Improving Learning Transfer from Stencils-Based Tutorials

Kyle J. Harms
Washington University in St. Louis
One Brookings Dr.
St. Louis, MO 63130
harmsk@cse.wustl.edu

Jordana H. Kerr
The University of North Carolina at Charlotte
9201 University City Blvd.
Charlotte, NC 28223
jwhodge1@uncc.edu

Caitlin L. Kelleher
Washington University in St. Louis
One Brookings Dr.
St. Louis, MO 63130
ckelleher@cse.wustl.edu

ABSTRACT
To support children learning to use new software applications independently, tutorial systems should prevent errors and ensure that users are able to transfer tutorial skills to a new context effectively. In this paper, we describe the formative development and evaluation of on-request stencils, an interaction technique that both prevents children from making errors within a tutorial and significantly improves their ability to transfer tutorial skills to a related task. Using on-request stencils, users can attempt a task independently. If they encounter difficulty, users can request step by step tutorial overlays to guide them through the current task. In a study comparing tutorial performance, task performance, and attitudes, we found that users of on-request stencils successfully completed 47% more transfer tasks than users of persistent stencils. There were no significant differences between the two groups in tutorial performance or attitudes towards the software system.

Categories and Subject Descriptors
H.5.2 [**User Interfaces**]: Training, help, and documentation.

General Terms
Design, Human Factors.

Keywords
Information transfer, tutorials, user interface design, Stencils.

1. INTRODUCTION
To support children learning to use new software applications independently, tutorials that can both prevent errors and facilitate transfer to a related task are particularly important. Existing tutorial support systems can either help to prevent errors or increase task knowledge retention, not both. Additional work is necessary to develop tutorial systems that adequately support both error prevention and transfer.

We conducted a formative study with 46 participants to identify the difficulties surrounding knowledge transfer using an existing tutorial interaction technique, stencils [8], within the novice programming environment Looking Glass [4]. Stencils uses a series of graphical overlays to draw users' attention to needed components and intercepts user interface events over components not needed for the current step. Based on observed problems, we describe how we modified stencils to facilitate transfer.

Additionally, during our formative studies, we observed that modifying stencils-based tutorials to enable users to request stencil help, rather than showing stencils for all steps in the tutorial, significantly improved transfer. A follow up study with 18 users comparing persistent and on-request instructional stencils found that users of on-request stencils tutorial performed 47% better on a near transfer task than users of a persistent stencils tutorial.

IDC 2011, June 20-23, 2011, Ann Arbor, USA.

2. RELATED WORK
Prior work in tutorials systems has explored how to present procedural instructions both outside (e.g. on paper or in a separate window) and within the application context.

2.1 Tutorials Outside of the Application
Many systems present tutorials as textual instructions supplemented by pictures either on paper or in a separate window [2, 3]. Users of these tutorials often struggle to map instructions to the interface, inadvertently skip steps, or mistake pictures of components for active components [10]. Additional research suggests that users learn more quickly with animated tutorials than text only tutorials, but have difficulty retaining the information [11, 12].

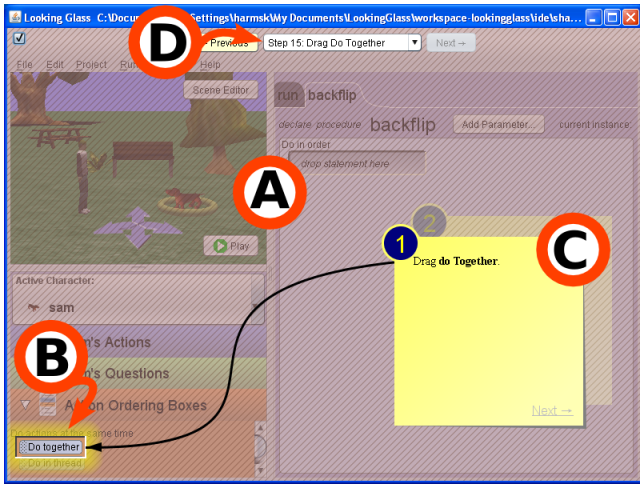

Figure 1. Looking Glass' stencils tutorial. Stencils components: (A) Semi-transparent overlay, (B) holes to the underlying components, (C) notes with instructions, and (D) navigation bar.

2.2 In-context, Interactive Tutorials

Some applications present tutorial information within the context of the application using attention grabbing markings [1], graphical overlays [8], and embedded video [5]. These systems can help to reduce errors [6, 8] or support transfer [5], but not both.

