RePlay: Contextually Presenting Learning Videos Across Software Applications

C. Ailie Fraser*
Design Lab, UC San Diego
La Jolla, CA
cafraser@ucsd.edu

Tricia J. Ngoon
Design Lab, UC San Diego
La Jolla, CA
tngoon@ucsd.edu

Mira Dontcheva
Adobe Research
Seattle, WA
mirad@adobe.com

Scott Klemmer
Design Lab, UC San Diego
La Jolla, CA
srk@ucsd.edu

ABSTRACT

Complex activities often require people to work across multiple software applications. However, people frequently lack valuable knowledge about at least one application, especially as software changes and new software emerges. Existing help systems either lack contextual knowledge or are tightly-knit into a single application. We introduce an application-independent approach for contextually presenting video learning resources and demonstrate it through the RePlay system. RePlay uses accessibility APIs to gather context about the user's activity. It leverages an existing search engine to present relevant videos and highlights key segments within them using video captions. We report on a week-long field study ($n = 7$) and a lab study ($n = 24$) showing that contextual assistance helps people spend less time away from their task than web video search and replaces current video navigation strategies. Our findings highlight challenges with representing and using context across applications.

CCS CONCEPTS

• **Information systems → Search interfaces**.

KEYWORDS

contextual search; video tutorials; cross-application

ACM Reference Format:
C. Ailie Fraser, Tricia J. Ngoon, Mira Dontcheva, and Scott Klemmer. 2019. RePlay: Contextually Presenting Learning Videos Across Software Applications. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3290605.3300527

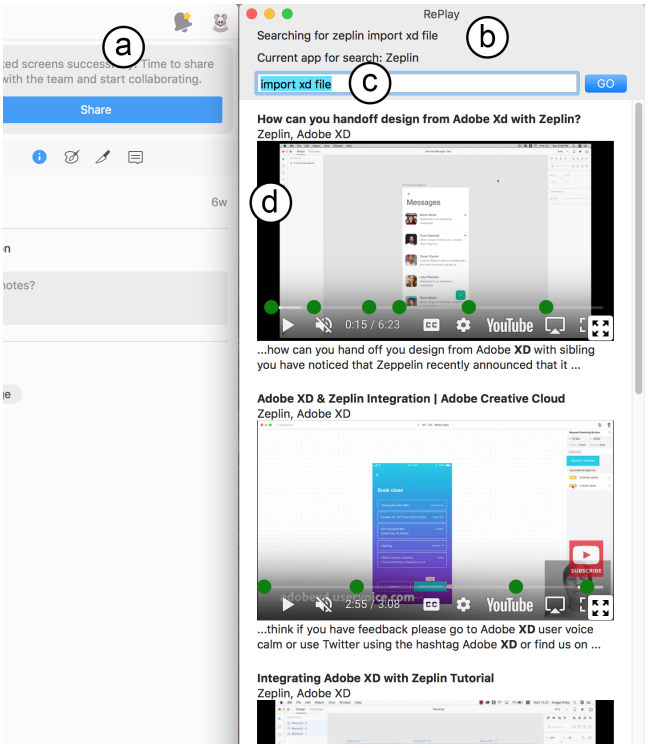

Figure 1: RePlay - shown to the right of Zeplin (a) - includes a status area (b) and search field (c) at the top. Video results (d) include timeline markers and caption excerpts to highlight moments relevant to the user's query and context.

1 INTRODUCTION

Most software activities span multiple applications. The slogan "there's an app for that" illustrates that we live in a world filled with specialized apps that each focus on a few specific tasks each. To accomplish larger activities requires composing multiple applications together into a "toolbelt" [49]. For example, designing an interface might comprise drawing a logo in Illustrator, mocking up a prototype in Sketch, adding animations using Flinto, and presenting it to a client using Keynote. Analyzing data might involve formatting it using Python, viewing and graphing it in Excel,

*Also with Adobe Research.

Permission to make digital or hard copies of all or part 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 components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CHI 2019, May 4–9, 2019, Glasgow, Scotland UK
© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-5970-2/19/05…$15.00
https://doi.org/10.1145/3290605.3300527

modifying graph aesthetics in Photoshop, and reporting results in Word. Toolbelts help users tailor custom ecosystems and support distributed innovation. However, this bricolage creates a user experience problem: even with design guidelines, every app is different [6]. As new applications appear and existing ones change, few people are fluent experts in all the steps towards their goals.

Presenting learning resources in-application [7, 11, 19, 23, 32, 33] and augmenting search queries with contextual information [7, 13] can offer a more fluid experience with lower cognitive load. However, existing solutions require deep integration with applications. And since today's applications are "walled gardens" with limited integration across software vendors [6], help resources typically focus on one application at a time. This leaves gaps when users want to move from one application to another (*e.g.,* export an Adobe XD prototype to Zeplin) or interleave applications (*e.g.,* coding a website in Sublime while debugging in Chrome and resizing graphics in GIMP).

Web search results can of course include community-created resources that span applications. However, generic web search poses two problems. First, search is blind to relevant contextual information that could connect users to better resources [13, 14, 26]. Search engines place the burden on users to articulate an appropriate query, an almost paradoxical requirement for users who are there because they don't know the domain [44]. Second, search is divorced from the application UX, requiring users to bounce back and forth to connect the content [15]. These challenges are amplified when users work with multiple applications, each with its own terminology and conventions.

We introduce an application-independent approach for contextually presenting video learning resources. We embody this approach in the RePlay system, which enables users to search for learning videos based on their application usage. RePlay gathers application context using system accessibility APIs. It extends online video search and cues videos to relevant segments based on their captions. We focus on video assistance because despite video's growing popularity (Cisco predicts that by 2021, 82% of all internet traffic will be video [2]), searching and browsing video remain cumbersome [25, 39, 40]. Video is popular for content creators as it is often easier to author than tutorials or diagrams (which require careful curation). Learners value video for its efficacy in communicating complex or continuous visual actions such as brushing or setting parameters [10]. However, interacting with videos remains difficult because they are harder to navigate and scan for steps than text [10].

We report on two studies observing how people use RePlay and web video help: a week-long field study ($n=7$) and a lab study ($n=24$) where half the participants used RePlay and half used web video search. Both used visual design as the domain, as video is especially helpful for visual tasks. The field study examined how designers with varying experience used RePlay in-situ. Participants used an average of 17 different applications in a week, emphasizing the importance of system-wide integration. It also suggested that contextual video assistance benefits targeted tasks more than open-ended ones. The lab study found that contextual video assistance helps people spend less time away from their task than web video search, and replaces strategies typically used in navigating between and within videos. This work makes the following contributions:

(1) An application-independent method to find relevant clips in learning videos that leverages user context,
(2) the RePlay system, which demonstrates this method using accessibility APIs and online video corpora, and
(3) insights from two studies that highlight the importance of multi-app support and promise of cross-app search.

2 RELATED WORK

This work builds on prior contextual search and video segmentation work with a novel focus on multi-app activities.

Multi-app activities are hard to support consistently

People often work across multiple applications that each support an individual task to perform higher-level activities [49]. Help systems for such applications tend to focus on their individual tasks rather than the transitions and interactions between them [37]. Implementing system-wide assistance that captures activity context is difficult, as every application has its own conventions and interface, and software interoperability tends to be limited [6]. Accessibility APIs are one useful entry point for diverse system-wide extensibility, including visualizing user behavior [34], voice control [30, 53], and modifying or enhancing existing user interfaces [9, 12, 47]. RePlay uses accessibility APIs for detecting system-wide application context.

