

Designing Tools to Support Multimedia Authoring By Incorporating Problem-Solving Strategies

Yasushi Akiyama¹, Sageev Oore²

¹Dalhousie University, Faculty of Computer Science
6050 University Avenue, Halifax, Nova Scotia, Canada B3H 4R2
sageev@cs.smu.ca

²Saint Mary's University, Department of Mathematics and Computing Science
923 Robie Street, Halifax, Nova Scotia, Canada B3H 3C3
yasushi@cs.dal.ca

Abstract- We have witnessed rapidly increasing interest in multimedia authoring due to the easy access to various devices such as tablets and mobile phones to view and create multimedia data. While many authoring tools are historically designed for advanced users, we started to see more introductory tools that allow novice users to perform authoring tasks by providing simplified interfaces and using template-based approaches to bypass complex steps of the authoring tasks. Our design approach presented here differs from these tools in that it complements authoring tasks by supporting some strategies used in problem-solving and providing the contextual information of the task. The discussion of possible implementation is given based on this approach. The initial evaluation suggests that our approach potentially provides benefits that are not usually present in standard multimedia authoring tools.

Keywords: Multimedia authoring tools, interface designs, problem-solving for multimedia authoring, cognitive models.

© Copyright 2014 Authors - This is an Open Access article published under the Creative Commons Attribution License terms <http://creativecommons.org/licenses/by/3.0>. Unrestricted use, distribution, and reproduction in any medium are permitted, provided the original work is properly cited.

1. Introduction

With the emerging technology of multimedia devices, we are surrounded with digital media – TV/radio shows, movies/videos, and music are examples of common multimedia formats that we encounter on a daily basis. Multimedia data are usually created with specialised authoring tools regardless of

whether they are large-scale Hollywood films or home videos of a trip to the Grand Canyon. While many of these authoring tools traditionally offer advanced features required by experts, in recent years, we started to see more introductory tools targeting less advanced users. These applications offer simplified ways to author multimedia content; for example, wizard- or template-based authoring tools, often seen in mobile applications, allow users to bypass complex steps using predefined parameters and workspace templates while others simply omit or hide advanced features that are too difficult for novices to use and learn. These tools usually look simpler but often do not allow users to create and modify fine details of the media. Our approach presented in this paper also attempts to support novice users of multimedia authoring tools. It differs from the aforementioned approaches, however, in that we consider multimedia authoring tasks as a type of problem-solving and task-planning processes, and attempt to support users by incorporating an approach that is grounded on the cognitive models originally developed in the context of task-planning and problem-solving. We will illustrate how the proposed approach might translate to practical examples and suggest possible support features for existing authoring tools.

1.1. Motivation

Not only does multimedia authoring require knowledge and skills of creative tasks, but it is also inevitable for users to possess thorough knowledge of digital data manipulation. Studies [1, 2, 3, 4] show that designing creativity support tools is still difficult. Based

on the related literature [5, 6, 7] and our own general observation of users performing multimedia authoring tasks, novices and experts exhibit behaviours that can be summarized with respect to Norman's classic "Seven stages of actions" [8]. Common problems novices encounter when performing multimedia authoring tasks are well described by referring these stages, and experts tend to display superior performance at each stage. Consequently, when using the same tools, experts can effectively minimize the gulfs of both execution and evaluation due to their familiarity with the tasks and the tools. Furthermore, the ability to come up with proper goals and to find ways to reach the goals are essential for generating mental representation of a task, and generating a proper mental task model is one of the critical processes for successful task performance [7]. Supporting users generate and maintain proper representation of a task can be therefore beneficial especially when they have little experience with multimedia authoring tasks.

