SUPPORTING NOVICE MULTIMEDIA AUTHORING WITH AN INTERACTIVE TASK-VIEWER

by

Yasushi Akiyama

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
July 2015

Table of Contents

List of Tables

List of Figures

Abstract

Many novice users struggle with multimedia authoring (MMA) tasks. MMA tools often have extensive feature sets and correspondingly complex user interfaces that impede these users' creative pursuits. My goal was to provide support for novice users so they can more effectively utilize MMA systems.

First, I conducted User Study 1 to observe both novice and advanced users, and from this group of users, I identified approaches that experts often employed but novices did not. Drawing on an analysis of MMA systems and theory of general problem-solving, I hypothesised that these approaches were related to what I call *meta-tasks*. A meta-task involves the generalized work of both constructing and maintaining an appropriate representation of a task. Advanced users execute meta-tasks effectively while novice users find them significantly more difficult. Based on this assumption that the issues experienced by this set of novice users could largely be explained by considering meta-tasks, I identified potential challenges underlying novice difficulties and developed a set of relevant design guidelines to address them.

Next, I followed an iterative approach to design a prototype MMA environment that is conducive to effectively satisfying the design guidelines. The overall design process consisted of several cycles of development, evaluation, and refinement. In User Study 2, I explored and compared different possible design configurations in order to establish a set of support features. This MMA environment consists of a standard MMA system together with a new support tool called the Interactive Task-Viewer (ITV). The ITV allows novice users to view and explore the complex task structure of MMA effectively.

Finally, I conducted a usability study (User Study 3) to evaluate a prototype of this new environment. The results revealed that, compared to the traditional MMA environment, (1) users were able to complete specific MMA tasks faster in this new environment, (2) users found the new environment easier to use, and (3) the new environment positively addressed the challenge of meta-task difficulties for novice users, thus this new environment effectively supported this set of novice users.

Acknowledgements

This work could not have been completed without support of many people:

> I am very grateful to my supervisors, Dr. Sageev Oore and Dr. Carolyn Watters. Their expertise, guidance, and *patience* have allowed me to shape my ideas into what I have accomplished in this work.
>
> I wish to acknowledge the contribution of the members of my Ph.D. committee, Dr. Derek Reilly and Dr. Tami Meredith, whose insights and suggestions have inspired me and prompted great improvements throughout this project. I would also like to thank my external examiner, Dr. David Mould, for his insightful comments and feedback on my work.
>
> I would like to extend a special thank-you to Dr. Gary Woodill, who encouraged me to pursue this academic endeavour and lifted my spirit during the difficult times.
>
> There are not enough words to adequately express my gratitude to Sharon, Tanaka, and Kagan. They have supported me tirelessly throughout this journey.
>
> *"After all, I'm forever in your debt."*

Substantial portions of this thesis have already been published in refereed journals and conference proceedings: the motivational phase and the early results of the user observations and interviews (in Chapter 1 and parts of the preliminary user-studies

(User Study 1) in Section 3.1) were first presented at the *2013 International Conference on Multimedia and Human-Computer Interaction (MHCI 13)* and published in its proceedings [5], and then the extended version of the paper that included the early testing with low-fidelity implementation (Section 4.2.2) was published in the *Journal of Multimedia Theory and Application (JMTA)* [6]. The theoretical background (Chapter 2) and the framework for a multimedia authoring environment and its design development, including the results of the design exploration user studies (User Study 2) (Chapter 4) were published in *Multimedia Tools and Applications (MTAP)* [7].

Chapter 1

Introduction

Multimedia consumption has increased to the point where it saturates our daily lives [62,122]. The demand for media content creation has correspondingly increased, and people without prior experience have taken great interest in authoring their own multimedia content. As is the case for novices in an unfamiliar task domain, significant difficulties can arise when learning new multimedia authoring tasks and tools. The research presented in this dissertation investigates approaches to address novice difficulties: it was thus necessary to identify novice-specific difficulties, understand their underlying challenges, and formulate a viable approach to address them.

In this chapter, I will first present the motivation of the research (Section 1.1), and then define the research questions that I endeavoured to answer (Section 1.2). I will then define the scope of this research (Section 1.3) and describe the contributions (Section 1.4). The overview of the dissertation (Section 1.5) is also provided in this chapter.

1.1 Motivation of Research

Our consumption of multimedia has proliferated due to easy access to numerous platforms and more people now have interest in creating multimedia content. For example, people watch hundreds of millions of hours on YouTube daily and over 400,000 hours of video are uploaded to YouTube every day [132]. With easy access to multimedia content on various devices (e.g., computers, mobile phones, tablets, portable media players), people without prior experience have taken great interest in authoring their own multimedia content. This research focuses on this group of users.

New users often have difficulties in unfamiliar task domains: Morris and Ward [130] suggest that one of the common issues with performing unfamiliar tasks is the inability to establish clear goals for the tasks at hand. Even when novice users have clear goals, however, they often fail to find effective ways to reach these goals. I propose investigating this issue in the context of general task planning processes: when compared to novices, experienced users tend to have many more options in their task-plans and they have the ability to easily recover when they encounter problems. Experts are also able to evaluate their plan options earlier than novices, and thus are able to recover from unwanted situations earlier [40, 54].

By considering the general problem-solving literature [51, 54, 94, 130], I explain that these differences between novices and domain experts in relation to the mental model of a task. Domain experts establish more concrete mental models of their tasks than novices do, and they employ task-decomposition processes more often than novices [59]. That is, experts divide the ultimate goal into smaller, more manageable sub-goals, and think of possible actions to reach each of the sub-goals, while novices often tend to have problems with these steps. These factors—clear goal formation, sub-goal recognition, and the ability to come up with a set of plans to achieve the goals—are the key elements for generating an appropriate mental representation of a task, which, in turn, is one of the most important steps toward successful task performance [130].

Furthermore, some of the MMA tools are particularly difficult to learn. Standard MMA systems were originally developed to emulate the specialized hardware devices in the early days of multimedia productions [33]. While the feature set is extensive, one detrimental result of these digital skeuomorphs is correspondingly complex user interface design (e.g, popular video editing tools, Adobe Premiere Pro [2], Apple Final Cut Pro [10], and Sony Vegas Pro [119], all have over several hundred action commands). Feature-rich software designs have been found difficult to learn and use [71, 86, 87]. Thus, the same factors that helped those users who had experience with the hardware counterparts, also prevented many new users from learning and using these tools with a high degree of proficiency [4, 33, 35].

I will begin by carefully considering the characteristics and approaches of novice users, especially in relation to those of expert users. Once these differences are identified, I will explore how to support novices with the intention of bridging this gap.

1.2 Research Questions

Driven by the motivation to support novice users of MMA systems, in this dissertation I sought to answer the following research questions (RQ).

RQ1: Can we better understand the problems that prevent novice users from performing MMA at expert levels?

(a) Are there common issues experienced by novices?

(b) If so, are there common challenges underlying these issues?

RQ2: How can we address these challenges?

(a) What are possible solutions to provide viable support?

(b) Which solutions are worthwhile implementing initially?

RQ3: Do these solutions help novice users?

(a) Do they help novices to complete MMA tasks faster?

(b) Do novices find the new approach easier to use?

(c) Does the new approach successfully address the identified challenges (in RQ1)?

1.3 Research Scope

I define the scope of this research as follows:

Primary problem to address

Novice users face significant challenges when learning how to use common MMA systems. Aside from the obvious unfamiliarity with typical MMA paradigms and tools, one known difficulty that domain novices typically face is both generating and maintaining an appropriate mental representation of an unfamiliar task. These are two of the important steps for successful task performance [130].

This research investigates approaches to providing support for these users through the process of generation and maintenance of a representation of MMA tasks so that novice users can effectively utilize unfamiliar MMA tools.

Types of MMA tasks

While multimedia may refer to various formats including hyper-text documents, presentation slides, and more, the focus in this research is on two of the quintessential tasks of multimedia production: video editing and music/sound production. These two tasks account for the creation of vast amounts of multimedia content consumed daily [56]. Video editing and sound production tasks both share key elements [64]:

- The media have an inherent temporal element; the created media have a beginning and an ending and can be played back (as opposed to static media such as photos).
- Users deal with one or more data streams (tracks) that can be played back simultaneously.

Types of MMA systems

The type of support that this research sought to provide is designed to be used with standard MMA systems. Although this support is not tied to any particular MMA tool, I will primarily focus on MMA systems with common interface designs and features that experienced users normally use (e.g., Adobe Premiere Pro [2]). Tools with simplified and unique interfaces designed for MMA (e.g., introductory tools that are specifically designed for domain novices by entirely omitting extensive advanced features) are not the primary focus of this research. Discussion of existing MMA tools is provided in detail in Section 2.1.3.

Target user group

The target users of this research are novices who:

1. are unfamiliar with standard MMA and editing tasks and terminologies (i.e., no experience with digital or analogue media authoring tools).
2. may have been occasionally exposed to simplified tools mentioned above, but are not frequent users of any type of MMA tools.

1.4 Research Contributions

Having defined the scope of the research, I summarise the research contributions, organized according to the research questions described in Section 1.2.

RQ1: Can we better understand the problems that prevent novice users from performing MMA at expert levels?

Contribution 1: **Identify novice issues**: I identify common issues hindering novice users from performing essential MMA tasks by comparing behaviours of users with different skill levels (Section 3.1).

Contribution 2: **Introduce and define task-view and meta-tasks**: I introduce the concept of a *task-view*, which is a (full or partial) representation of the structure of a task (Section 3.2.2). I also introduce a novel concept of *meta-tasks* that MMA users perform to construct, modify, and utilize an appropriate task-view (Section 3.2.3).

Contribution 3: **Identify underlying challenges**: I use this concept of meta-tasks to articulate the challenges that may be underlying the observed novice difficulties (Section 3.2.3).

RQ2: How can we address these challenges?

Contribution 4: **Provide design guidelines**: I develop design guidelines for new MMA environments, with the intention of addressing the challenges, and therefore indirectly also the observed difficulties (Section 3.3).

Contribution 5: **Prototype support system**: Informed by these guidelines, I design and implement a prototype MMA environment with a new support tool called the Interactive Task-Viewer (ITV) (Chapter 4).

RQ3: **Do these solutions help novice users?**

Contribution 6: **Present evidence of relevant support**: I conduct a usability study and describe how the guidelines and prototype ITV do indeed support novices in the use of MMA systems (Chapter 5), and describe limitations and future work (Chapter 6).