Application context improves relevance and presentation of learning content

The most teachable moment for software tasks is often when the user is actively working... but stuck. In-task help for a specific goal is one of the biggest reasons people seek web [27] and video [4] tutorials. However, tutorials tend to show a full task from start to finish, much of which may not be relevant to the user's current goal. This leaves it to the user to both find a relevant tutorial and locate the segment(s) within it that contains the needed information.

Effectively searching the Web is an acquired skill; coming up with the right keywords and search settings can be difficult, especially for novices [44]. In addition, web search

environments lack context that human tutors use to proactively offer help and tailor feedback [45]. Adding contextual terms automatically to search queries (*e.g.,* OS version, application, recently-used tools) can help improve the relevance and utility of search results without requiring the user to know app-specific terminology [7, 13, 33]. RePlay augments queries with the current application name and uses context from both the current and recently-used applications to support cross-app activities when ranking search results.

Adding contextual cues to search results provides information scent that can help users more quickly and easily navigate the results (Figure 2). Such cues show how and why a result is relevant to the user's own situation and direct them to the relevant information within a result [13, 15]. RePlay expands these ideas to video, displaying contextual cues for recently-used applications and tools.

Bringing learning resources directly into applications reduces the need for context switching. For example, proactively recommending content in-situ based on user context can lead users to resources they might not have even thought to search for [11, 16, 23, 32, 35]. In-app search also helps users stay focused on their task while learning [11, 16, 28]. RePlay brings these strategies into an application-independent system, functioning alongside the user's applications.

Videos are helpful for learning but hard to navigate

Videos are popular for many kinds of tasks, especially visual ones, because they show clear demonstrations that might be harder to explain or understand in text [10, 41]. Videos are widely available and relatively easy to make. Software is always being updated, and new video demonstrations are constantly added to popular platforms by the user community to keep up with updates and current trends. However, while text-based documents are easy to skim and search, videos are not [25, 39, 40]. The predominant video search approach displays results as thumbnail images with a title and summary (*e.g.,* YouTube, Vimeo). This presentation only provides a broad overview without cues about matching content; prior work has shown that people look for indications of how search results are relevant to their query [21]. Navigating within videos is typically limited to hovering across the timeline with a small visual preview.

Automatically dividing videos into conceptual chunks can improve peoples' ability to find useful information for their task [5, 10, 19, 32, 36, 41]. RePlay uses captions to select relevant clips [17, 39, 40]. Automatic clip selection plays a limited but growing role in web search. Google displays a "suggested video clip" [48] as the automated summary for some searches, and Bing's "smart motion preview" feature [1] shows a 30-second preview for video search results. In contrast, RePlay focuses on searching with and presenting results within application context. Chi et al. [10] showed people benefit most from a mix of text and video; RePlay combines video and text instruction by presenting captions with videos for both navigational and learning assistance.

Methods for navigating between video segments include interactive timeline markers [5, 20, 24, 25, 32], thumbnail images [5, 10, 19, 24, 40, 41], transcript text [24], and clickable elements overlaid on the video [36]. Like Ambient Help [32], ToolScape [25], and Chronicle [20], RePlay overlays markers on the timeline indicating command or tool use. We use timeline markers over other options as they take up little space and allow for pop-up text previews, which aid browsing. While prior work marks *all* instances of tool use or interface actions, RePlay marks only *contextually-relevant* moments (recently-used tools and words from the user's query) to reduce clutter and unnecessary detail in a small interface.

3 REPLAY SYSTEM

This section describes RePlay's user interface, context detection, and system architecture.

User interface

The RePlay panel (Figure 1) can be positioned and sized as desired; its default size is 465px ×1055px. It is designed to fit next to the user's primary applications to minimize switching windows. The narrow pane makes videos small but easier to browse and watch in context. The interface comprises a status area, search field, and video results.

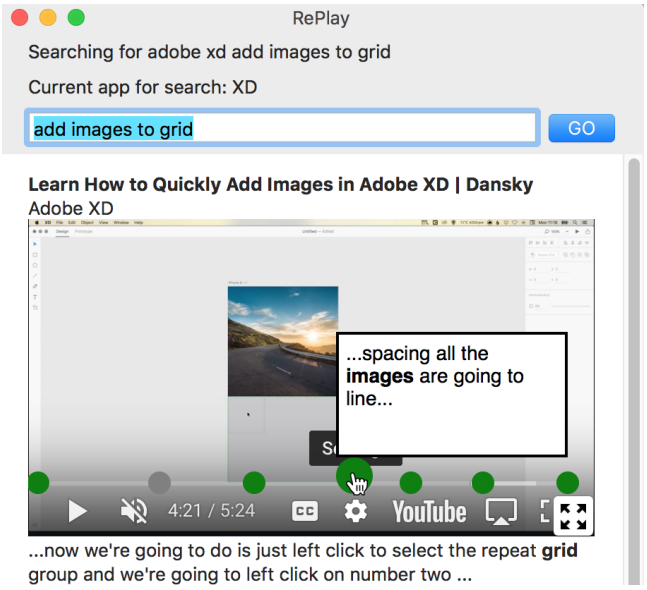

Figure 2: RePlay overlays green markers on the video timeline to indicate moments where the captions match a user's search term. Mousing over a marker shows a pop-up with an excerpt from the captions at that moment; clicking it plays the video from that moment.

The status area updates as the user works, displaying the name of the last tool clicked and the current application (Figure 3). "Tools" refer to interface elements or commands within an application. The status area provides awareness of what context RePlay will use for search (*i.e.,* so the user does not need to include the application name in their search query). When the user initiates a search, the status area updates to show the query (Figure 1b).

As the user works, the search field updates with the name of the last tool clicked (Figure 3). Users can edit or delete it to form their own query. Pressing the GO button or return key triggers a search. RePlay displays the top five resulting videos, each cued to start at a relevant moment. A two-line excerpt from that moment's caption appears below the video with query words highlighted in bold [21].

Often, videos have multiple moments that may be relevant. RePlay renders green markers on the video timeline to indicate these moments. Mousing over a marker invokes a pop-up text area displaying a caption excerpt with words from the query in bold (Figure 2). This pop-up obscures YouTube's default thumbnail pop-up but provides more useful information, as software videos tend to show an entire screen and shrinking this to a thumbnail makes it hard to see. Clicking a marker starts the video from that moment.

RePlay also displays contextual cues [13] on search results (Figure 4). For each video, RePlay lists the three most-recently used applications that are mentioned in the video (Figure 4a). This list is especially useful when users move between applications and want videos that mention both the current and recent applications. RePlay renders grey timeline markers to indicate moments where recently used tools are mentioned (Figure 4b). Caption pop-ups italicize tool names.

RePlay's panel shows all results at the same time, allowing users to quickly skim multiple videos, and browse other results while one video plays. This ability to "see inside" multiple resources from a single page increases foraging efficiency [18, 38, 51].

Clicking the full-screen button in a video's bottom-right corner opens the video in a separate window that stays above all other windows while it is open, so that users can watch a video at a larger size when desired.

When the user switches to a new application and clicks a tool, RePlay automatically searches for the application name.