2. Preliminary Studies

2.1. User Observation, Interviews, and Surveys

From the early general observation of users, multimedia authoring tasks seem to share certain characteristics of problem-solving and task-planning, and therefore users may benefit from tools that support them perform complex multimedia authoring tasks in such a way to accommodate strategies used in problem-solving. To further reinforce this premise, we have conducted additional user observation, interviews and online surveys with professionals and amateurs (both advanced and novice users). The purpose of these preliminary studies was twofold: 1) to understand the ways advanced and novice users prepare for and perform multimedia authoring tasks and how they differ, and 2) to see how task-planning and problem-solving strategies might help these users in the context of multimedia authoring tasks. The settings of each of the preliminary studies are as follows: for user observation, we had two novice users with a little experience in home video editing but less than 1 year, performing video editing using Adobe Premiere and Windows Movie Maker respectively, and one advanced video editor working on a short-film using Apple Final Cut Pro and Adobe Premiere were observed; we interviewed five professional and two amateur video editors/sound engineers were contacted for interviews with regards to their typical workflow, their opinions/concerns about existing tools; eight novice

users responded to online surveys on multimedia authoring tasks and issues with tasks and existing tools.

2. 2. Key Findings from the Preliminary Studies

The findings from these preliminary studies suggest that the multimedia authoring processes do in fact share some of the characteristics of task-planning and problem-solving processes, and thus they seem to support our initial premise. They also suggest possible designs and features of multimedia authoring support tools, which we will discuss in details in Section 4.

One salient finding was that concrete workflow was typically formed by experts while novices had problems with coming up with one. This was observed in all the stages of the studies, but particularly in the user observation, novices started tasks in a more exploratory manner, but advanced users were able to plan ahead what needed to be done. Another key finding was that novices did not seem able to easily find a set of actions to escape from unwanted situations. For instance, when a desired action was unavailable, novices often could not know what actions were needed to be performed to make those desired action performable. And for another example, one of the respondents of the survey noted that "the biggest obstacle was that it was hard to think creatively because I was just unaware of the options." These examples illustrate that even when novice users know what they want to accomplish, they may still have problems with actually figuring out *how* to accomplish it.

Many novice users also expressed that they would like to be able to see the overview of the task that they were working on. This is understandable as existing tools usually only provide undo lists that do not provide much useful information on task context. While advanced users may be able to mentally maintain task context, it can be difficult for those who lack experience in multimedia authoring tasks.

3. Related Work

Although, to our knowledge, there is no previous research conducted to develop a multimedia authoring support tool that attempts to enable application of problem-solving strategies, several approaches to support novice users during multimedia authoring tasks have been investigated. One obvious approach is to reduce the complexity of the interface design used in existing tools. This approach is often seen in commercial products such as [9, 10, 11, 12, 13], and in research projects such as [14], which reduce the complexity of its interface based on the context of the

task. Another approach seen in quite a few commercially available products, especially in mobile apps, is to use the template- and wizard-based approach. The examples include [15, 16, 17]. These tools first ask the user to import/select media files that they would like to use such as videos, still images, and music/sound track(s). The users will be then asked to select a theme or style of the final product, and then the tools will automatically create appropriate media based on the predefined parameters. Another group of tools [18, 19, 20] use an approach to replace existing interface paradigms and provide users with unique interface designs.

Our design approach presented in this paper clearly differs from these tools in that it complements authoring tasks by supporting some strategies used in problem-solving and providing the contextual information of the task.

Furthermore, while not within creative domains and their main focuses are not to support novice users, there are studies that attempt to model tasks and task structures, and to visualize those using unique notation and visualization methods [21, 22, 23, 24].

4. Multimedia Authoring and Problem-Solving

In this section, we will elaborate the above premise by first analysing general multimedia authoring tasks and tools, and then presenting the relevant theories of problem-solving and task-planning primarily developed in the fields of Cognitive Psychology and Artificial Intelligence.

4.1. Multimedia Authoring Tasks and Authoring Tools

The current style of multimedia authoring and editing has emerged from the early days of film and sound production devices, and techniques developed with those devices [25]. The interface of software tools have been designed so that the authoring styles and techniques used with the hardware counterparts can be closely emulated in digital environment and thus, the appearance of typical tools is extremely similar to the look-and-feel of the corresponding hardware devices. Figure 1 shows screenshots of common multimedia authoring tools.