1.5 Chapter Summaries

Following this introductory chapter, Chapter 2, "Background," provides the theoretical foundation in the related domains such as MMA tools and associated task. It also provides some background on theories of problem-solving with a main focus on common cognitive models, and an emphasis on their possible application for the design of MMA support tools.

Chapter 3, "Identification and Examination of Novice-Specific Challenges," details the process of identifying and examining novice-specific challenges and developing design guidelines. This process began with User Study 1. The chapter provides a detailed overview of this study including the analytical framework, the analysis, and the findings, that led to the identification of *meta-tasks*. This process culminated in a set of design guidelines that aim at addressing the underlying novice challenges when performing meta-tasks.

Chapter 4, "Design Development of the New Multimedia Authoring Environment" details the designing process of the guidelines into a new MMA environment. User Study 2 provided insights necessary for developing a design space that would best meet the demands of the design guidelines developed in Chapter 3. This chapter elaborates on the methodology of and insights derived from User Study 2 and the subsequent rationale for the design choices that were made in the development of the new MMA environment.

Chapter 5, "Evaluating the New Multimedia Authoring Environment," first describes a prototype of the new MMA environment, with a support tool called the *Interactive Task-Viewer*, which is designed to effectively satisfy the guidelines. The chapter then details the evaluation process of this MMA environment. User Study 3 assessed the support provided by the environment for novice users and measured its effectiveness.

Chapter 6, "Conclusions," presents an overview of insights derived from this work. This chapter addresses the implications for applications of the approach presented in this dissertation in other authoring contexts, limitations and subsequent directories for future work.

Chapter 2

Background

This chapter provides background on a number of related topics. The first section focuses on multimedia authoring (MMA): I present a brief history of MMA tasks and associated tools (Section 2.1.1); I discuss characteristics of MMA tasks, and I develop important concepts such as decomposition of general MMA tasks and the construction of a task structure for MMA (Section 2.1.2). Next, I present theories of problem-solving and task-planning processes, especially with respect to known cognitive models of established problem-solving strategies (Section 2.2). This chapter also provides discussion of relevant work from several domains such as MMA systems development, interface designs for novice users, visualization tools for multimedia data, and task notation and visualization approaches (Section 2.1.3).

2.1 Understanding Multimedia Authoring

MMA tasks typically consist of a series of smaller interdependent subtasks and these subtasks must be structured in specific ways to accomplish particular goals for the authoring tasks. Based on my informal observation of users with varied skill levels, variation exists as to how people perform MMA tasks—some who have little experience start their tasks without any initial plans and perform arbitrary subtasks in an ad-hoc manner, while more experienced digital media creators can begin with more concrete plans. These initial plans may change during the task processes, but experienced multimedia authors often use these initial plans as a guide through complex tasks [32].

This initial planning is closely related to the generation of a mental representation of the task, which is one of the keys to successful task performance [130]. However, due to the lack of the necessary experience and knowledge, those who are new to the domain often fail to come up with any plans altogether, or rely on inadequate plans that do not result in an ideal outcome [48]. Novices may fail to properly identify key elements of the task structure such as effective and accurate goals and subtasks, and the relationships between them. This may lead to an inaccurate or inappropriate task representation. In other words, novices tend to have a larger "gulf of evaluation" as well as a larger "gulf of execution" [95].

Before discussing cognitive representation of task structures, I first review multimedia authoring. This section gives a brief history of digital media authoring techniques and technologies, and then discusses and defines key elements of MMA tasks. Finally, it examines existing MMA systems.

2.1.1 A History of Multimedia Authoring Tools

Although historically many digital tools have been designed for professionals, in recent years there has been an increase in tools targeting inexperienced users. As multimedia authoring and editing styles have emerged from the older days of analogue video and music production, we can better understand current tools by considering those from which they evolved.

2.1.1.1 Physical Film Manipulation: Silent Films and Analogue Devices

The earliest films date back to 1895 when motion pictures started to move from a novelty to an established business [31, 34, 39]. These early productions were silent, mostly due to the technical difficulties of synchronizing sounds with the moving pictures. Most were less than a minute in length, and they used a single-shot perspective. Therefore, editing was not performed at all or was very minimal.

In 1903, a film pioneer, Edwin Stanton Porter created a film called *The Life of an American Fireman* that consisted of twenty shots, and it is considered one of the earliest films to employ extensive editing processes. In contrast to the preceding films, it used new film editing techniques such as shot-by-shot alternation of locations (e.g., interior and exterior of a building) that added tension to the story and the use of newsreel footage of a real fire that conveyed a sense of authenticity [31]. Beginning in 1908, another film director, D. W. Griffith adopted many of the modern film editing techniques that are still in use; the adaptation of various shots to create an impact (e.g., extreme long shots, close-ups, cutaways, and the tracking shot), cross cutting (often called parallel editing) that alternates two or more scenes that are occurring concurrently in different locations, and the effective use of variations in pace.

There are other notable film makers from this era: Vsevolod I. Pudovkin attempted to develop a film editing theory to go beyond the classic editing methods of Griffith. Sergei Eisenstein developed a theory of editing, consisting of five components, 1) metric montage, 2) rhythmic montage, 3) tonal montage, 4) overtonal montage, and 5) intellectual montage. Dziga Vertov treated film as an experiment of realism as opposed to the fictitious narrative approaches of his predecessors and contemporaries. Until the introduction of non-linear editing systems (described in the following section), film editing was done by physically cutting pieces of film and pasting them together, using devices such as guillotines and splicers, and then threading edited films on a machine with a viewer such as a flatbed editor [39]. This editing style and terminology, such as splicing and cutting, are still in use in the field of film editing, even after the physical manipulation of film has been almost entirely abandoned and replaced by digital technologies.

With the advent of sound technologies for recording multiple sources of sound in separate tracks [31], and later, the ability to mix and synchronize the tracks with pictures, many film makers experimented with sound added to their film production.

The evolution of sound technologies introduced the notion of multi-track recording and editing, which has essentially remained the mainstream style of film and sound production until this day, and has greatly influenced the interface designs of computer software tools for authoring and editing. A detailed discussion of existing tools is provided in Section 2.1.3.

2.1.1.2 Videotape Recording and Editing: Invention of Magnetic Tapes

The invention of the videotape recorder in 1951, was a significant advance, though it took years for its potential to be realized. The first videotape recording devices were very expensive, and the loss of video quality during the copying processes was significant. Consequently, video editing did not replace traditional film editing until improvements in quality and economy were made, and new technologies were developed [80].

With the invention of magnetic tape, editing for both audio and video tape was done linearly in a similar fashion to film editing. Videotapes were cut with razor blades and guillotines, and spliced to create a linear sequence of cuts [31, 80]. A disadvantage of magnetic tapes, however, was that pictures and audio were recorded in separate tracks, while the read heads were several inches apart. This caused some discrepancy in timing between the video and audio, thus it was impossible to perform a physical edit that would work for both tracks. One of the work-around methods was to make cuts only for the video (and possibly a portion of the audio), and then copy it so that it maintained synchronization with the audio track. Another disadvantage was the cost of tapes: editors could not reuse edited videotapes when there were too many edits/splices, and even though the cost of videotapes had decreased, the cost of not being able to wipe and reuse videotapes was significant.

Despite some of the disadvantages in editing, magnetic tapes continued to be the main media for recording and storing analogue and digital data. The variants of consumer magnetic tapes include tapes used for reel-to-reel devices (for analogue audio), Compact Cassette (for analogue audio), Betamax (for analogue video), Video Home System or VHS (for analogue video), miniDV (for digital video). With the invention of cassette recorders (both video and audio) and consumer camcorders, magnetic tapes gained popularity, becoming a common household item until other forms of digital media storage started to appear (such as compact discs and DVDs).

2.1.1.3 Nonlinear Film Editing: Moving to Digital Devices and Software Tools

The first nonlinear video editing system, CMX 600 introduced by CMX Systems in 1971 [106], consisted of two monitors, one to display the preview video and perform editing actions using a light pen, the other to display the edited video. The data were stored digitally on disk packs and this allowed editors to randomly (or nonlinearly) access any edit points. This was a contrast to films and electronic tapes that only allowed editors to access data linearly. For example, suppose one already knows the timecode of a video frame which they are trying to reach. In linear editing systems, editors would usually have to shuttle back and forth to find this particular point on the film or tape. In nonlinear editing systems, however, editors could enter this timecode and the system would directly move to this position on the video. In addition to the timecode information, modern systems usually allow some other information (metadata) to be attached to the stored media such as "take" numbers, scene numbers, clip names, and so on, thus allowing the retrieval of and access to data in ways that are not possible in linear editing systems.

Although it was the first nonlinear editing system, CMX's main purpose was to allow off-line editing, an editing technique that preserves the original raw footage intact (i.e., by making copies) and performs editing on the copied medium. CMX allowed the recording of an ordered list of editing processes known in the field as an edit decision list (EDL) without modifying the original data, so that the EDL can be used later when the actual on-line editing occurs. EDL is basically a list of instructions on how and where on which film (tape, digital data, etc) each edit action should be performed. These instructed edits are first performed on copies of the original media rather than on the actual raw media. In the analogue era, the benefits of off-line editing were fairly obvious in that editors did not have to modify the original media until they were satisfied with all the performed edits on the copied media. Even with digital tools that allow nonlinear editing, however, off-line editing has a huge benefit—the original video and sound data are usually recorded at extremely high-quality settings and are often too large to be smoothly edited on even high-performance computers. The off-line edit can be performed on copied digital data with reduced quality much more efficiently. An EDL is created during this off-line editing, and then later it is used to perform on-line editing on the original high-resolution data for the final production. The concept of EDL is still in use today, and many video editing software products including Adobe Premiere Pro [2], Final Cut Pro [10], and Sony Vegas [119], support import and export of EDL files.

2.1.2 Multimedia Authoring Tasks

2.1.2.1 Characteristics of Multimedia Authoring Tasks

In a broad sense, MMA tasks can refer to processes of creating and editing digital media in many different formats, e.g., videos, music, hyper-text documents, presentation slides, and more. The focus in this research, however, is on video editing and music/sound production. These two tasks account for the creation of vast amounts of multimedia content consumed daily[1], from YouTube videos and podcasts, to more

[1] For example, in 2014, an average Canadian watched TV for 1758 minutes per week and listened to radio for 1065 minutes per week [56], and people watch hundreds of millions of hours on YouTube every day, and 300 hours of video are uploaded to YouTube every minute [132].

general music, movies, TV shows, and commercials.

Video editing and sound production tasks both share key elements [64]:

- The media have an inherent temporal element; the created media have a beginning and an ending and can be played back (as opposed to still/non-interactive media).
- Users deal with one or more data streams (tracks) that can be played back simultaneously.