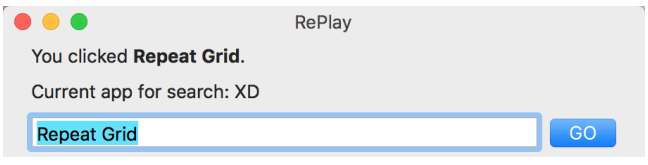

Figure 3: RePlay's status area displays tool names after they are clicked and adds them to the search field.

This seeds the panel with app-relevant videos that give users a starting point.

Detecting application context

RePlay uses contextual information to augment a user's query and search within video results. The motivation for context-augmented search is to increase relevance, especially when users don't know what to ask for. However, if not done well, adding terms has the opposite effect: excluding relevant results and/or presenting irrelevant ones [14]. We tried several heuristics with RePlay; the current implementation includes the three most recent applications and tools.

How to detect context? RePlay leverages OS accessibility APIs to detect every click. On each click event, RePlay retrieves the name of the click's source application, the type of element clicked, and the element's accessibility description (when present). If the element is a button, checkbox, text field, slider, or menu item, RePlay stores it as a recent tool and updates the status area and search field with the tool's name. If the user switched applications since their last click, RePlay updates its status area to reflect the new application, and resets its list of recent tools.

Challenges with detecting context via accessibility. Extracting accessibility text obviates the need for hard-coded knowledge

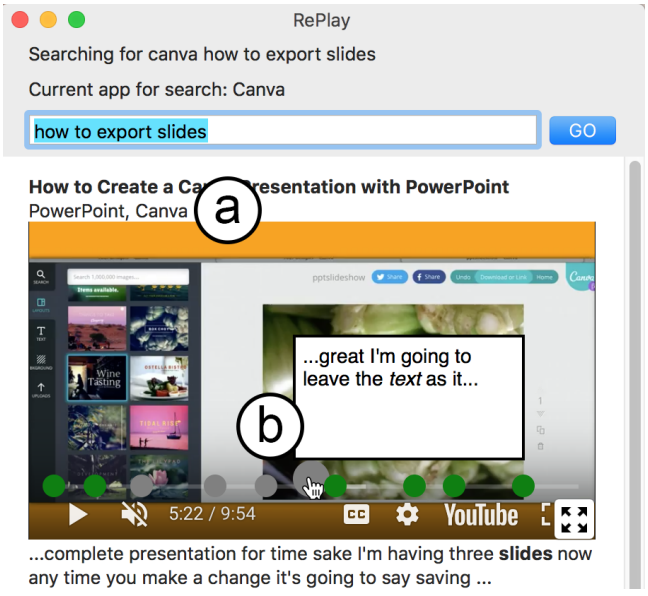

Figure 4: RePlay displays contextual cues based on recent app and tool use. a) RePlay lists the three most recent apps that the video mentions. b) Grey markers indicate mentions of recently-used tools. In this example, the user recently used the "text" tool in Canva. Mousing over a marker shows a caption excerpt; clicking a marker plays the video.

about specific applications. The challenge of this system-wide approach is that despite platform accessibility guidelines, applications vary widely in what accessibility they offer and how [22].

In MacOS, RePlay can always extract the application name and menu items. Buttons and other interface elements have accessibility labels in many applications (*e.g.,* Adobe XD, Microsoft Office, iMovie, Tableau, Sketch), but not in others (often long-existing software, *e.g.,* Photoshop). Applications differ in which accessibility fields they support and what information is in what field. For example, tool names may be in the *Title*, *Help*, *Description*, or *Value* attribute. RePlay checks all four, preferring them in that order. RePlay also gathers accessibility information for websites, as long as browser accessibility access is enabled. It is by default in Safari, and as an option in Chrome. Many sites implement accessibility labels (*e.g.,* Canva, Wordpress, Sketchpad, Snappa, G Suite). Those that do not still include some information by default (*e.g.*, text area contents via the *Value* attribute). RePlay infers website names from the URL and website title.

Video search & ranking

RePlay leverages existing online video search engines to retrieve video results. It then finds and ranks relevant clips within these videos (Figure 5). RePlay's architecture requires no prior understanding of the applications or videos; it relies on captions for clip extraction and ranking. Video authors often talk about things they are doing and provide tips about tools; captions thus provide narrative information beyond the visual content in the video that can be useful for learners.

Available data. RePlay's current video corpus comprises all English videos on YouTube that have a caption track. Most do: YouTube auto-generates captions by default. We used YouTube for its popularity and captions; any video search engine with an API could be used. For example, Vimeo (vimeo.com) and Dailymotion (dailymotion.com) also provide API access to videos and captions.

Searching. Video search requires more steps than document search, because captions are obtained separately. This two-step search means that issuing multiple queries with context terms added (such as tool usage) like prior work [13] would be too slow. To speed responses, RePlay constructs and issues a single query concatenating the current application's name with the user's query. Leaving context terms out of the query also ensures that the user-provided search terms are not "washed out". RePlay queries YouTube and selects its top five video results that have English captions and mention the current application in any of the title, description, or captions (to avoid results that may contain other keywords but do not pertain to the current application).

Finding clips and re-ranking videos. Several techniques automatically extract instructional video clips from screencasts of software use. The dominant approach leverages application usage [10, 20, 29, 52], requiring that the video be recorded in an instrumented version of the software. Alternatively, computer vision can detect tool-selection events [32, 41], even without prior knowledge about the specific software [5]. To be application-independent and embed online videos directly without waiting to download and process them, RePlay instead uses metadata and caption text to rank and segment videos.

For each video result, RePlay divides its captions into 30-second segments, searching each for the queried keywords (with stop words removed) and names of the three most-recently used tools in the current application. It ranks all segments by the total number of keyword matches. To break ties it uses number of tool name matches. The highest-ranked segment determines the video's start time. Timeline markers denote the top ten segments: green for those with a query term; grey if only a tool is mentioned. RePlay re-orders the video results based on the total number of matching clips. To break ties it uses the total number of matching keywords within the clips.

Although automatic captions are far from perfect, we found them to be sufficient for searching in RePlay. Captions are already an approximation of what the demonstrator is doing, so despite some errors, they work well enough for identifying potentially relevant moments. Having any aids for navigating within videos is still an improvement over

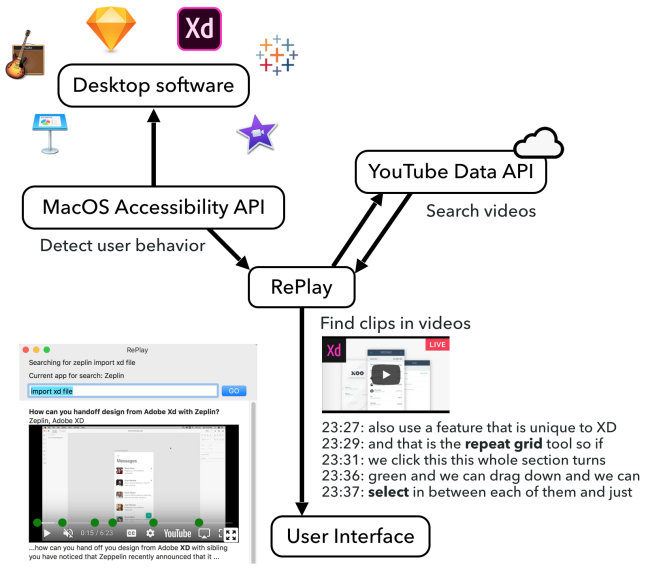

Figure 5: RePlay uses accessibility APIs to detect user context, which it uses to augment the user's query and to search and rank videos. RePlay finds matching clips by searching video captions for the user's query and recent tool names.

standard viewers. Still, future systems could allow viewers to easily correct errors as they watch to improve caption accuracy for future viewers.