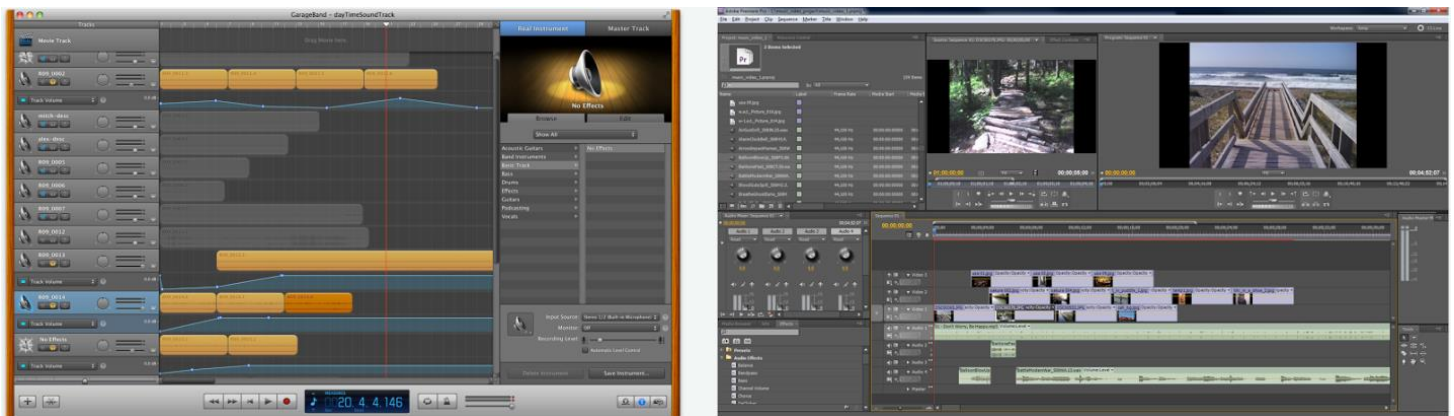


Figure 1. Screenshots of popular authoring tools Apple GarageBand (left) and Adobe Premiere (right).

While in a broad sense, multimedia authoring tasks can refer to creation and editing of digital media in many different formats, we focus on two of the quintessential tasks; video editing and music/sound production. Through these tasks, we author the majority of multimedia content we encounter in daily life. These tasks share some key elements such as creation of media that have a temporal element so that they can be played back, dealing with one or more data streams that can be simultaneously played back, and structuring one or more data clips for appropriate data

streams (called tracks) along the time axis to specify the timing of the clips to be played.

Multimedia authoring requires knowledge and skills derived from both creative tasks and digital data manipulation. In multimedia authoring, proficiency in generating creative ideas and working towards ideal goals becomes practical only when complemented by the familiarity with the tools that translate the creative ideas into digital media. It is not uncommon therefore to see artists and creators often struggle when transitioning to software tools. Furthermore, while people with experience in general creative tasks may

find at least some critical skills transferable to the digital authoring, novices seem to find it challenging to progress to reach a certain level of proficiency.

4.2. Creative Tasks and Problem-Solving Strategies

Creative tasks typically involve processes that are the key aspects of the problem-solving, such as identifying goals, assessing the current situation, recognising available actions, and pursuing the chosen actions (or modify them accordingly) towards the desired goals. When attempting to solve complex problems such as creative tasks, humans often use heuristics and strategies to either reduce the complexity of the problem and/or systematically undertake only smaller parts of the problem. The following are cognitive models of these strategies.

The top-down approach [26, 27, 28, 29] is often referred to as problem-reduction or task-reduction, and its basic idea is to first break down the original problem into smaller subproblems and then recursively apply the reduction until each sub-problem becomes trivial enough to be solved directly without further reductions. The bottom-up strategy [6, 28, 29, 30, 31] is triggered by lower-level, detailed information, and is often regarded as a data-driven approach. In this strategy, the task performer starts with focused lower-level actions/subtasks that directly operate on/for given data, and then the future actions will be planned and performed by gradually working towards more abstract and contextual solutions of the task. The opportunistic strategy [32, 33, 34] is modelled after a much more unstructured way that humans generally perform task-planning and problem-solving. The idea behind this cognitive model is that humans often make problem-solving decisions spontaneously rather than strictly in the previously planned manner, thus both the top-down and bottom-up strategies are employed throughout the task. The task performer more freely shifts the focus of the abstraction and hierarchical levels of the task-space.

5. Towards Designing Multimedia Authoring Task Support Tools

To facilitate the designing process of support tools for multimedia authoring tasks, we will now discuss how the theories of task-planning and problem-solving may be applied to multimedia authoring tasks.