Designing comprehensive interfaces to support creativity still remains challenging [27,37,61,115,125,131]: digital media creators need to have skills both in the creative domain (e.g. "how do you tell a story through video?") and also in the digital one (e.g. "how do you implement a particular split-scene effect?"). It is not uncommon to see media creators having problems when transitioning from traditional methods to computer software tools. While artists may find at least some skills transferable to digital authoring, many beginning users find it difficult to effectively utilize these digital tools [33].

2.1.2.2 Decomposition of Multimedia Authoring Tasks

The basic operations typically iterated throughout the MMA are (1) choosing desired data clips, (2) creating/recording new clips, (3) modifying clips, (4) placing clips in the workspace, (5) saving and export of the workspace, and rendering files to appropriate sound or video formats. Each of these operations generally has sub-operations resulting in a hierarchical task structure.

(1) Choosing (importing) desired data clips:

This operation consists of several sub-operations (e.g. iteratively browsing and previewing, and then importing media) and is usually repeated throughout a project.

(2) Creating new clips:

The sub-operations typically include specifying media properties (e.g., sampling rate, resolution), setting hardware devices (e.g., microphone), switching to the creation mode, and reviewing the created media. Again, many of these sub-operations are often repeated.

(3) Modifying clips:

This operation includes sub-operations that modify media clips (e.g., trimming, slicing, and resizing) and that change clip properties (e.g., gain and opacity). It also includes sub-operations to apply effects (e.g., sound distortion and video transitions). This operation is almost inevitable for newly created clips, but it is also very common to modify details of existing clips. Cycles of modifying and reviewing occur frequently throughout authoring tasks.

(4) Placing clips (specifying timings and tracks):

One of the distinctive tasks in MMA is specifying the timings of clips. Each clip is placed along the horizontal time-axis indicating its playback timing, often in relation to other clips. Vertical position of a clip indicates which *track* it belongs to. A track is a place-holder for clips of the same media type, and multiple tracks can be played simultaneously.

The sub-operations typically involve deciding horizontal position (timing), deciding vertical position (tracks), reviewing the media, and adding or deleting tracks. Adjusting the timings of the clips is also performed repeatedly until the desired layout is obtained. Moving a clip may not only cause other clips to be repositioned, but may also require them to be modified (e.g., trimmed) to accommodate the changes in locations of the clips.

(5) Saving and export of workspace and rendering:

Most authoring tools offer options to save only some parts of the workspace (e.g., individual media clips) in addition to saving the entire workspace. Exporting is the operation to convert the workspace to file formats compatible with other authoring tools. Rendering is the operation to convert the workspace into a desired media format so that it can be viewed/played in common media players. Each of these operations involves sub-operations such as specifying parts to be saved/exported/rendered (data and time duration), specifying a file location/name, and selecting a file format (e.g., size and desired quality of a video).

To work on an MMA project, the user needs to structure a task by specifying the order of these operations. As each of the operations involves multiple sub-operations and there are often constraints that certain (sub-)operations need to happen before others (e.g., one needs to *arm*[2] a track before recording can be initiated), the construction of a task structure can be a complex process, especially for novice users.

2.1.3 Existing Multimedia Authoring Tools

Even with the recent emergence of MMA tools targeting less experienced users, the principal paradigm of media authoring styles still remains the same: it was developed during the era of the direct manipulation of physical film, and it is now inherited to digital nonlinear editing as described in Section 2.1.1. This authoring style is found difficult for novices to learn [4, 12, 35, 84].

In this section (2.1.3), I will describe common interface designs of MMA systems, as well as tools and approaches to visualization of multimedia data and task structures.

2.1.3.1 Standard Tools based on a Multitrack Layout

Existing MMA tools typically provide work-spaces containing one or more *tracks*, each of which holds one data stream. These tracks are laid out horizontally along a time-axis, similarly to a standard music score (shown in Figure 2.1). In the latter, each musical staff is generally designated for a part (instrument or voice), and the staffs are aligned vertically so that notes played at the same time are roughly in vertical column.

Multimedia tools employ an analogous approach to the standard music score for specifying the timing of the data streams. Figures 2.2 and 2.3 show screen-captures of several popular MMA tools. What all these tools have in common are tracks

[2] Arming a track means that you are preparing it for recording.

positioned vertically and aligned along a horizontal time-line (time axis), and media clips placed in these tracks specifying their playback timing. The vertical location of a clip indicates which track it belongs to. The horizontal location of a clip represents the timing of the clip to be played much like the notes in music notation. Media clips from different tracks will be played simultaneously when these clips overlap in time.

Figure 2.1: Standard format of music score. A group of five horizontal lines is called a *staff*. Each staff (or a set of staffs) represents notes to be played by the specified instrument, and the horizontal locations of the notes specify the timings of those notes to be played.

The use of multitrack layout and the metaphors derived from hardware device are still a part of a popular design practice for MMA systems to this day [3, 4, 21, 33, 35, 112]. Only in the last decade or so, other styles of MMA tool design practices emerged and have become a part of the popular design convention as described next.

Figure 2.2: Screen-capture of Apple GarageBand. An example of standard MMA tools' UI design: tracks positioned along a horizontal time-line.

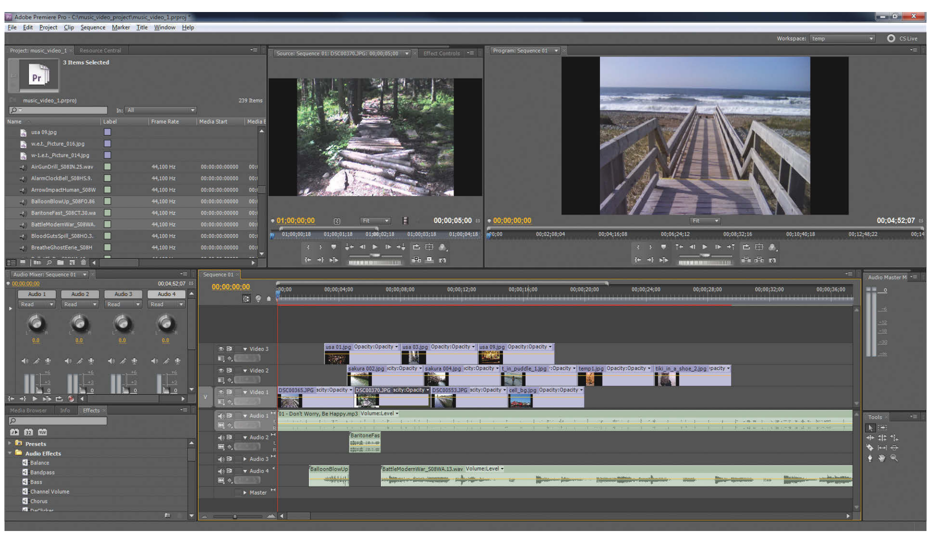


Figure 2.3: Screen-capture of Adobe Premiere. Another example of standard MMA tools' UI design.

2.1.3.2 Interface Designs for Complex Programs

There have been studies that investigated approaches to address issues of complex interface designs. These approaches can be very roughly grouped into three categories: (1) reducing complexity by altering the way systems look and function (i.e., providing new interface designs), (2) allowing users to view subsets of full features, and (3) gradually adapting designs. I will organize these studies based on these categories.

(1) Systems that provide new interface designs

Studies [3, 4, 41, 109] investigated an approach to design new interfaces that are different from standard designs such as discussed in Section 2.1.3.1. As part of MIT Media Laboratory's Toy Symphony project [81], Farbood et al. introduced HyperScore [41], a sketch-based composition tool for novice users. The users place "droplets" to create a musical motif, and then using those created motifs, they specify the structure of a music piece, by drawing lines (simple repetition of original motifs) or curves (with moderate pitch alterations) (Figure 2.4). HyperScore allowed children who participated in the Toy Symphony project to successfully compose music for string orchestra within five instructional sessions.

Moving away from the time-axis concept and the standard multitrack layout, PlaceAndPlay [3, 4] introduced an approach that allowed users to create and playback multitrack music. The system consisted of a small tangible multi-modal input device (Figure 2.5) and a workspace that contained three distinct areas: the "stage" area that allows placed clips to play back, the "recording" area that records incoming sound from a microphone to the placed clips, and the neutral area, where clips which are not involved in either playback or recording are placed (Figure 2.6). Users place icons, each representing a piece of stored sound, in one of those three areas to manipulate sounds and accomplish multitrack music production. The results from their focus group studies showed that children quickly understood the concept of

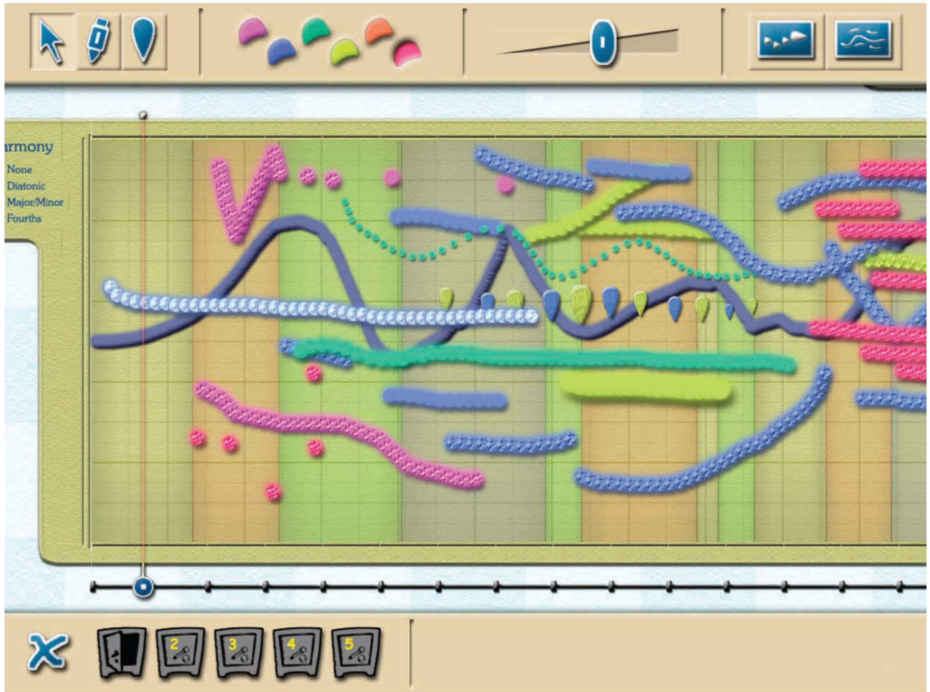


Figure 2.4: Screen-capture of HyperScore (Image reproduced from [41] copyright©2004 IEEE). It is less clear to see the multi-track layout in this example, but each line (curved or straight) essentially acts as an individual track.

spacial configuration of music clips and the system allowed these users without prior experience to successfully perform multitrack recording.