Implementation

RePlay is implemented as a MacOS application in Swift. It uses the MacOS Accessibility API to extract information from input events. For both studies, RePlay used a whitelist to only detect clicks in certain applications and websites. A blacklist could be used instead; we explore this in the discussion.

When a search is triggered, RePlay queries the YouTube Data API's `search` method, which returns an ordered list of video IDs. For each ID, RePlay checks if English captions are available using YouTube's `get_video_info` method. If they are, this method's response includes a URL that RePlay follows to obtain subtitles in XML, with time stamps for every 5 to 10 words. A search that returns all new videos can take up to a few minutes to finish depending on network speed, due to the multiple requests needed to retrieve captions for each video. To speed up future searches, RePlay caches all retrieved captions locally (since they are pure text, this takes up little space). This could be further improved by proactively caching results for common search queries and buffering results as they come in. RePlay's video player is implemented in Javascript on a custom server; it embeds a YouTube video in an `iframe`, cues it to the given start time, and overlays timeline markers and pop-up captions. RePlay displays each video by loading this web player in a Swift `WKWebView` object.

4 STUDY 1: HOW DOES REPLAY SUPPORT ACTIVITIES IN THE WILD?

A week-long field study with seven participants investigated whether and how people might use contextually-augmented video search in their own work. We focused on visual design, as videos are especially helpful for visual work [10]. Participants were recruited from the local design community and a prominent creative software company (Table 1). Participants had mixed amounts of experience with the main design software used during the study (either Sketch or Adobe XD); all participants were proficient in at least one other creative application. Several had recently switched to Adobe XD or Sketch from other software, or had recently started their current job. After an initial interview, we installed RePlay on each participant's computer. Participants kept RePlay open throughout the next week while they worked (one used it for 10 days). Participants returned for a followup interview and received a $45 gift card as compensation for their time.

Participants used an initial version of RePlay (Figure 6); based on their feedback, we revised it to the version presented in this paper. Study 1's RePlay did not consider cross-app context; it only used context from the current application.

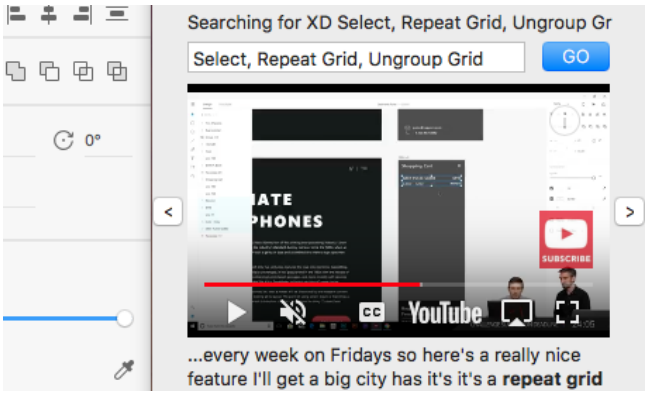

Figure 6: Study 1 used this initial version of RePlay, shown here next to Adobe XD. It did not show video titles or timeline markers; instead it had arrow buttons next to each video to skip between that video's ranked clips. It also added the 3 most recent tools to the search query instead of one.

This RePlay only monitored Adobe XD and Sketch to avoid capturing unrelated or private data from other applications. RePlay logged the following events to a server when it was open: all clicks on interface elements in Adobe XD and Sketch, all interactions with RePlay, and all switches to and between other applications.

We took notes on all interviews, noting similar answers to questions and identifying common themes. This data and feedback helped motivate the final version's focus on cross-app support, helped us understand what kinds of tasks RePlay may be most useful for, and highlighted some advantages and challenges of contextual support.

Results

Over the week, participants reported spending between 1.5 and 30 hours on their design work (Table 1). Some said they kept RePlay open the entire time; others closed it at times to focus on their work.

Generally, participants appreciated having help readily available. As *P2* explained, *"this gives me an interface where I can search and do everything and it's automatically there right next to the design. There are no extra steps." P4* enjoyed seeing RePlay react to her actions: *"it felt like I had a buddy."* Only one (*P6*, the expert) did not use it at all: he was fluent in the software and did not seek assistance.

Contextual clip search was most useful for specific tasks. Four participants said they tended to use RePlay when stuck trying to figure out how to do something specific. Three also said they used it to find out what a particular tool could do or how to use it. All but one participant worked on targeted tasks (see Table 1). *P3* wanted general resources for getting started with Sketch, and did not find RePlay helpful.

Three users recounted similar stories of searching for a particular question and quickly finding a clip within a video that answered it. *P1* described searching for "Make Symbol" after clicking on the "Make Symbol" tool in Adobe XD. The answer he needed came from an auto-selected moment near the end of a 3-hour video. *P1* added, *"if I had searched for that myself I would've given up."* Similarly, *P5* found what he needed in a 1.5 hour video that was cued to a moment 20 minutes in. He was trying to create margins, and searched for "layout grid", and *"The first video in the list showed me exactly what I needed to do"*, which was to check a box he hadn't noticed. *P4* found RePlay useful for indicating that her specific goal was *not* possible: she wanted to customize a grid system in Sketch and searched for "grid settings", but the caption excerpts indicated that none of the results mentioned customizing grids.

Contextual video clips sometimes invited opportunistic learning: *P1* and *P7* recounted instances where a video they were watching taught them something they didn't know and hadn't thought to look for. *P1* continued watching the "make symbols" video as it described grouping and layering symbols. This part *"wasn't originally what I was searching for but it was exactly what I needed ... [I] gained a lot more knowledge." P7* described how he *"searched for 'character styles' and actually found new information that I've never seen before"* about the Libraries panel.

Tool context helps, but not in the search query. All participants who searched with RePlay preferred deleting tool names from the query and instead typing their own. Five participants mentioned that including three tools in the query was too many, as it made the query too general. *P3* said this was because the automatic query seemed *"stuck on the last thing I did, which might not be relevant to what I'm thinking now."* Both *P4* and *P5* said tool *names* were not as useful because they wanted to search for an *action*, not its constituent tools (*e.g.,* "rotate object" or "export nested artboard"). Higher-level activity inference may provide more useful assistance.

Two participants said they liked that RePlay populated the query with tool names because *"coming up with the right search terms is really hard and I don't know the names of the tools [...] so it's nice not to have to think of them"* (*P4*). Even *P6*, the expert, said *"I know how to use everything but if you asked me the names of the tools, I have no idea."* Participants appreciated that RePlay added the application name to the query, as it allowed them to *"spend more brain power on the details of [the query]" (P2).*

Participants frequently switched between applications. RePlay logged every time users switched between *any* applications when it was running. Excluding system applications (such as Dock, Finder, and System Preferences), participants used an average of 17 different applications while RePlay was running ($SD = 8$), and switched between applications a mean of every 6.6 minutes ($SD = 8.5$min). If we exclude all continuous periods of over 3 hours (assuming that these sessions were breaks of some sort), this average lowers to every 2.5 minutes ($SD = 1.8$min). This diverse app usage supports our motivation for supporting cross-app workflows.

