

The Right Content at the Right Time: Contextual Examples for Just-in-time Creative Learning

C. Ailie Fraser

Design Lab, UC San Diego

La Jolla, CA 92093

cafraser@cs.ucsd.edu

ABSTRACT

People often run into barriers when doing creative tasks with software because it is difficult to translate goals into concrete actions. While expert-made tutorials, examples, and documentation abound online, finding the most relevant content and adapting it to one's own situation and task is a challenge. My research introduces techniques for exposing relevant examples to novices in the context of their own workflows. These techniques are embodied in three systems. The first, RePlay, helps people **find solutions when stuck** by automatically locating relevant moments from expert-made videos. The second, DiscoverySpace, helps novices **get started** by mining and recommending expert-made software macros. The third, CritiqueKit, helps novices **improve their work** by providing ambient guidance and recommendations. Preliminary experiments with RePlay suggest that contextual video clips help people complete targeted tasks. Controlled experiments with DiscoverySpace and CritiqueKit demonstrate that software macros prevent novices from losing confidence, and ambient guidance improves novice output. My research illustrates the power of user communities to support creative learning.

Author Keywords

creativity; contextual assistance; recommendations

LEARNING WITH CREATIVE SOFTWARE

People love to be creative. Software gives people the ability to produce incredibly creative work, such as illustrations, movies, and visualizations, which can even lead to world-changing breakthroughs (e.g., genomic research using creative visualizations to discover new biological pathways [9]). However, with power comes complexity. To allow flexibility and control, creative tools are often “bloated” with features [6]. People must use multiple tools or commands correctly in order to achieve their goals, and the gulf of execution [3] between these tools and the user's goals is often

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

UIST '18 Adjunct, October 14–17, 2018, Berlin, Germany

© 2018 Copyright is held by the owner/author(s).

ACM ISBN 978-1-4503-5949-8/18/10.

<https://doi.org/10.1145/3266037.3266127>

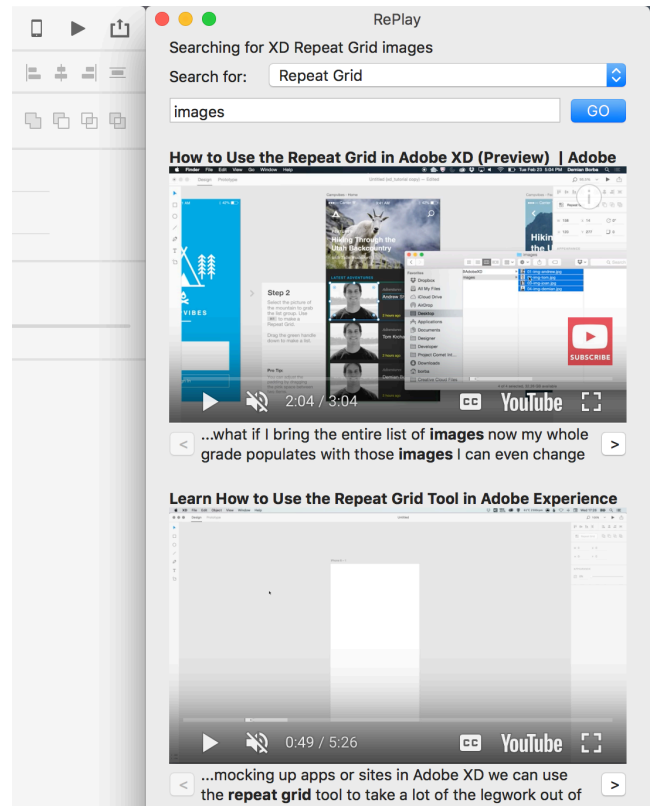


Figure 1: The RePlay MacOS panel gathers context about the user's actions in any accessibility-enabled application (in this example, Adobe XD), and uses it to augment search queries for more relevant results. It cues videos to relevant moments for quick, in-the-moment support.

prohibitively large. People spend years practicing and taking formal training to master creative software like Adobe Photoshop or Autodesk Maya. However, the growing prevalence of free and subscription-based software means that casual users now have access to tools that were once only used by professionals. Anyone can make their own animations or illustrations... if they can figure out how.

While some applications help onboard new users by integrating tutorials or helpful tips directly into the interface, these tend to be restricted to only basic introductory help. Consequently, people seek help online from the plethora of existing tutorials, documentation, and discussion fora. While this information is abundant, and most questions have been answered already somewhere, it can be challeng-

ing to *find* the right answers among this breadth of information. People must know exactly what they’re looking for, or spend time looking at potentially irrelevant information. In addition, information found through the web browser is out of the application’s context, requiring the user to figure out how to map it back to their own interface and task.