A multi-layer UI is also employed to address issues of complex UI: *CommandMaps* (Figure 2.7) [109] display all available action commands at once on the screen in a spatially-stable manner, capitalizing on users' spatial memory of frequently-used commands. Their experimental results show that while CommandMaps did not improve novice users' task performance for searching action commands, it yielded significantly faster performance of experienced users than the standard UI.

There are also several commercially available tools that use simplified interfaces by omitting functionality that is not commonly used or that is too complex to be learnt by novice users. For example, Sony's Super Duper Music Looper [118] and FlexiMusic Kids Composer [46] both have an interface that allows children to use a virtual paint brush and a virtual eraser to determine lengths of music samples (Figures 2.8 and

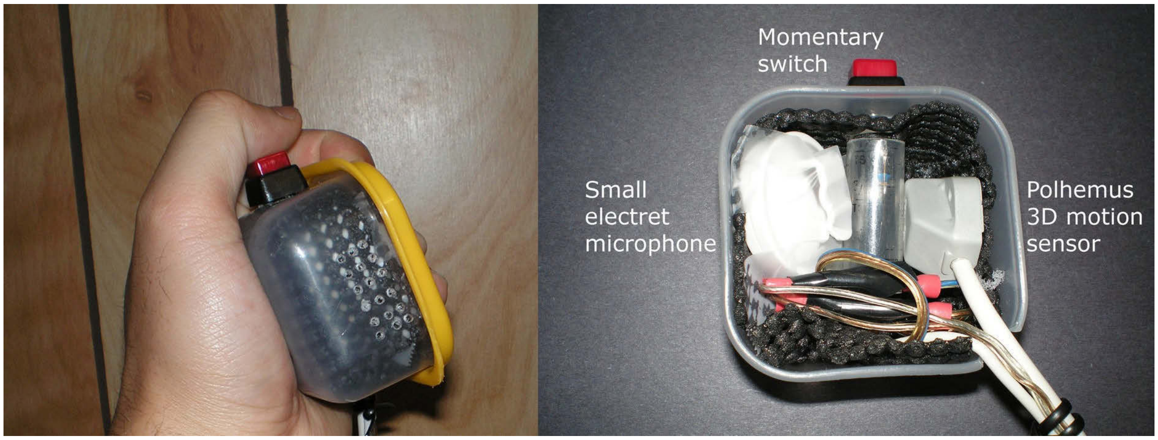


Figure 2.5: The input device for PlaceAndPlay, consisting of the in-air pointing device, a microphone, and a button to trigger certain actions.)

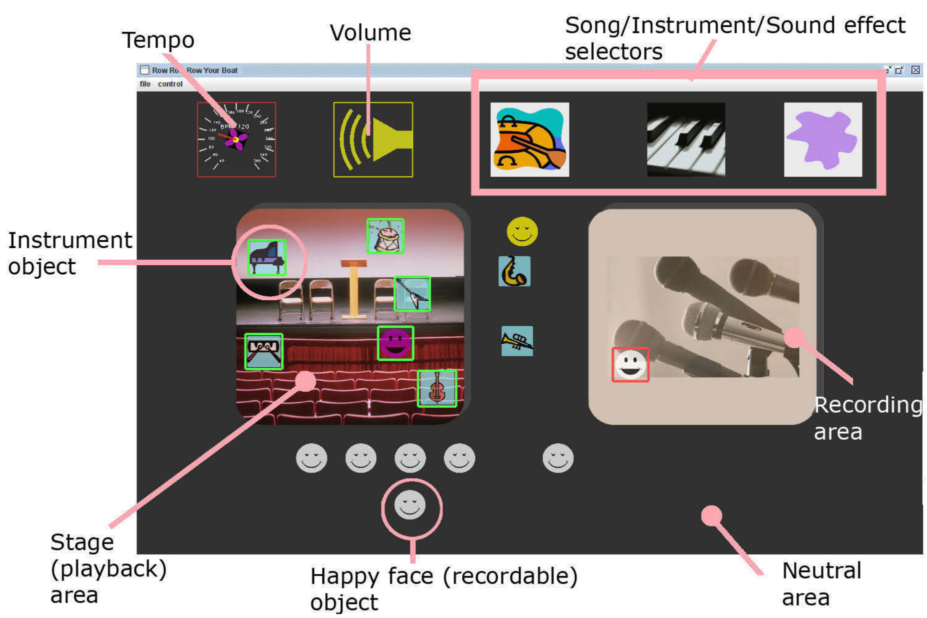


Figure 2.6: Screen-capture of PlaceAndPlay, showing the three distinct areas of stage, recording, and neutral areas. (Image reproduced from [4] copyright©2008 ACM)

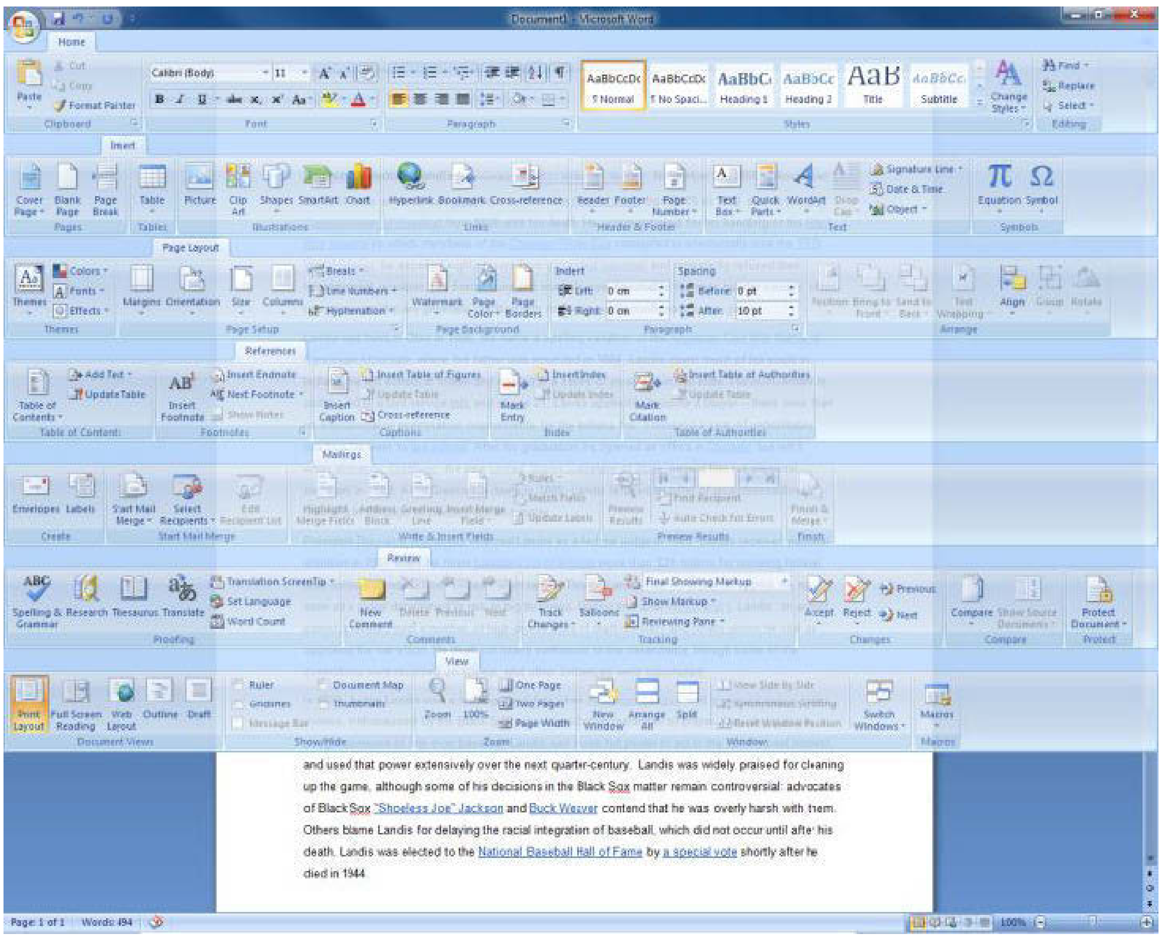


Figure 2.7: Screen-capture of CommandMaps. (Image reproduced from [109] copyright©2012 ACM). It displays all available action commands at once on the screen in a spatially-stable manner, capitalising on users' spatial memory of frequently-used commands.

2.9). Another commercial product is Groovy Music from Sibelius [116][3]. Instead of a paint brush and an eraser, its composition tool ("create module") offers many "shapes" that represent elements of music such as melodies, rhythm patterns, tempos, bass lines, and chords. Children drag these shapes onto a multitrack layout with the timing indicated by the horizontal axis.

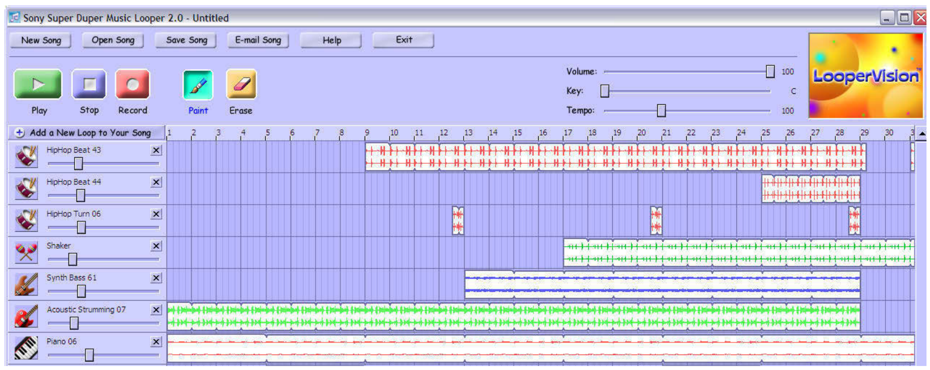


Figure 2.8: Screen-capture of Sony Super Duper Music Looper. An example of simplified UI for MMA, which employs the multi-track layout, typical of more advanced tools.

Figure 2.9: Screen-capture of FlexiMusic's Kids Composer. Another example of simplified UI for MMA. This tool also employs multi-track layout.