5.1. Problem Solving Strategies and Multimedia Authoring

In the previous section, three major strategies have been identified, and we will discuss here some of the most common cases in which these strategies are invoked in multimedia authoring tasks. The top-down strategy is used when attempting to solve complex problems by first setting the overall goal, and then dividing it into smaller, more manageable sub-goals. In multimedia authoring tasks, a user may start the task by sketching out a very rough idea of the kinds of media that she is creating; e.g., 30-second TV commercial. She may then determine what media materials are needed, given the basic ideas of what the medium is supposed to deliver to the audience. From there, she can look for particular media, think about how these media should be edited, and so on, thus working towards more detailed parts of the task. On the other hand, the bottom-up strategy is motivated by (new) incoming data at a detailed level that triggers some changes in the higher level in the hierarchy. For example, suppose the video editor finds a new media clip that is more suitable for the current project. She might then tentatively replace the old clip with this new one, which consequently forces a sequence of operations in order to accommodate this new clip into the current workspace such as adjusting the lengths and changing several parameters of both this new and the other existing clips. That is, a sequence of actions are triggered by this new data clip and thus it forces the user to “go up” the hierarchy levels to examine more contextual information of the task in order to perform other necessary actions. The opportunistic strategy can be considered as a combination of both the top-down and the bottom-up strategies and it provides more flexibility in terms of the ways users perceive and perform tasks. To some extent, with existing tools, users perform the opportunistic strategy as these tools are designed to accommodate the most flexible styles of authoring processes typically required by advanced users.

One of the issues of the standard authoring tools is that they usually lack explicit support for viewing the task structures. While it may not be a big problem for experienced users, it is generally difficult for novices to correctly generate and maintain the mental representation of the task, especially when the task progresses and the performed actions and their hierarchical and sequential relationships become too complex to be handled by humans. This could cause

problems even for advanced users when they need to revisit their past actions to modify what they have done, or more commonly, when they resume their task after a long period of inactivity from the task. Task interruptions occur frequently during a multimedia production that sometimes lasts for months. Another advantage that existing tools may gain from explicit support for viewing the task structures is that it clarifies the context of the task and it will therefore likely help novice users to be more aware of what their short- and long-term goals are. We will elaborate these issues in the following section.

5.2. Implication of Interface Designs

As existing tools do not usually provide explicit support for viewing structured tasks and actions, the available resource closest to this kind of support is the undo list, which is a linear sequence of performed actions, and, to some extent, the menu structures which group items based on the types of actions. For example, the menu entry *File* may have such items as *Open*, *New*, and *Save*, all of which are actions that involve file operations. While this is one way to create a hierarchical structure of the task, this organisation is seriously limited when it comes to reviewing tasks. Consider the following case in which we are trying to organise the undo list of actions by grouping them using the parent menu labels of these actions. Figure 2.a. shows a partial list of actions performed during a video production task by the authors. One way we could build a hierarchical structure of this task-space is to group actions under menu labels whenever one or more consecutive actions are from the same menu label. This is shown in Figure 2.b.

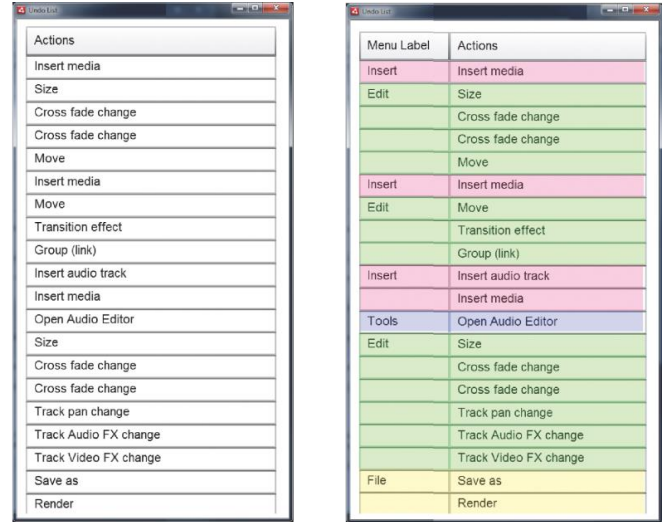


Figure 2. (a) Action list (left) and (b) Action list with Menu Labels (right).