Improvements to RePlay. Because participants thought that auto-including three tools was too many, Replay now includes only the most recent one. Four participants also mentioned that more general information like video titles would be helpful, as caption excerpts often seemed *"out of context"* or *"snatched out of middle of a sentence"* (*P2*). Consequently, RePlay now includes video titles. Four participants wished they could watch videos at a larger size; RePlay now includes a resizable video player window. Two participants also mentioned that some video results did not pertain to the current application; RePlay now excludes results that do not mention the current application in the metadata or captions.

5 STUDY 2: HOW DOES REPLAY COMPARE TO CURRENT METHODS FOR VIDEO ASSISTANCE?

A between-subjects study ($n = 24$) investigated how contextual assistance for multi-app activities might affect behavior compared to standard web video search. It found that contextual assistance can reduce time spent searching and navigating videos.

Study procedure

Participants were asked to imagine that they were designers working for a client developing a travel journal mobile

	Job title	Main app	Experience w/ app	Design work done	Hours designing	Time w/ RePlay open	# queries	# videos watched
P1	Sr. Product Manager	Adobe XD	Beginner	Style guide and wireframing	20	100%	3	4
P2	Freelance Designer	Adobe XD	Beginner	Wireframing / prototype design	4.5	100%	10	5
P3	Freelance Designer	Sketch	Beginner	Trying to learn Sketch	1.5	100%	4	2
P4	PhD Student	Sketch	Intermediate	Screen design and grid customizing	5	40%	10	0
P5	UX Designer	Sketch	Beginner	Creating templates and logos	30	70%	24	8
P6	Sr. UX Designer	Adobe XD	Expert	UX workflows	20	25%	0	0
P7	Sr. UX Designer	Adobe XD	Intermediate	Wireframing an app UI	12	33%	19	13

Table 1: Study 1 participant background and usage. Participants self-reported their experience, design work done, hours spent designing, and % of that time with RePlay open. # queries and # videos watched were calculated from RePlay's usage data.

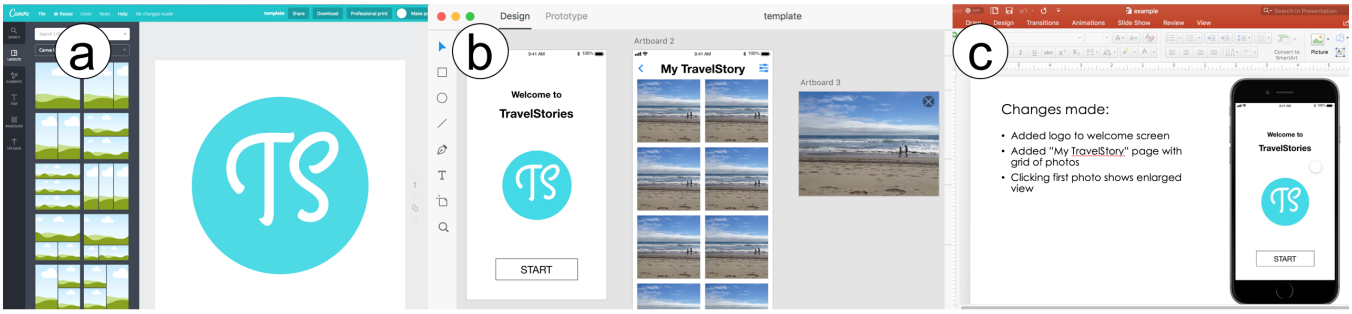

Figure 7: Study 2 asked participants to make changes to an initial logo in Canva (a), update a prototype in Adobe XD (b), and make a presentation showing their changes in PowerPoint (c).

app. To replicate a multi-app activity, the study task spanned three applications: Canva (canva.com), Adobe XD, and Microsoft PowerPoint. Participants were asked to improve and change an initial design for their client (Figure 7). The task had both creative and technical requirements. Creative requirements included making the logo more travel-themed, adding visual appeal to the prototype welcome screen, and making a PowerPoint presentation to show to the client. Technical requirements included adding additional photos to a grid, rounding their corners, and recording a video walkthrough. If participants needed help, they were instructed to search for tutorial videos using either RePlay or YouTube in a web browser. RePlay participants were introduced to its features prior to the task. To ensure that all participants had access to the same resources, they were asked not to use other search engines or resources. Participants had 45 minutes to complete the task, and answered questions about their experience and help-seeking at the end. Participants were compensated with a $15 gift card.

Participants. Twenty four participants (14 female) were recruited through online and paper advertisements on a university campus. Prior to the study, participants filled out a survey about their design experience. Participants were randomly assigned to either the RePlay (n = 12) or Web (n = 12) condition and counterbalanced based on experience. "Novice" was defined as completing at most two courses in design and reporting experience with at most two design applications. "Experienced" was defined as completing more than two courses in design or reporting experience with more than two design applications. Six of the 24 participants were "experienced".

Participants rated their pre-study familiarity with each of the three study applications on a 5-point Likert Scale. Participants were generally familiar with Powerpoint ($mean$ = 4.2), and unfamiliar with Canva ($mean$ = 1.5) and Adobe XD ($mean$ = 1.2).

Measures. We were primarily interested in participants' search behavior. We measured the number of queries, their length, and time spent in the search interface. Qualitative measures included how participants determined which videos to watch and their navigation strategies (observed through screen recordings). We also gathered participant feedback in the RePlay condition on its features.

Results

RePlay participants averaged 3.3 queries each (33 total); Web participants averaged 3 queries each (24 total) (x^2 = 1.42, df = 1, p = .23). Web participants typed longer queries ($mean$ = 4.33 words, SD = 1.41) than RePlay participants ($mean$ = 2.53 words, SD = 0.59)(t = 2.52, df = 15.98, $p < .01$), because Web participants often manually added the application name, whereas RePlay auto-included it. 72% of search queries were for Adobe XD and 28% were for Canva; none were for PowerPoint since participants were more familiar with it. Four Web and two RePlay participants did not search for any assistance; we cover this in the Discussion.

Participants varied considerably in the amount of time and effort they spent. This is a challenge with a task that has both creative and technical components; people prioritize these components differently. Many participants spent a long time perfecting their design and had to be cut off after 45 minutes; others did the minimum required and finished in as few as 23 minutes. Because the task was open-ended, we did not compare completion times across participants. Our analysis focuses on search behavior.

Contextual assistance lessens time away from task. Web participants spent nearly twice as long in the search interface ($mean$ = 214.5 seconds, SD = 127.1) as RePlay ones ($mean$ = 116.2 seconds, SD = 58.9). Due to the high variance and small sample size, the difference was marginally significant (t = 2.02, df = 9.4, p = .07) (Figure 8 left). When the time each participant spent in search is averaged by the number of queries they made, the difference is significant: RePlay participants spent about 40% less time in search per query ($mean$ = 42.83 seconds, SD = 29.5) than Web participants ($mean$ = 72.62 seconds, SD = 22.4)(t = 2.52, df = 15.28, p = .02) (Figure 8 right). Though RePlay pre-cached many

video captions beforehand, we could not predict all queries users would make. Thus, some query responses in RePlay took up to two minutes. Because this latency could be reduced, we subtracted loading times in both conditions from the total time in search.