Other simplified tools use template or wizard-based approaches to allow users to bypass complex steps of MMA tasks. Such tools have became popular, especially with the increasing use of mobile devices. Examples include muvee Reveal X [91]

[3]This product has been discontinued.

(Figure 2.10), Magisto [83] (Figure 2.11), and Vidify [129]. All these software tools first ask the user to import/select media files such as videos, still images, and music/sound. The users will then be asked to select a theme or style for the final product, and then the tools will automatically create appropriate media based on the predefined parameters. The automatically created work can sometimes be modified to adjust details of the media, but this modification step, when available, might still be omitted by the users due to its complexity.

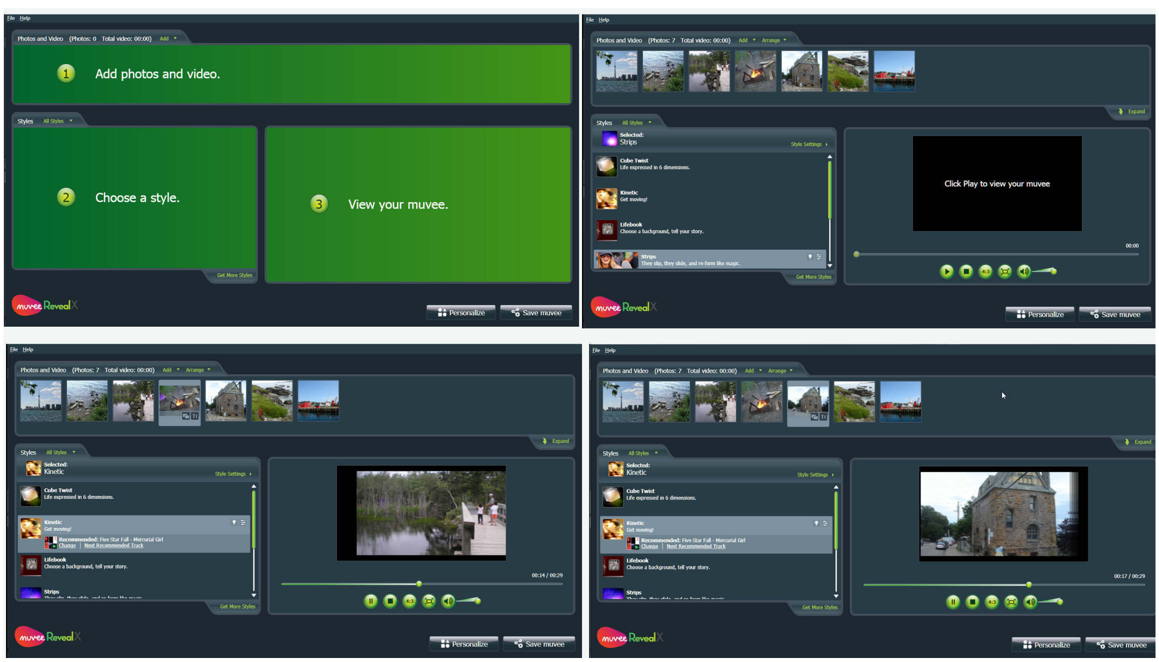


Figure 2.10: Screen-captures of muvee. An example of template or wizard-based approaches to allow users to bypass complex steps of MMA tasks. Users follow a small number of simple steps (top-left) to complete the creation of media.

(2) Systems that allow users to view subsets of full features

There are also systems [22,23,43,69–71,86] that allow users to customize interface designs based on individual users' needs and task-flows, thus allowing only subsets of full features to be visible at a given time. Lafreniere et al. [69] introduced AdaptableGIMP (later developed to *Workflows* [70,71]), a *task-centric* interface that adapts based on tasks at hand by specifying keywords: the users first enter a keyword in a search box

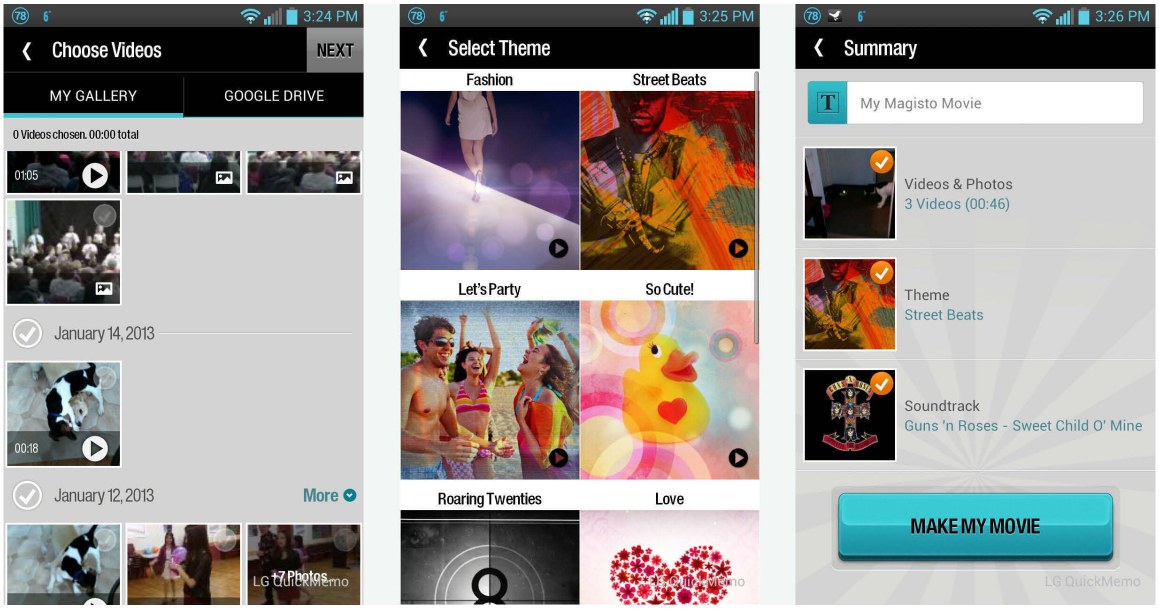


Figure 2.11: Screen-captures of Magisto. Another example of template or wizard-based tools. Essentially the same 3 steps: choose media clips to use, select style/theme, and make/view the resulting video.

(Figure 2.12a), based on which the system displays a list of names and descriptions for command sets for performing specific tasks (called *task sets*). From this list of task sets, they can select relevant one for their intended task (Figure 2.12b), and the system will load the corresponding command icons into the toolbox (Figure 2.12c). Task sets that are brought up by specific keywords are refined as collaborative processes by other users of this tool in the community. The principle behind their approach was motivated by a finding that users were task-focused and they do not spend time for initial learning for the general usage of a system, but rather tend to focus on immediate task itself [28, 30, 82, 105]. The findings by Lagreniere et al. [69] suggest that task-centric interfaces allowed users' exploration of a subset of features and help avoid difficulties caused by feature-rich software.

McGrenere et al. [86] and Bunt et al. [23] investigated approaches to allow users to personalize their interfaces to address the complexity of feature-rich software (Figure 2.13) [87]. Users can switch between the personalized UI and the default UI

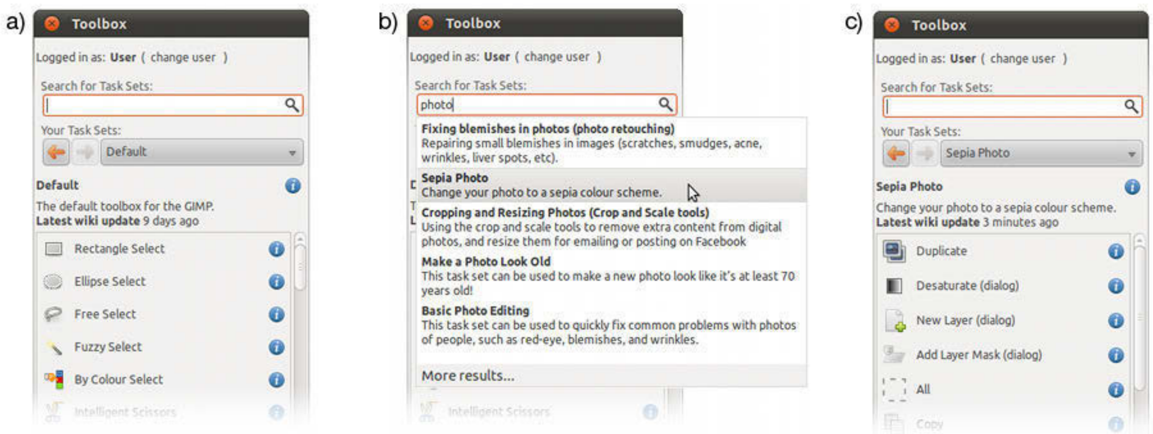


Figure 2.12: Screen-capture of AdaptableGIMP. (Image reproduced from [71] with permission of the author.) (a) the users first enter a keyword in a search box, based on which, the system displays a list of names and descriptions for command sets for performing specific tasks (called *task sets*). (b) From this list of task sets, they can select relevant one for their intended task, and (c) the system will load the corresponding command icons into the toolbox.

with all the standard features. Their design framework consists of three elements: it integrates two interfaces (full and personalized), it is adaptable, and it begins with a small subset of the full menu until users start modifying it. The features in the personalized UI are a subset of those in the standard UI with the order in which the action commands appear in each menu group is preserved. The study results [86] showed that participants navigated through the menus and toolbars better with this approach than with the standard interface.