However, neither this list of the menu labels (the left column of Figure 2.b.) nor the original list of low-level actions (Figure 2.a.) tells us exactly what kinds of sub-tasks were performed. Furthermore, as the task progresses further, this list of the menu labels only gets longer and it will become even harder to decipher what types of sub-tasks were performed at different stages of the task. Therefore, we need a slightly different approach for grouping and organising actions and sub-tasks for our purpose. What is missing from the above lists is the information about the context of the task. To illustrate this, let us use descriptions of sub-tasks to group the list of actions. The result of this new labelling is shown in Figure 3.

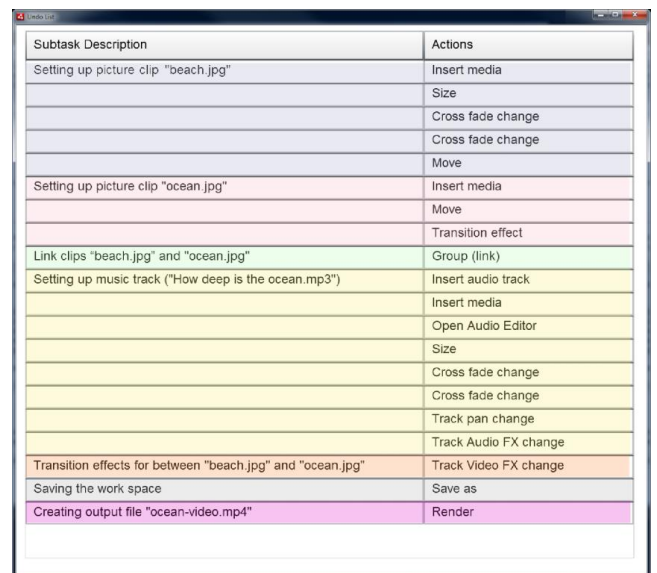


Figure 3. Action list with Sub-task Headings.

Using these sub-task descriptions, the new list will look contextually much clearer (*Setting up picture clip "beach.jpg" → Setting up picture clip "ocean.jpg" → Link clips "beach.jpg" and "ocean.jpg" → Setting up music track ("How deep is the ocean.mp3") → Transition effects for between "beach.jpg" and "ocean.jpg" → Saving the workspace → Creating output file "ocean-video.mp4"*).

With this slight change to the way in which the task is organised, our list of past actions becomes much more contextually meaningful, and thus it will presumably increase the understanding of the task structures. When users need to use a top-down approach, the task-space representation can focus on these high-level sub-task descriptions, and when she

proceeds down the hierarchical levels to investigate the details of the task-space, or when a bottom-up strategy should be triggered, the tool can reveal these lower-level actions, as shown in Figure 4. A key observation here is that in addition to reviewing past actions, this organisation can also be applied to possible future actions. For example, if a user is performing a subtask labelled "Setting up picture clip 'beach.jpg,'" then instead of menus and toolbar icons which typically show all the available actions, it might be more effective for the system to allow users to see and perform only those options that are relevant to the current subtask such as "modify the picture clip" or "apply visual effects."