Easier navigation both within and between video results may explain why RePlay participants spent less time in the search interface per query. Web participants used various strategies for navigating within videos, including keyboard shortcuts to fast-forward and rewind, increasing video speed, and hovering over the timeline. Navigating between different video results in YouTube required selecting a video, watching or skipping through it to determine whether it was relevant, and if not, going back to the results page and selecting another. RePlay's timeline markers replaced many of these strategies, increasing efficiency: participants could first examine the timeline markers before deciding if and at what point to watch a video result. Eight RePlay participants hovered over timeline markers to read the caption previews, often hovering over multiple points before deciding where to watch. RePlay's panel interface also enabled participants to simultaeously play one video and examine others. We observed three participants do this, likely to decide whether another video was better without giving up on their first choice. One such participant said they wished for better visual cues of video relevance to help decide which to watch. Other participants browsed videos one at a time, perhaps to focus on the playing video.

Search queries were action-oriented, not tool-oriented. As in Study 1, participants removed tool names from queries. No participants in either condition used tool names. Instead, participants wrote action-oriented queries: the most common were "crop photos", "round corners", and "record video."

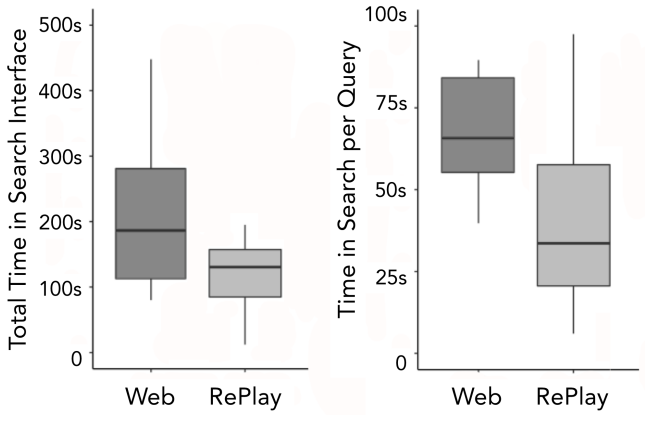

Figure 8: RePlay participants spent marginally less time overall in the search interface than Web participants ($p = .07$, left) and significantly less time *per query* ($p = .02$, right).

RePlay Feature	Title	Thumbnail	Caption	Timeline
Helpfulness	2.8	3.1	2.5	4.2

Table 2: Average helpfulness ratings for RePlay's contextual features ($n = 12$). Most participants found the timeline markers most helpful for determining which videos to watch.

Despite RePlay adding the current tool name to the search field, in all instances but one, participants deleted the tool name from their query.

Intra-video context is most helpful. In interviews, participants rated timeline markers as the most helpful (4.2/5) (Table 2). RePlay participants hovered over timeline markers a total of 69 times and clicked on markers 22 times. One participant mentioned that timeline markers provide a *"scaffold of what to look for and where to start watching."* Participants rated caption excerpt and video title as less helpful. A few participants mentioned ignoring titles and excerpts in favor of timeline markers and video thumbnails. Twice in each condition, participants selected the first video result even though the title mentioned the wrong application (Photoshop or Illustrator). This suggests that the video region is a strong magnet for people's attention.

6 DISCUSSION & FUTURE WORK

Observations and feedback from the two studies suggest several opportunities for future work.

How can contextual assistance aid exploration?

Study 2 participants often preferred manually exploring when it would have been more efficient to search for help. The study's subjective, open-ended task may have encouraged such exploration. Only one participant used the best method to update the photo grid - using Adobe XD's repeat grid feature to update all photos simultaneously - after learning about it in a video. Other participants used various less-efficient methods (*e.g.*, un-grouping the repeat grid and manually adding and re-sizing photos). Because these methods achieved their desired goal, participants may not have thought to investigate whether a faster option existed.

Interface exploration and browser search each have shortcomings. Exploration is a common problem-solving strategy because it can be enjoyable, it builds on domain knowledge, and each individual action is low cost [28, 42]. However, for novices especially, exploration is cumulatively slow, its learned knowledge is hard to integrate, and it induces a high cognitive load [28, 50]. A limitation of learning unfamiliar domains via exploration is that people may settle for suboptimal methods or strategies because they are unaware of a better alternative. Moving from an application to a web search can yield better results, but has a higher initial cost

as it lurches people out of their task. We believe that contextual search has the potential to offer the benefits of both approaches without either of the drawbacks.

Six participants in Study 2 did not search at all; most felt they could figure things out via exploration. One participant said they *"felt like I could find it if I searched for it [in the interface], which I did. I was able to figure it out."* Two other participants mentioned that they felt searching for help would be more time-consuming than trial-and-error exploration. One stated that he wanted to search for help, but *"I knew I didn't have time. I wanted to complete the task so I just hacked it."* If Study 2's result that contextual search reduces search time holds, these perceptions may change over time. RePlay's occasionally-slow loading times may have also affected this perception; prior work shows that a difference in latency of search results as small as 300ms can discourage people from searching [8].

How might contextual assistance encourage productive exploration while providing intervention when needed? While proactive support can be beneficial, a challenge is providing assistance without being too disruptive [32]. To minimize distractions, the RePlay interface mostly changes only in direct response to user input. However, novices may not even realize they need help, making proactive suggestions more valuable. For example, RePlay could automatically refresh video results when the user seems stuck, or suggest relevant queries when the user begins to search. Future work should examine how these alternatives might change people's behavior and workflows over time.

What and how much context to include?

While logging recent tools can help suggest next steps [35], we found that using recent tools explicitly in search queries is not useful. Participants in both studies did not use tool context in their queries, preferring action-oriented queries instead. Usage history is by definition retrospective (*i.e.*, it describes what the user has already done). In contrast, search is often prospective (*i.e.*, looking for something the user hasn't done yet). Tool context may only be helpful closer to where users interact with tools (*e.g.,* as part of tool tips [19]).

Interestingly, people used the same terms and concepts across different applications. Study 2 participants searched "crop photos" for both Canva and Adobe XD despite neither application having an explicit crop tool. This highlights both a challenge and an opportunity: people bring mental models that may not carry over into different applications. Study 1 suggested that displaying tool names may help people learn app-specific terminology. However, for both knowledge and speed reasons, users sometimes omit valuable terms. RePlay's interface benefits are reduced when users don't search using the same terms as videos. A natural language mapping [3] between video captions (along with other natural language data like comments and tags) and the commands they mention may increase captions' value for search.

What design challenges remain?

Many design decisions were motivated by our focus on cross-app workflows; *e.g.,* showing results in a separate window in a consistent location. Showing all results together made it easy to browse multiple resources at once. However, some participants still preferred focusing on one video at a time. Future work should consider how different layouts influence browsing behaviors and which behaviors lead to more effective workflows.

Currently, users must explicitly whitelist an application for RePlay to capture its events. While this approach offers more privacy, it also adds burden for users. Blacklisting, on the other hand, would allow RePlay to respond to all applications except for explicitly-omitted potentially-sensitive ones (*e.g.,* Messages), offering broader benefits. For our initial studies, we chose the greater privacy of a whitelist. For real use, users could choose whitelist or blacklist, and/or RePlay could request approval for each new app, similar to websites that ask for a user's location.