Findlater et al. [43] proposed ephemeral adaptation by temporally changing the GUI to improve performance and reduce visual search time. The menu system used in their study first rendered adaptively predicted items (i.e., a subset of the full menu items) when the menu is opened, and non-predicted menu items gradually appear (i.e., animated from a hidden state to a fully visible state in place). Comparisons of the ephemeral adaptation approach against standard static menus revealed that ephemeral menus were faster than static menus for cases in which the system predicted menu items correctly while the ephemeral menus did not perform worse when

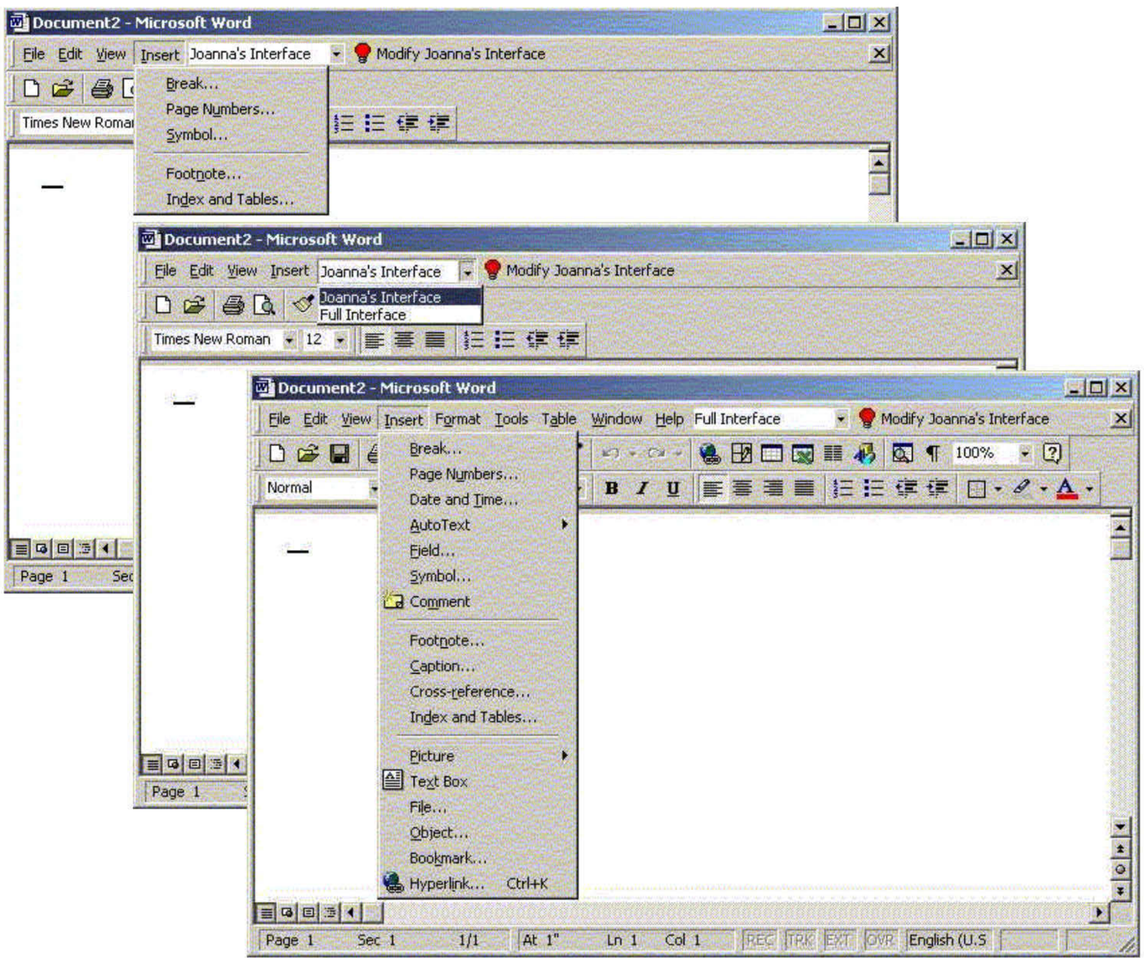


Figure 2.13: Screen-capture of Personal Interface by McGrenere et al [23, 86] (Image reproduced from [86] copyright©2002 ACM). Users can switch between the personalized UI and the default UI with all the standard features: (Top) It shows the Insert menu of the personalized UI, (middle) the user switches the UI's, (bottom) it now shows the Insert menu of the original UI.

predictive accuracy was low.

(3) Systems that gradually adapt their interface designs

Other studies use an approach to gradually adapt the UI: Shneiderman [114] proposed the interface design to accommodate needs for users at different levels of skills and experience to allow them to gradually adapt to more complex design layers. A multi-layer interface approach is also used by Leung et al. [74] to introduce a reduced functionality interface layer for older adults before the users progress to a more complex interface layer.

Scarr et al. [110] compared these different styles of the interface designs, using extensions of CommandMaps [109]; *StencilMaps* and *EphemeralMaps*. As described above, CommandMaps employ a new way to present menu commands that capitalize on static locations of menu items. StencilMaps utilizes a semi-transparent stencil overlay to hide some parts of CommandMaps to only display a subset of action commands in its menu UI. EphemeralMaps use the ephemeral technique to first display only a subset of commands and then the rest of the commands fade into view gradually. The authors found that these extensions to highlight certain commands within CommandMaps did offer benefits for visual search of target commands, but the highlighting did not help for longer-term performance as it did not fully capitalise on users' spatial memory.

The approach that I will take differs from the above approaches. Rather than simplifying the original interface, I will present a tool that augments it with an additional view that is simpler yet connected to the original interface. Rather than reducing the visible functionality to a subset, the original view is preserved while the additional view provides easy access and visual "links" to a relevant subset. Rather than having the original interface slowly adapt, the additional view can act as a guiding template and the user is free to switch to the original interface as they become more proficient. Finally, I note that the current work is in fact complementary to many of the above approaches, in that many of them could easily continue to work if they were to be augmented with an additional view as I will present in this thesis.

Toward this goal, I will discuss approaches to visualize structures of complex tasks in the next section (2.1.3.3).

2.1.3.3 Visualization Tools for Multimedia Authoring

Visualization of task structures can potentially lead to ways of providing assistance to novices who struggle to generate and maintain appropriate mental representation of a task: generation and maintenance of mental models of a task are related to human problem-solving more generally [130], and involve aspects of familiar problem-solving strategies as conceptualized in standard cognitive models (as will be discussed in detail in Section 2.2). As these general tasks often occur at more contextual levels, and often independent from specific functionalities of MMA systems, providing necessary support for novices to perform these tasks will likely require a different approach from what existing interface designs offer.

Jourdan et al. [65] proposed interfaces that consisted of multiple views such as a presentation view, a scenario view, a objects view, and a storyboard view, to inspect multimedia documents from different perspectives. Each view was a result of the application of a specific filter on the same document, and thus all the views could be simultaneously updated by modifying data in any one of the views. Layaïda and Vion-Dury [73] presented an interface that also utilized multi-views, consisting of a structural view, a resource view, and a presentation view, for authoring multimedia documents. The idea of multi-views of the same data is not new and has been employed in many MMA tools, most notably in 3D animation and modelling tools such as Autodesk Maya [13] and Blender [17]. The concepts of the multi-views in these particular examples of 3D modelling systems, however, differ from the multi-view concept used in the studies described above. The multi-views in the 3D modelling examples refer to *multi-camera* views, while the studies described above targeted multimedia *documents* which do not exhibit obvious physical view modes but a more conceptual perspective of the data. It is worth noting that, when applying the multi-view ideas to MMA systems, how data are perceived can vary depending on the context of the task (e.g., a single musical note can also be seen as a part of a phrase).

For viewing tasks instead of data, Soundium's Design Tree [90] collects and organizes artwork and attempts to visualize undo/redo processes of artwork realization. It uses a multi-level hierarchical structure represented by a tree-style visualization, each node representing a design state that occurs during the creation process. The idea is similar to the multi-branch undo mechanism[4] that uses a tree-structure to show all the performed actions including ones that are undone [16, 38, 52, 67, 68, 88].

Novel approaches to visualize tasks-models have also been proposed outside of MMA and creative domains. Paternò and colleagues [19, 101, 102] proposed ConcurTaskTrees, a hierarchical task specification notation and corresponding tools that utilized this notation. The ConcurTaskTree employs a tree-style visualization to indicate hierarchical structure of a task, and symbols to indicate the order and the concurrencies of subtasks. It can be used to describe tasks at different levels of abstraction as well as their temporal relationships. Lu et al. [79] presented an approach to visualize tasks using UML, which is automatically constructed from task models represented in a semi-formal notation, and the associated editor called Tamot [78]. XML-based user interface description languages were developed by Stavness and Schneider [120] to support user interface portability across multiple platforms. Balbo et al. [14] developed a taxonomy for classifying task notations, using six different axes to describe and classify them. These task visualization and notation methods are typically used

[4]The discussion of the multi-branch is presented in Section 4.2.1.

to describe tasks and work-flows, and thus were meant to be used for application interface and interaction designers.

Having reviewed relevant work from several domains such as MMA systems development, approaches to address issues of complex UI designs, visualization tools for multimedia data, and task notation and visualization approaches, I will present theories of problem-solving processes in the next section.

2.2 Theories of Problem-Solving

I present background on theories and cognitive models of problem-solving, with a focus on approaches based on task-planning. This theoretical background provides insights on issues that novice users experience during the generation and maintenance of mental models of a task. As full literature on this topic is beyond the scope of this dissertation, the focus here is on common cognitive models of problem-solving strategies, developed mainly in the domains of Cognitive Science and Artificial Intelligence, with emphasis on possible application for the design of MMA support tools.

Creative tasks can be considered high-level problem-solving tasks [36, 97, 98]. The subtasks typically involve key aspects of problem-solving, such as: identifying high-level goals (e.g., "What underlying story should be told in this video?"), assessing the current situations/states (e.g., "Does this set of video edits flow well?"), and recognizing the available options (e.g., "I can either move this clip slightly or resize it so that it starts with the music track").

One view of the general problem-solving task is to consider it as a search through a space of connected problem states [130]. In this respect, a problem-solving task has two stages. The first stage is to form a suitable mental representation of the problem space, including elements such as an initial state, a set of possible actions, and a goal. Once this representation is formed, then the second stage is to find a path from the initial state to the goal [121].

For example, consider the three-disc Tower of Hanoi (ToH) puzzle[5] (shown in Figure 2.14). Solving this puzzle is clearly different from many creative tasks, but it is the consideration of those differences that draws our attention to some important characteristics of the latter tasks. In the ToH problem, there are nine possible actions and 27 different states, with a single clear goal state. The puzzle solver performs actions to move between states. At each state, there are two or three possible actions and in this way, each state is connected to two or three other states. In this simple problem, it is possible to visualize the entire state-space of this particular puzzle (Figure 2.15). Using this visual aid, one can visually search for the goal state among a limited number of the possible outcomes, and then work backwards to find a path between the initial and goal states. However, for harder versions of this puzzle with more discs and pegs, and for real-life tasks such as MMA tasks, the number of possible states is extremely large.