3. FORMATIVE EVALUATION

We conducted a formative study to explore how to better support transfer in stencils tutorials [8]. Stencils are effective at preventing errors, but do not improve transfer over a paper tutorial [8].

3.1 Stencils

The original stencils interface consists of several components: the semi-transparent overlay (Figure 1-A), holes that enable users to interact with components in the underlying interface (Figure 1-B), notes containing information or instructions (Figure 1-C), and a navigation bar (Figure 1-D) [8]. The stencils overlay has two functions: 1) it directs attention to a particular interface area while muting the rest of the interface and 2) the overlay intercepts user events, such as mouse clicks, to prevent user mistakes. Stencil holes allow the user to interact with components in the underlying application that are necessary for the current step. Notes drawn on top of the stencil present relevant information or instructions.

When a user has correctly completed a step in the tutorial, the tutorial automatically advances to the next step; a new note displays instructions for the next action and the overlay highlights the components needed for that action. Users cannot advance before the current step is complete and correct.

3.2 Looking Glass

We implemented stencils within Looking Glass (see Figure 1) [4], a programming environment for middle school children and the successor to Storytelling Alice [9]. Looking Glass users construct programs by dragging and dropping graphical tiles and selecting parameter values using menus. Because many of our target users lack access to computing domain experts, the ability to successfully transfer knowledge to a new context is particularly important.

3.3 Participants

We recruited our participants for this study from among the visitors to the Saint Louis Science Center. The Saint Louis Science Center has a broad range of visitors from many different socioeconomic backgrounds. In total we recruited 46 participants (20 female and 26 male) with ages ranging from 10 to 16 years of age. None of the participants had prior programming experience. We gave all participants a $5 museum gift certificate in acknowledgment of their participation.

3.4 Materials

We created a stencil-based tutorial and a transfer task.

3.4.1 Tutorial

In order to provide a motivating context, we structured the tutorial around a story of a man, Dave, teaching his dog, Sam, to perform back-flips for a competition. Over the course of the tutorial, users create a new method, add basic method calls to the new method, and call the new method from the program's main method. The tutorial is divided into thirteen sections and takes approximately ten minutes to complete. See Figure 2 for the completed backflip tutorial program.

3.4.2 Transfer Task

In order to determine how well participants learned from the tutorial, we designed a transfer task. We asked participants to make a bunny hop three times sequentially, starting from an empty program containing a bunny object. This task is similar to the back-flip from the tutorial. See Figure 2 for a completed transfer task.

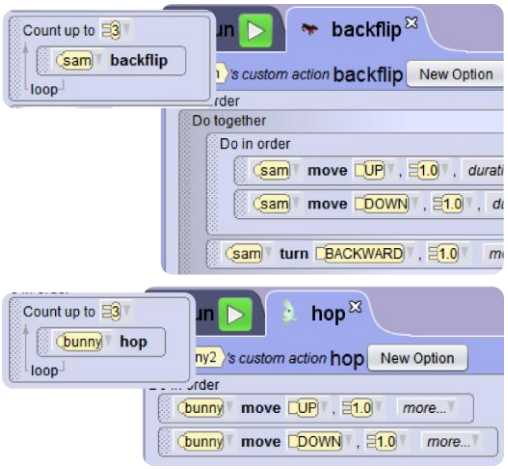

Figure 2. Completed tutorial program: Backflip. Complete transfer program: Hop.

3.5 Methods

We conducted our formative evaluation in a series of single participant, thirty minute sessions. Participants completed the tutorial and then attempted the transfer tasks while thinking aloud. They could refer to the tutorial at any time.

3.6 Data

Throughout each session, we recorded field notes about each participant's errors, points of confusion, behavior, and comments. We also saved participants' transfer programs.

3.7 Lessons Learned

We learned several lessons about how to support users in transferring tutorial learning to new contexts.

3.7.1 Enable users to move among completed tutorial steps.

In early user tests, we observed that users frequently did not absorb all of the material while completing the tutorials. Unfortunately, to refer back to the tutorial, users needed to re-do all of the steps leading up to the material they wanted to revisit, something few were willing to do. In response, we added support that allows users to move directly to any tutorial step as long as they have previously completed that step. Note that this requires setting the appropriate state for both the user's artifact and the state of the interface for each step in the tutorial. Because users often focus on notes as they progress through the tutorial, we moved the navigation bar into the note (see Figure 3-D).