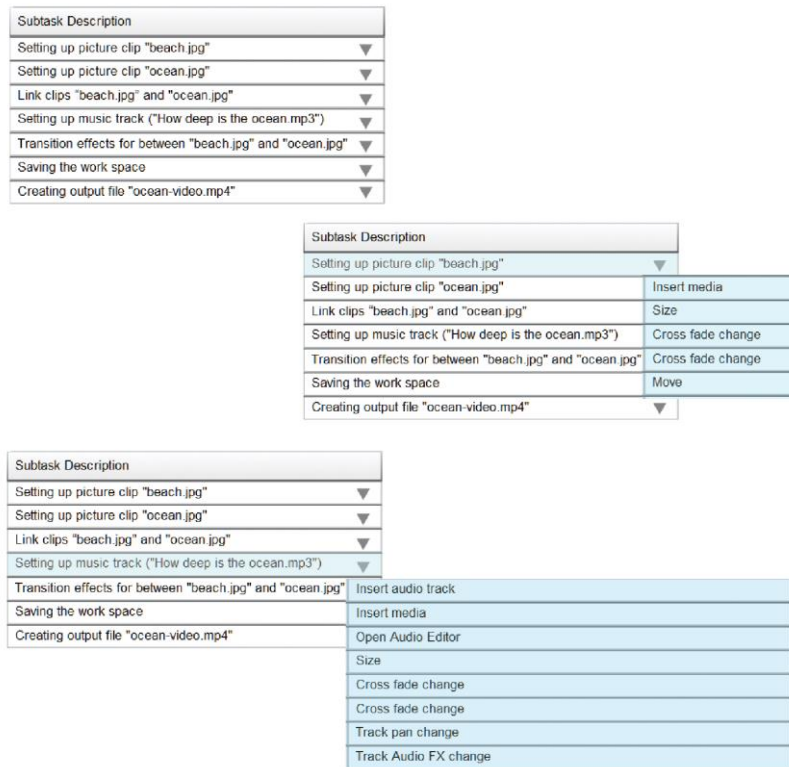


Figure 4. Sub-task Headings with partially revealed actions.

6. Paper-based Mock-up and Cognitive Walkthroughs

Prior to a large-scale evaluation in user studies, some of the possible feature implementations of our approach were evaluated using the paper-based mock-up and by means of basic cognitive walkthroughs. The evaluators were given steps to create a story-telling video from a set of raw media as described in Table 1. The evaluation was based on several versions of the

mock-up, differing features such as the organisation schemes and representations of the action groups as discussed in Section 4. A few samples of the mock-up are shown in Figure 5.

Although the process was primitive, the results of this evaluation process corroborated to our preliminary analyses and illustrated some potential issues that may be addressed in future implementations. As expected, the additional information on the task context, which

can be provided by the structured task-space, helped find target actions easily even when the user was not entirely familiar with the editing tool's menus or toolbar icons. Choosing from the list of a limited number of actions that were only relevant to the current sub-task seemed easier than trying to find the same action from the large pool of actions such as menus and the list of icons. One of the issues that were

observed in the preliminary studies was that novices were unable to find a proper set of actions to escape from unwanted situations. By reducing the number of possible actions by classifying them into contextually more meaningful groups, the support tool can presumably decrease the chance that these users may try to include inappropriate actions.

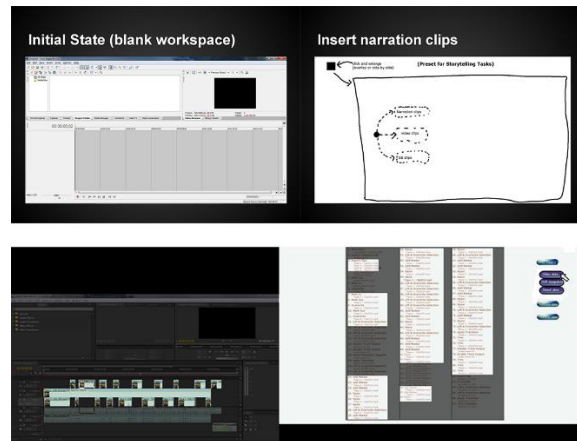


Figure 5. Some versions of paper mock-up used in the cognitive walkthrough.

The abstracted view of the task-space seemed helpful for following the overview steps of the task (shown in Table.1) smoothly while the grouped actions were useful when the focus was shifted to more local-level task planning. These results illustrate how this support tool potentially supports some characteristics of problem-solving strategies. For example, overview of the task corresponds to an abstract perspective typically employed at the initial stage of the top-down strategy, while the grouped actions can provide a local

level perspective which is needed for the bottom-up strategy. By allowing the different focus levels, the interface could be used to accommodate a more general approach of the opportunistic strategy. In addition to the structured task views, indicating the connection between the task-space and the data in the workspace seemed to improve the understanding of the task being performed.

Table 1. List of the provided media and the task procedures.