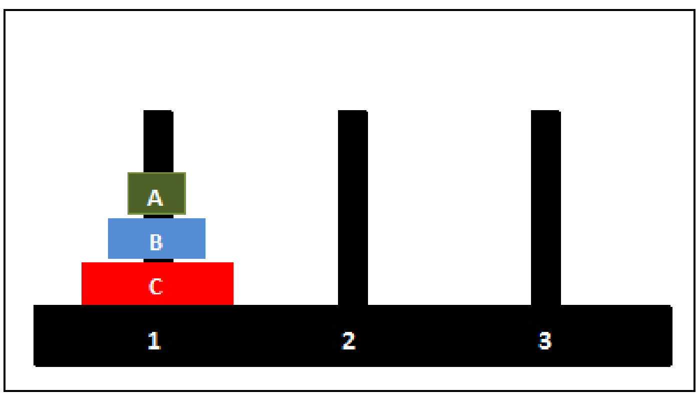


Figure 2.14: Tower of Hanoi puzzle: The puzzle consists of three discs of different sizes and three pegs as shown. The objective is to rearrange the discs to move the tower from one of the pegs to another while following some rules: (1) Only one disc can be moved at a time, (2) Only the top-most disc can be moved, and (3) No disc may be placed on top of a smaller disc.

For solving complex problems, humans typically use heuristics and other strategies to either reduce the complexity of the problem, or focus on smaller parts of the

[5]The Tower of Hanoi puzzle is a classic puzzle that has been widely used for problem-solving analyses [20, 44, 53, 54].

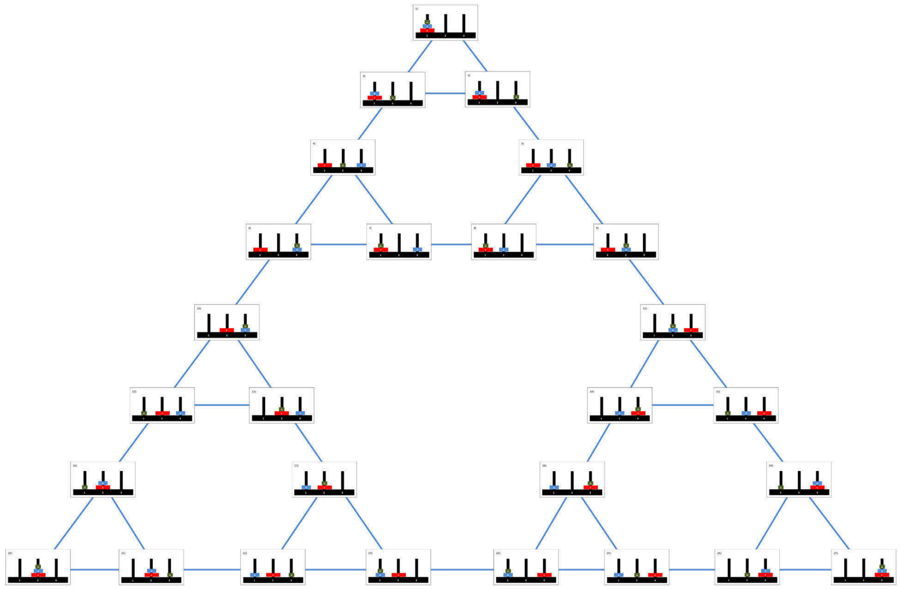


Figure 2.15: The state space of the Tower of Hanoi puzzle: The puzzle has 27 different states. By performing one or more actions in a sequence, the puzzle solver can move from one state to another. At each state, there are two or three possible actions, and the execution of each action will result in a new state, thus each state is connected to two or three other states.

problem [93, 94, 117]. An early cognitive model of problem-solving strategies is the top-down strategy [48,58,107]. This strategy is often referred to as problem-reduction or task-reduction in some disciplines. The basic idea is that the problem solver divides the original problem into smaller sub-problems and continues recursively until each sub-problem becomes simple enough that it can be solved directly without further divisions. In other words, at the beginning of this strategy, the task representation is abstracted with simplified problem definitions and one tries to solve the problem with fewer and simpler constraints [107]. A simplified task representation may reduce the cognitive load, but will lack detailed information.

In contrast, the bottom-up strategy [15, 54, 85] uses lower-level, detailed information to plan actions, and is often regarded as a data-driven approach [54]. In this strategy, one starts by focusing on lower-level actions/subtasks that operate directly on given data. Using the detailed information at this level, they plan future actions by gradually working toward more abstract and contextual solutions of the task. For example, one may notice that there are some dirty dishes in the sink, and as the person starts doing the dishes, they maybe lead to an array of other kitchen chores, or in case of the outage of the dish-soap, this low-level information may lead to a soap-shopping excursion. In this case, the person's action was initiated by the low-level detailed information (dirty dishes or no dish-soap), which triggered more contextual task goals (finish kitchen chores or go shopping). This process is much different from the top-down strategy in that, at the beginning of the top-down model, one sees the abstracted representation of the task without detailed information, while at the beginning of the bottom-up strategy, they focus on a small, local area of the task with much more detailed information that usually implies immediate action(s) to be performed.

Finally, in the opportunistic strategy, humans perform problem-solving tasks in a more unstructured way, and they make problem-solving decisions more spontaneously than in top-down or bottom-up strategies [50, 51, 99]. In this strategy model, both the top-down and bottom-up strategies are employed depending on the situation, and humans more freely shift the focus of the abstraction levels of the task. Guindon [50] suggests that this shift can be caused either by recognizing a partial solution in another part of the problem space, by dealing with inferred or newly added rules and requirements, by moving among partial solutions, or by alternating problem specification and solution development.

2.3 Conclusion

In this chapter, I discussed task structures of and existing systems for general MMA processes, and surveyed literature on problem-solving and task-planning, which are closely related to several issues observed in novice MMA tasks.

Most traditional types of MMA tools use the analogy of the original hardware devices to design their interfaces. As a result, the techniques used with the hardware counterparts have also been transferred to the new style of digital authoring processes, while a new group of interaction techniques have also been introduced that are unique to digital creation processes. The recent trend of user interface designs of MMA systems include simplified interface designs and allow users to skip some of the more complex steps of MMA tasks and (semi-)automatically generate multimedia data. There have also been systems that customize their interfaces based on kinds of tasks that users choose to perform, thus allowing users to view only subsets of full features.

Typical MMA tasks were analysed and decomposed into common subtasks such as importing media, creating new clips, and modifying clips. Each of these subtasks can be further decomposed, creating a hierarchical structure of an MMA task. In order to plan and perform general MMA projects, one needs to be able to construct an appropriate task structure by using the common subtasks and specifying their hierarchical and sequential orders.

The characteristics of problem-solving processes are also found in common MMA tasks. They are: identifying goals, assessing the current situations/states, recognizing the available options, selecting the next appropriate actions, and modifying the past actions. These characteristics were commonly seen during User Study 1 (as will be presented in Section 3.1), and they were observed to be linked to issues that novice users often experienced. A task performer executes these tasks by employing various strategies such as top-down, bottom-up, and opportunistic strategies.

The top-down strategy recursively breaks down the original problem into smaller sub-problems and solves these more manageable sub-problems at the lower level(s) of the task hierarchical structure. The bottom-up strategy, on the other hand, first focuses on more local and detailed levels where actions are usually performed directly on data. This focus is due to the fact that the bottom-up strategy is typically invoked by new incoming information at the low level in the task hierarchy, and actions are performed to accommodate this new data. The task performer gradually shifts the focus from the lowest level to more contextually meaningful levels in order to decide the future plans of the task. The opportunistic strategy is basically the combination of the top-down and the bottom-up strategies, switching between the two to whichever suits the given situation. Consequently, the scope of the task also changes based on the current strategy.

In the next chapter (Chapter 3), I closely examine the issues that novice users experience during common MMA tasks.

Chapter 3

Identification and Examination of Novice-Specific Challenges

In this chapter, I first present a user study (Section 3.1) in which I explored and compared approaches employed by users with different skill levels. I then introduce and define important concepts in this research such as *task-view* and *meta-tasks* (Section 3.2). By analysing novice difficulties in relation to these concepts, I will identify more general challenges that may be underlying the observed novice issues during multimedia authoring (MMA) tasks. Assuming these challenges are indeed the underlying causes of many of the observed difficulties, I develop a set of design guidelines that aim to address these challenges (Section 3.3), and therefore, also indirectly address the original novice issues.

3.1 Preliminary Studies (User Study 1)

My first goal was to understand what prevents novice users from performing MMA tasks in the same ways as experts. To explore this, I conducted a 3-part exploratory user study:

- Contextual observations of two novice and one advanced MMA users (Section 3.1.1)
- Interviews with seven highly experienced users (Section 3.1.2), and
- Surveys of eight novice users (Section 3.1.3).

Together, these studies allowed me to compare the approaches of novice and experienced users, and from this, identify some novice-specific issues so as to answer RQ1(a): *Are there common issues experienced by novices?*.

3.1.1 Observation of Novice and Advanced Users

3.1.1.1 Participants and Settings

I conducted observations of three users while they performed general MMA tasks. The purpose of these contextual observations was to explore how novice and advanced users perform MMA tasks, investigate the types of issues they might have, and observe individual variations in addressing the issues.

Among the three observed users, two were novice users, each with less than one year of experience with home video editing, and one advanced video editor with over ten years of experience. The two novice users performed a simple video editing task using Adobe Premiere Pro and Windows Movie Maker respectively, and the advanced user performed several post-production tasks of a film using Apple Final Cut Pro and Adobe Premiere Pro.

3.1.1.2 Procedure

During observation of the two novice users, the output screen as well as the conversation between the users and the observer were recorded for later transcription and analysis. For each session, two monitor screens were used: one for the user who was observed, and the other for the observer to view the activity. During the observation of the advanced user, only notes were taken. Two monitor screens were also used, but unlike the novice users, both the monitors were used by the user for the editing tasks.

All users followed Think-Aloud Protocols [76] to explain what they were doing and what issues they were experiencing. In order to clarify what the users were attempting in the observed tasks, I sometimes asked the users general questions about authoring tasks and tools and issues they have encountered. I also asked the novice users to perform certain operations to test their understanding of the task and to observe their capability of performing possibly unfamiliar actions.

3.1.1.3 Key Observations

Upon completion of this initial investigation, the following list of observations was compiled.

Novice Users

N1. Novice users did not understand all the terms.

Some common words (e.g., “sequence,” or “splice”) have specialized meanings in the context of MMA tasks, but these meanings are not standardized among different tools. For example, one of the users wanted to cut a data clip into two sections. This operation is done with the “razor” tool in Adobe Premiere, but it is referred to as a “slice” action in Sony Vegas and “split” in Windows Movie Maker. This non-standardization of terms interferes with transfer of learning between different MMA systems [25], and it can make it difficult for novice users to find proper commands from the menu items even when they know conceptually what needs to be done.

N2. Novice users were not always aware of ways to improve the editing environment.

