonald. 2009. Learning How : The Search for Craft Knowledge on the Internet. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Boston MA USA, 1371–1380. <https://doi.org/10.1145/1518701.1518908>
- [40] Gaurav Verma, Trikey Nalamada, Keerti Harpavat, Pranav Goel, Aman Mishra, and Balaji Vasan Srinivasan. 2021. Non-Linear Consumption of Videos Using a Sequence of Personalized Multimodal Fragments. In *26th International Conference on Intelligent User Interfaces*. ACM, College Station TX USA, 249–259. <https://doi.org/10.1145/3397481.3450672>
- [41] Laton Vermette, Shruti Dembla, April Y. Wang, Joanna McGrenere, and Parmit K. Chilana. 2017. Social CheatSheet : An Interactive Community-Curated Information Overlay for Web Applications. *Proceedings of the ACM on Human-Computer Interaction* 1, CSCW (Dec. 2017), 1–19. <https://doi.org/10.1145/3134737>
- [42] Cheng-Yao Wang, Wei-Chen Chu, Hou-Ren Chen, Chun-Yen Hsu, and Mike Y. Chen. 2014. EverTutor : Automatically Creating Interactive Guided Tutorials on Smartphones by User Demonstration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Toronto Ontario Canada, 4027–4036. <https://doi.org/10.1145/2556288.2557407>
- [43] Di Wu, Xiao-Yuan Jing, Hongyu Zhang, Yuming Zhou, and Baowen Xu. 2021. Leveraging Stack Overflow to Detect Relevant Tutorial Fragments of APIs. In *2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, Honolulu, HI, USA, 119–130. <https://doi.org/10.1109/SANER50967.2021.00020>
- [44] Saelyne Yang, Jisu Yim, Aitolkyn Baigutanova, Seoyoung Kim, Minsuk Chang, and Juho Kim. 2022. SoftVideo : Improving the Learning Experience of Software Tutorial Videos with Collective Interaction Data. In *27th International Conference on Intelligent User Interfaces*. ACM, Helsinki Finland, 646–660. <https://doi.org/10.1145/3490099.3511106>
- [45] Dehai Zhao, Zhenchang Xing, Chunyang Chen, Xin Xia, and Guoqiang Li. 2019. ActionNet : Vision-Based Workflow Action Recognition From Programming Screencasts. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, Montreal, QC, Canada, 350–361. <https://doi.org/10.1109/ICSE.2019.00049>