Provided media set	Basic steps
1) Narration clips 2) Video clip with no sound (some needed to be trimmed, sliced, or grouped) 3) Sound effects	1) Initial state (blank workspace) 2) Insert narration clips in order 3) Choose a video clip 4) Place the video clip at an appropriate location 5) Trim/slice to extract the desired portion of the video 6) Readjust the location of the video clip 7) Repeat steps 3-6 to have enough number of video clips to go with the narration clips 8) Choose a sound effect (SE) clip 9) Place the SE clip at an appropriate location 10) Adjust the length of the SE clip 11) Readjust the SE clip's location 12) Adjust the volume of the SE clip 13) Repeat steps 8-12 to add enough number of SE clips 14) Render

Although there are some limitations of this style of evaluation, they will likely lead to design ideas for more comprehensive future studies. One such limitation is that the comparison of the support system was against the earlier observation of users performing arbitrary tasks. More systematic comparison of these tools on tasks with the same complexity levels may presumably expose the effect of this support tool. Another limitation is that this evaluation could not provide the insight for individual effects of implemented features. In other words, we were only able to evaluate the support tool as a whole, but were unable to isolate effect of each feature. Testing each feature of the support tool will help us analyze which feature will be most effective and suitable for which situation(s) in multimedia authoring tasks.

We have also encountered some issues that will need to be addressed in future implementations. Most versions of the mock-up used only two colours (black and white), but it would likely increase the understanding of the task if some sort of consistent colouring schemes were used for visualised objects. When using the sub-task headings, longer headings were not very clear at a first glance, and it could be even worse if two or more long headings share many words (e.g., "Prepare and layout picture clip 'sakura-1.jpg' " and "Prepare and layout picture clip 'sakura-2.jpg' "). A better naming convention and/or different ways to represent objects should be sought, and the consistent colouring schemes might possibly help to solve this issue as well.

7. Conclusion

In this paper, we have identified several issues when novice users try to perform complex multimedia authoring tasks by conducting the preliminary user studies. These studies consisted of the observation of current users performing authoring tasks, interviews with advanced users, and the online survey with novice users with regards to the general workflow of authoring tasks and some difficulties that they face when attempting to learn and use these tools to perform authoring tasks. The issues identified from these user studies were found to be closely related to the issues when humans perform problem-solving and task-planning activities.

By breaking down the characteristics of multimedia authoring tasks and comparing with the theories of problem-solving and task-planning, we have shown that

the application of these theories for designing support tools for multimedia authoring may open up the door to new users to learn and perform complex tasks of multimedia authoring. The discussion of possible designs to be integrated into existing tools is also provided.

We have also conducted the cognitive walkthroughs using the paper mock-up to evaluate some features of our approach. The results of the evaluation process conformed to our preliminary analyses and illustrated some potential issues that may be addressed in future implementations.

Acknowledgement

This research is funded with the support of NSERC and CFI. The authors would like to thank Carolyn Watters and Derek Reilly for their invaluable feedback, and also the anonymous reviewers for their very helpful comments that improved the quality of the manuscript significantly.

References

- [1] H. Johnson and L. Carruthers "Supporting creative and reflective processes" *International Journal of Human-Computer Studies*, 64(10), 2006, pp. 998-1030.
- [2] B. Shneiderman "Creativity support tools: A grand challenge for HCI researchers" *Engineering the User Interface*, 2009, pp. 1-9. Springer London.
- [3] G. Turner and E. Edmonds "Towards a supportive technological environment for digital art" in Viller & Wyeth, *Proceedings of OzCHI Asia-Pacific Computer Human Interaction Conference*, 2003, pp. 44 -51.
- [4] Y. Yamamoto and K. Nakakoji "Interaction design of tools for fostering creativity in the early stages of information design" *International Journal of Human-Computer Studies: Special Issue on Computer Support for Creativity*, 63(4-5), 2005, pp. 513-535.
- [5] S.T. Bulu and S. Pedersen "Supporting problem-solving performance in a hypermedia learning environment: The role of students' prior knowledge and metacognitive skills" *Computers in Human Behavior*, 28, 2012, pp.1162-1169.
- [6] J.-M. Hoc "Cognitive psychology of planning" Academic Press Professional, Inc., 1988, San Diego, CA, USA.