My research introduces methods for *leveraging existing expert examples and presenting them in-context*, with the goal of enabling people to produce better work with less frustration. I focus on the domain of creative software as it embodies the challenges described particularly strongly, however the techniques I present can extend to other domains involving complex tasks and expert examples. By leveraging existing work done by other users, my work enables people to learn from each other rather than relying solely on software documentation. My main technical contributions are back-end algorithms to curate and rank examples based on user context, and front-end interfaces that monitor user context and display interactive results.

My dissertation proposes three systems that support creative work at different stages of a task. The first, RePlay, helps people **find solutions when stuck** by automatically locating relevant moments from expert-made videos. The second, DiscoverySpace, helps novices **get started** by mining and recommending expert-made software macros. The third, CritiqueKit, helps novices **improve their work** by providing ambient guidance and recommendations. Preliminary experiments with RePlay suggest that contextual video clips help people complete targeted tasks. Controlled experiments with DiscoverySpace and CritiqueKit demonstrate that software macros prevent novices from losing confidence, and ambient guidance improves novice output when it adapts to their context.

CONTEXTUAL VIDEO CLIPS FOR TARGETED SUPPORT

One popular resource for learning and solving problems in software is online videos [8]. The visual nature of many creative tasks and their associated software interfaces makes videos especially useful, as they show clear demonstrations that might be harder to explain or understand in only text. However, videos are difficult to browse, search, and interact with. While text-based webpages are easy to skim through and search for specific words, most video platforms are limited to thumbnail and description views of search results, and skimming across the timeline for viewing. In addition, videos, like online tutorials or discussion fora, are usually viewed in the browser, out of the user’s context. This requires the user to articulate their problem and context appropriately to find a relevant result, and then switch back and forth between applications when watching the selected video.

My latest and ongoing work seeks to make videos more available to the user *in the context* of their current task. While some systems do exist for contextual video assistance [2, 5], these are specific to one application and use a preselected collection of videos. As software tasks become

more complex, they often involve cross-application workflows. I present a more general approach that makes use of the existing and growing set of videos online, and is application-agnostic, making no assumptions about the visual properties or content of videos.

I embody this approach in RePlay (Figure 1): a system that automatically extracts relevant clips from software videos based on the user’s context, and makes them available to the user while they work. RePlay is implemented as a MacOS panel in Swift, and it uses the MacOS Accessibility API to monitor the tools or commands used in any accessibility-enabled software. It uses this contextual information to augment the user’s search queries before sending them to YouTube. Video results are automatically re-ranked based on their similarity to the user’s context, and clips are selected from within videos that mention the user’s query terms. This search is conducted in real-time and makes use of publicly available data from YouTube (video metadata and captions) for ranking and clip selection.

Preliminary experiments conducted in the wild suggest that RePlay’s video clip search is useful for targeted tasks or specific questions. For users who instead want general information on a topic or do not yet have a goal in mind, watching full demonstrations or tutorials in the browser is likely more beneficial than short contextual clips. But when the user gets stuck on a particular question or problem while working, RePlay helps them to quickly find just that one piece of information they need to move forward. Several participants so far have encountered clips from within prohibitively long videos that gave them exactly the information and demonstration they were looking for.

We are currently iterating on RePlay’s design and search algorithm. The current method of literal keyword search finds most mentions of relevant tools (~68% in a sample dataset we annotated), but there is much room for improvement in both overall precision and ranking of these tool mentions. I am currently building a more robust semantic search algorithm to find more relevant moments. Initial studies have also suggested that the display and interaction methods for these videos is key to their utility; I am developing methods for displaying more informative thumbnails and enabling easier browsing within videos. Following these improvements, I plan to conduct an in-lab experiment to assess RePlay’s efficacy toward completing a specific task, as well as a longitudinal deployment in the field to assess RePlay’s ability to support real-world workflows across a variety of applications.

RePlay currently focuses on locating moments of procedural knowledge, but demonstration videos often also impart *conceptual* and *creative* knowledge. Learning technical software is only one piece of producing creative work; people must also learn how to generate ideas, make creative decisions in the moment, and find inspiration. Different techniques are likely to be necessary for finding and pre-

senting these types of knowledge, and I plan to explore these in the future.

EXPERT ACTIONS HELP NOVICES GET STARTED

While contextual video clips may help those who are already working on a task, what about new users who are just getting started? It can be difficult to get past a blank canvas, especially when one is not familiar with the software. Most users are typically only aware of a small percentage of software features [6]; they may not even know what results are possible. Even when users do have a goal in mind, figuring out which of the many low-level commands to use and how to combine them can hinder progress or even lead people to give up completely. Online tutorials can help but still must be interpreted and adapted to suit the user's own situation and goals.