3.7.2 Use task based questions as an answer to the user's own internalized questions

After implementing the changes to allow users to freely navigate the tutorial, we observed that many users struggled to make connections between tutorial topics and transfer tasks. To help users understand how tutorial steps relate to potential goals, we changed the step titles in the navigation bar to ask a question related to the step. For example, we changed the title of "Repeat three times" to "How do we make Sam back-flip three times?" In subsequent sessions, users were more successful using the navigation bar to find relevant steps.

Figure 3. The same tutorial step in a (A) Persistent and (B) On-Request stencils tutorial with (C) invisible hotspot. Further interface changes: (D) Navigation bar inside of note, (E) Show Me button, and (F) Cursor feedback.

3.7.3 Enable users to request help when they need it, otherwise let them work independently

During user tests, several users expressed frustration with following stencil overlays to complete tasks they were confident they could complete without help. In response, we explored allowing users to attempt to complete tutorial steps independently. We modified stencils to display a single note with a high level goal and remove the visual affordance of the overlay. Users could request step by step overlays and instructions when necessary using a *Show Me* button in the note (see Figure 3-E). These changes seemed to reduce frustration and increase transfer task success.

4. SUMMATIVE EVALUATION

To further explore the impact of on-request instructional information on users' tutorial experiences and performance, we ran a pilot study to compare the performance and attitudes of users who always see instructional information (persistent stencils) and users who can request instructional information when needed (on-request stencils).

4.1 Persistent Versus On-Request Stencils

Persistent stencils (control) offers instructional information in two ways: 1) a visual overlay that provides visual cues about which components are used in a step and 2) detailed instructions that guide users through completing a tutorial step, action by action. For example, a single step may ask users to click on a tab or drag a graphical tile. As users progress through actions, the text in the instructional notes updates to describe how to perform the next action.

In contrast, on-request stencils (experimental) removes all detailed step-by-step instructional notes. No visual overlay is displayed, however invisible hotspots of clickable components remain (see Figure 3-C) and a single note describes a high level goal, but contains no information on how to accomplish it. This note remains static throughout the task. For example, in Figure 3, the note directs users to "Add sam's turn backwards 1.0 action to the beginning of the do together." Using the "Show Me" button, users of on-request stencils can bring up persistent stencils for the current step.

The only difference between the two systems lies in the presentation of instructional information. Users perform the same sequence of steps in both tutorials. If users click on a component not needed in the current step, the cursor changes to indicate that it is inaccessible (see Figure 3-F).

4.2 Participants

We recruited 19 participants from among the visitors to the Saint Louis Science Center. We screened out participants with prior programming experience. However, during testing we discovered that one participant had previous programming experience. Excluding this participant, we had 18 participants (7 female, 11 male) with ages ranging from 10 to 16 years (12.4 average, 1.7 standard deviation). Each participant received a $5 museum gift certificate in recognition of their participation.

4.3 Methods

We randomly assigned users to either the control or experimental tutorial. Participants completed their assigned tutorial, attempted the same transfer task as the formative evaluation, and finally completed an attitude survey. Afterwards, we gave participants the option to continue using Looking Glass independently to gauge their interest in continuing to use the software.

4.4 Data

We collected timing information for the tutorial and the transfer task. We also recorded whether and when on-request users requested step by step stencils guidance.

4.4.1 Transfer Task

We scored participants' transfer tasks out of five points:

1. The user declared a hop method.
2. The user invoked bunny.move(UP) in hop.
3. The user invoked bunny.move(DOWN) in hop.
4. The user invoked hop in the program main.
5. The user used a loop to execute the hop method three times.

See Figure 2 for a successfully completed transfer task.

4.4.2 Attitude Survey

The attitude survey included eleven Likert scale items focusing on users' enjoyment, interest, and confidence when using Looking Glass. Participants rated their agreement from strongly disagree (-2) to strongly agree (2).

Table 1. Control and Experimental Performance.

Measure	Control (Persistent)	Experimental (On-Request)
Avg. (sd) Tutorial Completion Time	9 min. (s=2.2)	11 min. (s=2.7)
Avg. (sd) Transfer Tasks Correct	3.1 tasks (s=1.7)	4.6 tasks (s=1.0)
Avg. (sd) Time per Correct Transfer Task	9 min. (s=3.6)	7 min. (s=2.5)

4.5 Summative Results

We summarize our results in two categories: tutorial and transfer performance and attitudes and behavior.

4.5.1 Tutorial and Transfer Performance

Users of on-request stencils correctly completed 47% more transfer tasks than users of persistent stencils ($t(16)=2.2;p<.05$) (see Table 1). Users of on-request stencils completed both the tutorial ($t(16)=1.73;p=.1$) and the transfer tasks ($t(16)=-1.54;$ $p=.1$) more quickly, although the differences are not significant.