- [7] G. Ward and R. Morris "Introduction to the psychology of planning" *The cognitive psychology of planning, Current issues in thinking & reasoning*, 2005, Chapter 1. Psychology Press.
- [8] D. A. Norman "The Design of Everyday Things" 2002, Basic Books.
- [9] Apple. (last visited 2014). GarageBand. <http://www.apple.com/ilife/garageband/>.
- [10] Apple. (last visited 2014). iMovie. <http://www.apple.com/ilife/imovie/>.
- [11] FlexiMusic Kids Composer. (last visited 2014). <http://www.fleximusic.com/>.
- [12] Microsoft. (last visited 2014). Movie maker. <http://windows.microsoft.com/en-ca/windows-live/movie-maker>.
- [13] Sony Super Duper Music Looper. (last visited 2012). <http://www.sonycreativesoftware.com/>.
- [14] B. Lafreniere, A. Bunt, M. Lount, F. Krynicki and M. Terry "AdaptableGIMP: Designing a Socially-Adaptable Interface" *Proceedings of Symposium on User Interface Software and Technology (UIST '11)*, 2011.
- [15] Magisto. (last visited 2014). <http://www.magisto.com/>.
- [16] Muvee. (last visited 2014). <http://www.muvee.com/en/>.
- [17] Vidify. (last visited 2014). <http://www.vidifyapp.com/vidify>.
- [18] Y. Akiyama and S. Oore "PlaceAndPlay: a digital tool for children to create and record music" *Proceedings of SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*, 2008, pp 735-738.
- [19] M.M. Farbood, E. Pasztor, and K. Jennings "Hyperscore: A graphical sketchpad for novice composers" *IEEE Emerging Technologies*, 2004, pp. 50-54.
- [20] T. Fischer and W. Lau "Marble track music sequencers for children" 2006, *Proceedings of the 2006 conference on Interaction Design and Children*, 141-144.
- [21] S. Balbo, N. Ozkan, and C. Paris "Choosing the Right Task Modelling Notation: A Taxonomy" D. Diaper, editor, *The Handbook of Task Analysis for Human-Computer Interaction*, 2004, pp.445-465. Lawrence Erlbaum Associates.
- [22] G. Bieber and C. Tominski "Visualization Techniques for Personal Tasks on Mobile Computers" In *HCI International '03*, 2003, pp. 659-663
- [23] A. Bruno, F. Paterno, and C. Santoro "Supporting Interactive Workflow Systems Through Graphical Web Interfaces and Interactive Simulators" *International Workshop on Task Models and Diagrams for user interface design (TAMODIA '05)*, 2005, pp. 63-70.
- [24] S. Lu, C. Paris and K. Vander Linden "Tamot: Towards a Flexible Task Modeling Tool" *Proceedings of Human Factors*, 2002.
- [25] M. Davis "Editing out video editing" *IEEE MultiMedia*, 2003, 10(2), pp. 54-64.
- [26] M.L. Gick "Problem-solving strategies" *Educational Psychologist*, 21, 1986, pp. 99-120.
- [27] R. Jeffries, A.A. Turner, P.G. Poison, and M.E. Atwood "The processes involved in designing software" *Cognitive Skills and Their Acquisition*, 1980, pp. 255-283.
- [28] E. Nyamsuren and N.A. Taatgen "Top-down Planning and Bottom-up Perception in a Problem-solving Task" *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, 2011, pp. 2685-2690.
- [29] E.D. Sacerdoti "Planning in a hierarchy of abstraction spaces" *Artificial Intelligence*, 5, 1974, pp. 115-135.
- [30] A. Barrett and D.S. Weld "Task-decomposition via plan parsing" *National Conference on Artificial Intelligence*, 1994, pp. 1117-1122.
- [31] M.C. McFarland "Using bottom-up design techniques in the synthesis of digital hardware from abstract behavioral descriptions" *Papers on Twenty-five years of electronic design automation*, 1988, pp. 602-608, New York, NY, USA. ACM.
- [32] R. Guindon "Designing the design process: exploiting opportunistic thoughts" *Human-Computer Interaction*, 1990, 5(2), pp. 305-344.
- [33] B. Hayes-Roth and F. Hayes-Roth "A cognitive model of planning" *Cognitive Science A Multidisciplinary Journal*, 3(3), 1979, pp. 275-310.
- [34] A.L. Patalano and C.M. Seifert "Opportunistic planning: Being reminded of pending goals" *Cognitive Psychology*, 34(1), 1997, pp. 1-36.