What other domains might benefit?

RePlay's main insight is that given a source of user context, we can search, curate, and index into resources from a large corpus. RePlay demonstrates this approach using video; different activities (*e.g.,* programming) may benefit from other types of content (*e.g.,* text resources). RePlay could naturally be extended to any textual content (or content with textual metadata). Text results could be displayed as short summaries with clickable keywords to expand more detail [13]. For detecting context, RePlay used MacOS's accessibility API; other OSs (*e.g.* Windows [34]) also have similar APIs. Beyond software, RePlay's approach could extend to any domain for which online videos are abundant (*e.g.,* physical building tasks). To detect activity context, one could augment physical tools with sensors [31, 46] or track body poses with wearable sensors or computer vision. A challenge for future work is to convert sensor or vision data into text searches, or to index videos using the sensor or vision data directly.

7 CONCLUSION

This paper introduced an application-independent approach for contextually presenting videos and a demonstration of this approach in the RePlay system. RePlay shows how system accessibility features and video captions can be used to detect context and search within videos in a flexible, domain-general way. Like curb cuts or closed captioning [43], RePlay demonstrates how accessibility features can

provide universally-beneficial assistance. Expanding accessibility and increasing cross-app consistency through guidelines and enforcement would benefit everyone. It would also expand application tailoring, integration, and assistance with systems like RePlay. Two studies demonstrated that cross-app contextual video assistance helps users spend more time on their task and less time searching for help. We also observed how contextual assistance can sometimes be at odds with peoples' desire to explore and tinker, and that the context most easily accrued from software usage may not always be the most relevant. Future work should investigate these challenges and examine how contextual help affects workflows in the real world through a longitudinal study. This work brings us one step closer to leveraging the wisdom of the Web for personalized, just-in-time learning.

8 ACKNOWLEDGMENTS

We are thankful to Michelle Lee for her assistance in conducting Study 2, and to all study participants for their time and feedback. This work was supported in part by NSERC and Adobe Research.

REFERENCES

[1] 2013. Watch Before You Click: Smart Motion Preview. https://blogs.bing.com/search/2013/05/07/watch-before-you-click-smart-motion-preview

[2] 2017. *Cisco Visual Networking Index: Forecast and Methodology, 2016-2021.* Technical Report. Cisco. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf

[3] Eytan Adar, Mira Dontcheva, and Gierad Laput. 2014. CommandSpace: Modeling the Relationships between Tasks, Descriptions and Features. In *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14.* ACM Press, New York, New York, USA, 167–176. https://doi.org/10.1145/2642918.2647395

[4] Samiul Alam Anik. 2015. *Integrating Comments in Video Tutorials.* Master's thesis. https://mspace.lib.umanitoba.ca/handle/1993/31046

[5] 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 - UIST '12.* ACM Press, New York, New York, USA, 83. https://doi.org/10.1145/2380116.2380129

[6] Michel Beaudouin-Lafon and Wendy E. Mackay. 2018. Rethinking Interaction: From Instrumental Interaction to Human-Computer Partnerships. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18.* ACM Press, New York, New York, USA, 1–5. https://doi.org/10.1145/3170427.3170635

[7] Joel Brandt, Mira Dontcheva, Marcos Weskamp, and Scott R. Klemmer. 2010. Example-centric programming: Integrating Web Search into the Development Environment. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10.* ACM Press, New York, New York, USA, 513. https://doi.org/10.1145/1753326.1753402

[8] Jake Brutlag. 2009. *Speed Matters for Google Web Search.* Technical Report. http://services.google.com/fh/files/blogs/google_delayexp.pdf

[9] Tsung-Hsiang Chang, Tom Yeh, and Rob Miller. 2011. Associating the visual representation of user interfaces with their internal structures and metadata. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11.* ACM Press, New York, New York, USA, 245. https://doi.org/10.1145/2047196.2047228

[10] Pei-Yu Chi, Sally Ahn, Amanda Ren, Mira Dontcheva, Wilmot Li, and Björn Hartmann. 2012. MixT: Automatic generation of step-by-step mixed media tutorials. In *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12.* ACM Press, New York, New York, USA, 93. https://doi.org/10.1145/2380116.2380130

[11] Parmit K. Chilana, Andrew J. Ko, and Jacob O. Wobbrock. 2012. LemonAid: Selection-based crowdsourced contextual help for web applications. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12.* ACM Press, New York, New York, USA, 1549. https://doi.org/10.1145/2207676.2208620

[12] Morgan Dixon, Alexander Nied, and James Fogarty. 2014. Prefab layers and prefab annotations: extensible pixel-based interpretation of graphical interfaces. In *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14.* ACM Press, New York, New York, USA, 221–230. https://doi.org/10.1145/2642918.2647412

[13] Michael Ekstrand, Wei Li, Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2011. Searching for software learning resources using application context. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11.* ACM Press, New York, New York, USA, 195. https://doi.org/10.1145/2047196.2047220

[14] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: the concept revisited. *ACM Transactions on Information Systems* 20, 1 (jan 2002), 116–131. https://doi.org/10.1145/503104.503110

[15] Adam Fourney, Ben Lafreniere, Parmit Chilana, and Michael Terry. 2014. InterTwine: creating interapplication information scent to support coordinated use of software. In *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14.* ACM Press, New York, New York, USA, 429–438. https://doi.org/10.1145/2642918.2647420

[16] C. Ailie Fraser, Mira Dontcheva, Holger Winnemöller, Sheryl Ehrlich, and Scott Klemmer. 2016. DiscoverySpace: Suggesting Actions in Complex Software. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems - DIS '16.* ACM Press, New York, New York, USA, 1221–1232. https://doi.org/10.1145/2901790.2901849

[17] Andreas Girgensohn, John Adcock, Matthew Cooper, and Lynn Wilcox. 2005. A Synergistic Approach to Efficient Interactive Video Retrieval. In *Human-Computer Interaction - INTERACT 2005*, Maria Francesca Costabile and Fabio Paternò (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 781–794.

[18] Elena L. Glassman and Daniel M. Russell. 2016. DocMatrix: Self-teaching from multiple sources. *Proceedings of the Association for Information Science and Technology* 53, 1 (jan 2016), 1–10. https://doi.org/10.1002/pra2.2016.14505301064

[19] Tovi Grossman and George Fitzmaurice. 2010. ToolClips: An Investigation of Contextual Video Assistance for Functionality Understanding. In *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10.* ACM Press, New York, New York, USA, 1515. https://doi.org/10.1145/1753326.1753552

[20] Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2010. Chronicle: Capture, exploration, and playback of document workflow histories. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology - UIST '10.* ACM Press, New York, New York, USA, 143. https://doi.org/10.1145/1866029.1866054

[21] Marti Hearst. 2009. Chapter 5: Presentation of Search Results. In *Search User Interfaces.* Cambridge University Press, Cambridge, Chapter 5, 120–139. https://doi.org/10.1017/CBO9781139644082

[22] Amy Hurst, Scott E. Hudson, and Jennifer Mankoff. 2010. Automatically identifying targets users interact with during real world tasks. In *Proceedings of the 15th international conference on Intelligent user interfaces - IUI '10*. ACM Press, New York, New York, USA, 11. https://doi.org/10.1145/1719970.1719973