To reduce the execution gap between users' high-level goals and the low-level steps needed to achieve them, we developed DiscoverySpace (Figure 2), a recommendation interface for action macros created by the user community [1]. Action macros encapsulate a sequence of operations that is executed as a batch. The DiscoverySpace prototype is a Photoshop panel that recommends action macros based

on features of the user's photo. These actions can be applied in one click for quick and easy exploration. Each action was downloaded from a selection of popular sharing websites, and descriptive keywords were generated based on its title and description. DiscoverySpace's recommendation algorithm prioritizes actions with keywords that map to features of the user's photo (e.g., "smooth skin" for images containing faces) based on a mapping we defined. A between-subjects study where participants of mixed experience levels edited their own photos in Photoshop found that DiscoverySpace's action suggestions significantly prevented novices from losing confidence and from getting stuck, when compared to Photoshop without DiscoverySpace.

DiscoverySpace demonstrated how leveraging the work of other users can help novices experiment with potential results, and abstracting away individual operations in favour of goal-driven actions can lower the barriers to getting started in new software. This work can extend to any software where command sequences can be saved as macros.

GUIDANCE AND SUGGESTIONS IMPROVE FEEDBACK

The previous two projects demonstrated how contextual examples can help users get started on a new task or solve a problem during a task. Both techniques require the user to intentionally interact by typing a query or browsing examples in the interface. However, people are often unaware that their work can be improved. This project demonstrates how ambiently updating suggestions and guidance helps novices improve their output while they work [7].

Giving feedback on creative work is itself a creative task, as it requires critical analysis of the work, thinking of novel comments, and articulating them in a helpful way. There is increasing demand for quality feedback – and thus feedback givers – in learning and professional settings. However,

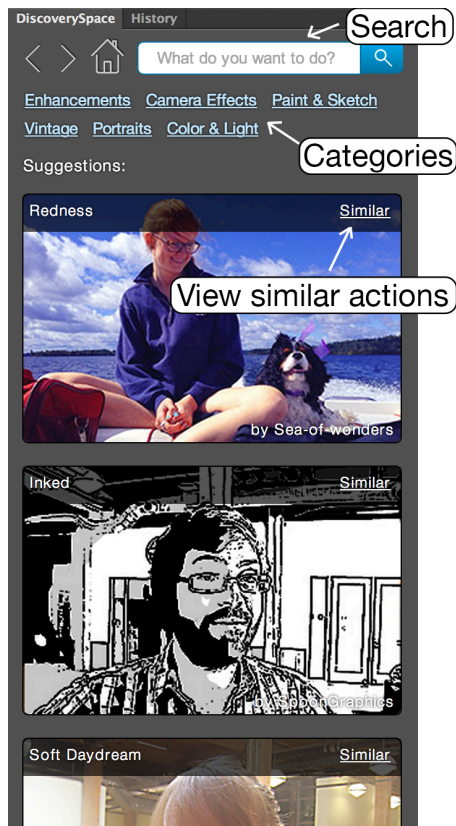


Figure 2: The DiscoverySpace recommendation panel. Each image represents a Photoshop action macro downloaded from the web. Clicking an action applies it to the user's own photo. Users can also search, browse categories, or browse similar actions to discover more options.

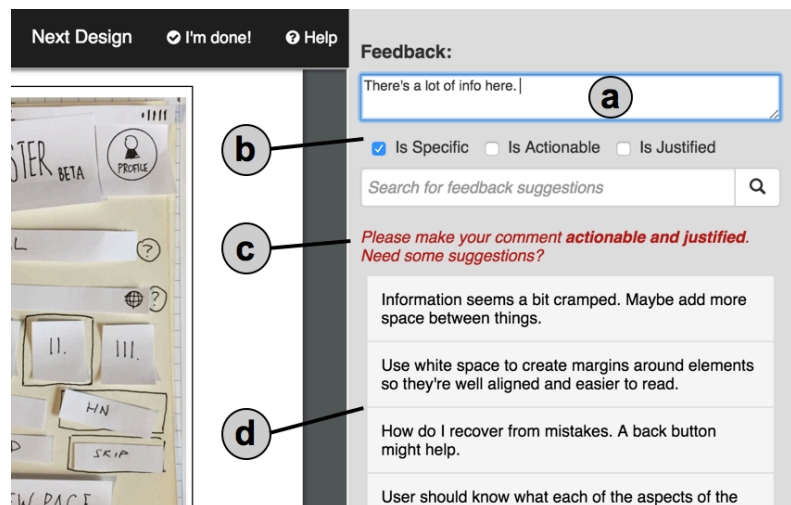


Figure 3: The CritiqueKit online system for giving feedback on creative work. a) The reviewer types their feedback in the textbox. b) The guidance checkboxes update based on this feedback. c) Adaptive prompts encourage reviewers to ensure their comment fits the guidance checkboxes. d) Feedback suggestions update based on the currently unchecked boxes.