4.5.2 Attitudes and Behavior

We examined user experience through an attitude survey and behavioral measures.

We used a one way MANOVA to analyze the results of the attitude survey. We found no significant difference in attitudes towards Looking Glass between users of on-request and persistent stencils ($F(11,6)=1.12$; $p=.47$).

The choice to continue using Looking Glass independently is a potential indicator of interest in the system. After participants completed their attitude surveys, a researcher thanked participants and presented the option to continue using Looking Glass on their own. In total 6 of the 9 on-request tutorial users elected to continue working with Looking Glass compared to 2 of the 9 control users. This difference is marginally significant ($t(16)=2;p=.06$).

5. DISCUSSION

A recent study found that testing helped college students retain 50% more material from a scientific passage than students who studied or created concept diagrams of the material [7]. On-request stencils may be tapping into a similar phenomena. In essence, an on-request stencil describes a goal for the user to complete but does not provide information about how to complete the task. Users typically pause and think about how to complete the task and may visually search for potentially relevant interface elements. At this stage, if a user then requests help, then that user may be better prepared to relate the full help steps presented to the higher level goal presented in the task. In contrast, we observed that users of persistent stencils appear to simply "click" through the tutorial rather than reflect on the high-level goal between steps.

This paper reports on a small, single session study within the context of Looking Glass. While we believe that on-request stencils may help users to learn a variety of graphical user interfaces, further research is necessary to explore the effectiveness of on-request stencils in other applications.

6. CONCLUSION AND FUTURE WORK

Tutorials are a common way for children to explore new software applications. Providing support that enables users to attempt tutorial steps independently and request help when needed appears to increase performance on transfer tasks, without negatively impacting attitudes towards the application. While we chose to focus on the use of on-request stencils in a programming environment, we believe that on-request stencils will be helpful for children using a wide range of GUI applications.

On-request stencils currently provide two levels of support: users can perform a task independently or complete that task using a step by step guide. Future work should explore finer grained support through mechanisms like hints or the ability to skip familiar material.

7. ACKNOWLEDGMENTS

This work was funded by NSF grant #0835438. We would like to thank the Saint Louis Science Center for their time and the use of their facilities.

8. REFERENCES

[1] Coachmarks: http://www.developer.apple.com/techpubs/mac/AppleGuide/AppleGuide-24.html.

[2] Goodall, S.D. 1991. Online help in the real world. Proceedings of the 9th annual international conference on Systems documentation (Chicago, Illinois, United States, 1991), 21-29.

[3] Goodall, S.D. 1992. Online help: a part of documentation. Proceedings of the 10th annual international conference on Systems documentation (Ottawa, Ontario, Canada, 1992), 169-174.

[4] Gross, P.A. et al. 2010. A code reuse interface for non-programmer middle school students. Proceeding of the 14th international conference on Intelligent user interfaces (Hong Kong, China, 2010), 219-228.

[5] Grossman, T. and Fitzmaurice, G. 2010. ToolClips: an investigation of contextual video assistance for functionality understanding. Proceedings of the 28th international conference on Human factors in computing systems (New York, NY, USA, 2010), 1515–1524.

[6] Huang, J. and Twidale, M.B. 2007. Graphstract: minimal graphical help for computers. Proceedings of the 20th annual ACM symposium on User interface software and technology (New York, NY, USA, 2007), 203–212.

[7] Karpicke, J.D. and Blunt, J.R. Retrieval Practice Produces More Learning than Elaborative Studying with Concept Mapping. Science.

[8] Kelleher, C. and Pausch, R. 2005. Stencils-based tutorials: design and evaluation. Proceedings of the SIGCHI conference on Human factors in computing systems (Portland, Oregon, USA, 2005), 541-550.

[9] Kelleher, C. et al. 2007. Storytelling alice motivates middle school girls to learn computer programming. Proceedings of the SIGCHI conference on Human factors in computing systems (San Jose, California, USA, 2007), 1455-1464.

[10] Knabe, K. 1995. Apple guide: a case study in user-aided design of online help. Conference companion on Human factors in computing systems (Denver, Colorado, United States, 1995), 286-287.

[11] Palmiter, S. and Elkerton, J. 1991. An evaluation of animated demonstrations of learning computer-based tasks. Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology (New York, NY, USA, 1991), 257–263.

[12] Palmiter, S. et al. 1991. Animated demonstrations vs. written instructions for learning procedural tasks: a preliminary investigation. International Journal of Man-Machine Studies. 34, (May. 1991), 687–701.