[23] Michelle Ichinco, Wint Yee Hnin, and Caitlin L. Kelleher. 2017. Suggesting API Usage to Novice Programmers with the Example Guru. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. ACM Press, New York, New York, USA, 1105–1117. https://doi.org/10.1145/3025453.3025827

[24] Juho Kim, Philip J. Guo, Carrie J. Cai, Shang-Wen (Daniel) Li, Krzysztof Z. Gajos, and Robert C. Miller. 2014. Data-driven interaction techniques for improving navigation of educational videos. In *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14*. ACM Press, New York, New York, USA, 563–572. https://doi.org/10.1145/2642918.2647389

[25] Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J. Guo, Robert C. Miller, and Krzysztof Z. Gajos. 2014. Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*. ACM Press, New York, New York, USA, 4017–4026. https://doi.org/10.1145/2556288.2556986

[26] Reiner Kraft, Farzin Maghoul, and Chi Chao Chang. 2005. Y!Q: contextual search at the point of inspiration. In *Proceedings of the 14th ACM international conference on Information and knowledge management - CIKM '05*. ACM Press, New York, New York, USA, 816. https://doi.org/10.1145/1099554.1099746

[27] Ben Lafreniere, Andrea Bunt, Matthew Lount, and Michael Terry. 2013. Understanding the Roles and Uses of Web Tutorials. In *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media*. Association for the Advancement of Artificial Intelligence.

[28] 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 - OzCHI '14*. ACM Press, New York, New York, USA, 49–58. https://doi.org/10.1145/2686612.2686620

[29] 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 32nd annual ACM conference on Human factors in computing systems - CHI '14*. ACM Press, New York, New York, USA, 4007–4016. https://doi.org/10.1145/2556288.2557142

[30] Toby Jia-Jun Li, Amos Azaria, and Brad A. Myers. 2017. SUGILITE: Creating Multimodal Smartphone Automation by Demonstration. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. ACM Press, New York, New York, USA, 6038–6049. https://doi.org/10.1145/3025453.3025483

[31] Paul Lukowicz, Jamie A Ward, Holger Junker, Mathias Stäger, Gerhard Tröster, Amin Atrash, and Thad Starner. 2004. Recognizing Workshop Activity Using Body Worn Microphones and Accelerometers. Springer, Berlin, Heidelberg, 18–32. https://doi.org/10.1007/978-3-540-24646-6_2

[32] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2011. Ambient help. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. ACM Press, New York, New York, USA, 2751. https://doi.org/10.1145/1978942.1979349

[33] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2011. IP-QAT: in-product questions, answers, & tips. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*. ACM Press, New York, New York, USA, 175. https://doi.org/10.1145/2047196.2047218

[34] 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 - CHI '13*. ACM Press, New York, New York, USA, 3227. https://doi.org/10.1145/2470654.2466442

[35] Justin Matejka, Wei Li, Tovi Grossman, and George Fitzmaurice. 2009. CommunityCommands: Command Recommendations for Software Applications. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology - UIST '09*. ACM Press, New York, New York, USA, 193. https://doi.org/10.1145/1622176.1622214

[36] Cuong Nguyen and Feng Liu. 2015. Making Software Tutorial Video Responsive. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*. ACM Press, New York, New York, USA, 1565–1568. https://doi.org/10.1145/2702123.2702209

[37] Donald A. Norman. 2005. Human-centered design considered harmful. *interactions* 12, 4 (jul 2005), 14. https://doi.org/10.1145/1070960.1070976

[38] Amy Pavel, Floraine Berthouzoz, Björn Hartmann, and Maneesh Agrawala. 2013. *Browsing and Analyzing the Command-Level Structure of Large Collections of Image Manipulation Tutorials*. Technical Report. http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-167.html

[39] Amy Pavel, Dan B. Goldman, Björn Hartmann, and Maneesh Agrawala. 2015. SceneSkim: Searching and Browsing Movies Using Synchronized Captions, Scripts and Plot Summaries. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15*. ACM Press, New York, New York, USA, 181–190. https://doi.org/10.1145/2807442.2807502

[40] 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 - UIST '14*. ACM Press, New York, New York, USA, 573–582. https://doi.org/10.1145/2642918.2647400

[41] Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, Jue Wang, Lubomir Bourdev, Shai Avidan, and Michael F. Cohen. 2011. Pause-and-Play: Automatically Linking Screencast Video Tutorials with Applications. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*. ACM Press, New York, New York, USA, 135. https://doi.org/10.1145/2047196.2047213

[42] John Rieman. 1996. A field study of exploratory learning strategies. *ACM Transactions on Computer-Human Interaction* 3, 3 (sep 1996), 189–218. https://doi.org/10.1145/234526.234527

[43] David H. Rose and Anne Meyer. 2002. *Teaching every student in the Digital Age: Universal design for learning*. Association for Supervision and Curriculum Development. 216 pages.

[44] Daniel M. Russell. 2011. Making the Most of Online Searches. *APS Observer* 24, 4 (apr 2011). https://www.psychologicalscience.org/observer/making-the-most-of-online-searches

[45] Donald A. Schoen. 1983. *The reflective practitioner: how professionals think in action*. Basic Books. 374 pages. https://books.google.com/books/about/The_Reflective_Practitioner.html?id=ceJIWay4-jgC

[46] Eldon Schoop, Michelle Nguyen, Daniel Lim, Valkyrie Savage, Sean Follmer, and Björn Hartmann. 2016. Drill Sergeant: Supporting physical construction projects through an ecosystem of augmented tools. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '16*. ACM Press, New York, New York, USA, 1607–1614. https://doi.org/10.1145/2851581.2892429

[47] Wolfgang Stuerzlinger, Olivier Chapuis, Dusty Phillips, and Nicolas Roussel. 2006. User interface facades: towards fully adaptable user interfaces. In *Proceedings of the 19th annual ACM symposium on User interface software and technology - UIST '06*. ACM Press, New York, New York, USA, 309. https://doi.org/10.1145/1166253.1166301

[48] Danny Sullivan. 2018. A reintroduction to Google's featured snippets. https://www.blog.google/products/search/reintroduction-googles-featured-snippets/

[49] Tamara Sumner and Markus Stolze. 1997. Evolution, not revolution: participatory design in the toolbelt era. In *Computers and design in context*, Morten Kyng and Lars Mathiassen (Eds.). MIT Press, Chapter 1, 1–26. https://dl.acm.org/citation.cfm?id=270319

[50] Juhani E Tuovinen and John Sweller. 1999. *A Comparison of Cognitive Load Associated With Discovery Learning and Worked Examples.* Technical Report 2. 334–341 pages. http://idtoolbox.eseryel.com/uploads/9/0/7/5/9075695/1999-03660-014.pdf

[51] 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

[52] Xu Wang, Benjamin Lafreniere, and Tovi Grossman. 2018. Leveraging Community-Generated Videos and Command Logs to Classify and Recommend Software Workflows. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18.* ACM Press, New York, New York, USA, 1–13. https://doi.org/10.1145/3173574.3173859

[53] Yu Zhong, T. V. Raman, Casey Burkhardt, Fadi Biadsy, and Jeffrey P. Bigham. 2014. JustSpeak: enabling universal voice control on Android. In *Proceedings of the 11th Web for All Conference on - W4A '14.* ACM Press, New York, New York, USA, 1–4. https://doi.org/10.1145/2596695.2596720