good feedback can be hard to generate, and people are not consistently skilled in doing so [4]. To help novices give better feedback on creative work, we developed the CritiqueKit system (Figure 3), which embodies two novel techniques: 1) Interactive guidance of feedback characteristics in the form of checkboxes that update ambiently based on the user's feedback, and 2) reusable expert suggestions that update based on the feedback's categorization to provide targeted ideas and inspiration.

Two deployment studies and two controlled experiments investigated the efficacy of these interactive techniques on the quality and characteristics of feedback. We iterated on the design and algorithms behind CritiqueKit based on feedback through the two deployments. The first experiment examined the impact of statically presented suggestions and interactive guidance on novice feedback, and found no effect. The second experiment examined the efficacy of adaptively updating suggestions in tandem with interactive guidance on novice feedback. We found that adaptively-presented suggestions and guidance significantly improved feedback quality. Reviewers found suggestions useful for inspiration, and the interactive guidance reminded them to ensure their comments met the criteria for effective feedback. These findings highlight the importance of adapting contextual assistance to the user's own situation; when suggestions were not adaptive, reviewers did not find them relevant or helpful and they did not improve output.

CONCLUSION

I would love to brainstorm ways to tie together the main themes and goals of my somewhat diverse projects into a concise and valuable dissertation with the UIST DC Committee. I welcome feedback on the overall framing and themes I have developed so far, as well as ideas for appropriate metrics to evaluate how well my systems support creativity, as creative work by definition cannot easily be judged as correct/incorrect.

Overall, my dissertation presents *techniques for supporting creative learning through contextual presentation of expert examples* that can be applied to software and tasks in many different domains. Specifically, my work makes the following contributions:

1. Novel algorithms and interfaces for aggregating and presenting three types of expert examples (videos, macros, and output) at three different stages of creative work (when stuck, getting started, and while working),
2. The implementations of three prototype systems that embody these methods, and
3. Experimental results that demonstrate the efficacy of these methods toward supporting creativity.

Through this research, I hope to enable more people to reach their creative potential and lower the barriers to getting started and completing projects.

ACKNOWLEDGMENTS

I am grateful to my advisors Scott Klemmer and Mira Dontcheva for their continual guidance and feedback. This work was supported in part by Adobe, NSERC, SAP, and UC San Diego.

REFERENCES

1. C. Ailie Fraser, Mira Dontcheva, Holger Winnemoeller, Sheryl Ehrlich, and Scott R. Klemmer. 2016. DiscoverySpace: Suggesting Actions in Complex Software. *DIS '16: Proceedings of the 2016 conference on Designing Interactive Systems*.
2. Tovi Grossman and George Fitzmaurice. 2010. ToolClips: An Investigation of Contextual Video Assistance for Functionality Understanding. *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, ACM Press, 1515. <http://doi.org/10.1145/1753326.1753552>
3. Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. 1985. Direct Manipulation Interfaces. *Human-Computer Interaction* 1, 4: 311–338. http://doi.org/10.1207/s15327051hci0104_2
4. Chinmay Kulkarni, Koh Pang Wei, Huy Le, et al. 2013. Peer and self assessment in massive online classes. *ACM Transactions on Computer-Human Interaction* 20, 6: 1–31.
5. Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2011. Ambient help. *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*, ACM Press, 2751. <http://doi.org/10.1145/1978942.1979349>
6. Joanna McGrenere and Gale Moore. 2000. Are We All in the Same “Bloat”? *Proceedings of the Graphics Interface 2000 Conference, May 15-17*, 187–196. Retrieved from <http://graphicsinterface.org/wp-content/uploads/gi2000-25.pdf>
7. Tricia J. Ngoon, C. Ailie Fraser, Ariel S. Weingarten, Mira Dontcheva, and Scott Klemmer. 2018. Interactive Guidance Techniques for Improving Creative Feedback. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, ACM Press, 1–11. <http://doi.org/10.1145/3173574.3173629>
8. Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, et al. 2011. Pause-and-Play: Automatically Linking Screencast Video Tutorials with Applications. *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*, ACM Press, 135. <http://doi.org/10.1145/2047196.2047213>
9. Ben Shneiderman. 2007. Creativity support tools: Accelerating discovery and innovation. *Communications of the ACM* 50, 12: 20–32. <http://doi.org/10.1145/1323688.1323689>