One user was not aware of that zooming could be done to focus at a detailed level of data, and even when told, still did not know how to zoom in. Another user kept using a “docked” very small window of a file browser, which could have been resized or undocked to expand to a full size. Thus, even when novice users know how to perform certain tasks, there are cases where their performance could be more efficient if they knew how to customize their work environment.

N3. Interactions to perform basic actions were not always clear to novice users.

At times, there appear to be multiple ways to achieve a certain goal, though in fact, only one approach actually works, and novice users had difficulty identifying the correct one. For example, one user wanted to modify font faces of a title clip and tried a right-click menu, looked in a clip property window, and examined the "Title" menu but to no avail. This user was later told that it could be done by switching to the edit mode, which can only be started by double-clicking on a target clip. In other cases, it seemed that no clear solutions were visible to novices.

Sometimes prerequisite actions for certain tasks were unclear. For example, when a user wanted to record a new narration clip, the user was unable to come up with a sequence of actions required to reach the point of initiating an actual recording.

N4. Novice users often chose inefficient approaches to accomplish goals.

One user wanted to insert a picture to create a scene with a sequence of pictures, starting from one that was already in the workspace, switching to another (a new clip), and going back to the original picture. This user accomplished this task by first resizing the original picture, bringing in a new picture, and then copying and pasting the original one. Many tools allow users to directly bring in a new clip so that it overwrites a portion of an existing clip, creating the desired sequence of pictures very quickly and efficiently.

N5. Novice users did not understand the purpose of the GUI windows.

One user did not notice a file browser from which files can be directly dragged and dropped into the workspace. Instead, this user used the "File" menu and then selected "Import" to open a pop-up window.

The same user also did not understand the project media window, which lists media used in the current project. She could have used this window to identify data clips in the workspace rather easily instead of playing back several clips. These options would have been more economical than the approaches that this user employed.

N6. For novice users, recovering from unwanted situations was not always smooth, and the "undo" function could not be used when needed.

One user noticed some unwanted clips from an earlier task sitting on the time-line. This user first attempted to undo actions to see if any of the recent actions had influenced these clips. After undoing several actions, the user realized that these clips were not the results of recent actions, and thus resorted to manually deleting or relocating the unwanted clips.

N7. Novice users had difficulties recovering after an interruption.

One user expressed the desire to continue editing when a short break was suggested. This user feared that it would be difficult to remember what had been done. Both novice users had to skim through the workspace trying to recall where they had left off with their tasks after only a short (10-20 minute) break.

N8. Novice users often performed the editing task without planning ahead.

When asked how they would proceed with the editing task, novice users were not sure about their plans in advance. They tended to have short-term goals, often in reaction to the outcome of the previous goal.[1]

[1] While exploratory approaches for creative tasks may often lead to unexpected positive results, this may not always be the most ideal approach in MMA projects, especially when they are relatively large projects that unfold over a long period of time.

Together, these issues suggest that many of the novice users' struggles often occur even prior to actually performing any actions in MMA systems. Indeed, many issues are closely related to those observed in problem-solving and task-planning processes (discussed in Section 2.2). For example, issues such as N3, N4, and N5 suggest that these users are often unaware of the full range of possible options available [130]. Some other issues such as N6 and N7 are also closely related to users' ability to follow and modify the initial plans accordingly throughout the project. These observations pose quite a contrast to the behaviours of the advanced user.

Advanced User

A1. The advanced user was very articulate about what to do in relation to the overall goal of a large project (cf. N8).

Consistent with information gathered through interview of advanced users (as will be seen in Section 3.1.2), it was apparent that the advanced user had more concrete overall plans compared to the novices. Further, at the beginning of the session, this advanced user estimated time that would be spent working on this particular task, and described the tasks that were planned for the next few days. This is quite a difference to the findings of the observations of the novice users who often performed the editing task without initial plans.

A2. The advanced user followed established procedures for common tasks and was quick to generate new ones when necessary.

In addition to the overall plan described above, the advanced user also had short-term plans for many subtasks (chunking [26]). When the user did encounter an unfamiliar subtask, the user was able to quickly plan the next few actions and successfully perform them. This new set of actions then became a part of this user's repository of action sets.

Once the new subtask became part of this user's established procedures, they did seem to stick with that solution even when it was not the most efficient one. This phenomenon was observed when this user was colour correcting video clips.

A3. Recovering from unwanted situations was smooth for the advanced user (cf. N6).

Unlike the cases of the novice users, the advanced user showed very quick recovery from unwanted situations. For example, when there was no audio output, the advanced user was quick to both diagnose the problems and to come up with suitable solutions. Similar to the repository of action sets, the user had a repository of recovery options given common problem situations.

A4. The advanced user was able to use many features across multiple tools (cf. N5).

Not only was the user familiar with all the GUI windows and their associated functionality, but this user was also able to move back and forth between the two authoring programs seeking preferred options for various subtasks.

With the above observations, it is unsurprising that the advanced user was not tentative when performing actions, nor did the user need to explore menu items to locate necessary action commands. The advanced user had clear overall plans for the project as well as short-term plans for common subtasks, and easily navigated within and between multiple MMA systems.

Further Observation

- Both the advanced and novice users sometimes had difficulties filtering history of actions performed on particular data.

  Although it happened more often with novice users, even the advanced user had trouble finding data in the workspace. For example, the advanced user was trying to determine to which data clips this user had applied a specific set of colour correction processes. The corrections were subtle, and thus it was difficult to identify the corrected clips merely by watching them during playback. Furthermore, some colour correction had been applied to all the clips, thus identifying the corrected clips was much more complicated. The solution this user came up with was to remove the colour correction from all the clips and then reapply the desirable set of corrections to appropriate ones. This was probably a faster solution than trying to investigate each clip one by one.

All the observations from both the novice and advanced users suggest that multimedia creators not only need to perform operations within MMA systems, but they also need to be able to effectively perform certain tasks such as organizing operations (from Observations N3 to N8 and A1 to A3) and understanding specific terms and environment (from Observations A1, N2, and A4) in order to utilize system features and views.

Although the sample size of users was small, I was able to observe that the list of issues experienced by novice users is extensive. These issues observed in the contextual observations are consistent with the similar past studies by others [12,33,35,84], and were further complemented by the rest of the preliminary studies, as will be discussed below.

3.1.2 Interviews with Advanced Users

A total of seven people agreed to be interviewed, all advanced users, five of whom were professional media editors/creators, and two non-professional with extensive experience in using various MMA systems.[2] The purpose of the interviews was to understand how advanced users perform MMA tasks in their own environments, such as in TV production companies or their own home studios, and to identify issues they might have during the preparation and the actual production phases using existing authoring systems. These interviews also sought to determine what features of the software tools they wished to have but are not currently available. Further, it was my expectation that understanding how advanced users perform general MMA tasks and approach known issues could point to issues that novice users may often face during these tasks and help to determine the differences in the approaches taken.

All the interviewees answered all the questions, which are listed on Appendix A.1.3. Two participants, however, did not have any additional comments on several of the questions posed.

3.1.2.1 Interview Answers and Discussions

The following are selected answers to the interview questions and the insights that were gleaned from them. I have extracted 41 key answers and they are summarised under categorical headings[3] for the purpose of analysis. The numbers in parentheses indicate the number of times that the topic was mentioned.

Affordance for interactions (10)

With virtual objects (7)

- "I don't like horizontal faders"
- "Only thing that bugs me at times is *virtual knobs* [...] manipulating knobs with a mouse never feels right to me"
- "Click on a region, zoom in and out real quick with a flick of my mouse-wheel"

[2]More detailed background information of the participants and their tools regularly used are described in Appendices A.1.1 and A.1.2.

[3]These categories are listed in order of the number of times that the topic was mentioned.

- "No, I can't use these features with a mouse"
- "Having suffered with really sore *mouse-wrist*"
- "Draw automation envelopes and quickly enter fades with various curves"
- "Acid allows you to *collapse* tracks into a folder track when you're not working on them"

With physical objects (3)

- "Shuttle jog dials, fader boards, outboard controls"
- "HUI (Mackie) [. . .] but not incredibly helpful to me on a daily basis"
- "Touch-screen would be nice"

Affordance issues directly affect operations in MMA systems for both novice and advanced users; however, advanced users often know alternate ways to perform the same operations [40], as also seen in the observation sessions (Section 3.1.1). For example, advanced users may use keyboard short-cuts and menu items when the provided controls are not easy to use.

Support for clear feedback (6)

Advanced visualization of data such as rhythms, transcription, spectral information (5)

- "In all honesty I don't think it [different visualization] would help me much"
- "for rhythmic content some sort of visual representing beats in a clearer way would be pretty handy"
- "transcription would be most helpful [...] if you could edit individual notes on a musical transcription"
- "take advantage of some of the newer technologies like hardware graphics acceleration"
- "spectral information would be nice"

Lack of information between data and other entities in workspace (1)

- "have to do a lot of back and forth referencing things"

These comments indicate that the types of visualizations provided by some tools are sometimes insufficient and different users wish to have specific types of feedback from the systems. As seen in Norman's classic seven stages of action [95], how effectively and efficiently users perform actions in MMA systems can greatly depend on the evaluation stages of actions, and therefore useful feedback can influence the outcome of their task performance.

Visibility of functionalities (5)

Clutteredness (3)

- "they can be cluttered"
- "I do like an uncluttered, easy to read interface with a logical intuitive interface"
- "Cubase's interface just looks a cluttered mess"

User unfriendliness (2)

- "when I first started using it [Soundscape] I found it to be very user UN-friendly"
- "I've tried Cubase, Sonar, Nuendo, EnergyXT, and a couple of other, but I just can't get into them. It's definitely a user interface issue"

These answers indicate that the appearance of existing systems can sometimes hinder systems' functionalities. This can be more troublesome for novices, who sometimes struggle to locate necessary functions, as described in the previous section (Section 3.1.1). While experienced users may have past knowledge and skills that can be applied during an approach called *exploratory learning* [105] of a program, it can be hard for novices without relevant experience in similar MMA systems to do so [8, 29, 96].

Mapping of functions to real-world metaphors (4)

Use of hardware metaphors (4)

- "paradigm is useful, but only to the point where it doesn't limit what can actually be done by the computer"
- "as I come from an analogue background, it is helpful to have the digital screen emulate analogue boards. Sometimes make no sense"
- "having grown up with the hardware, I find it comforting"
- "the mixer screen in logic closely resembles what you'd expect to see on a hardware mixer"

Metaphorical frameworks can help [95,104] as long as they do not interfere with functionalities of the system, and users can actually relate the metaphors to the functions of the digital tools. All the interviewees were familiar with the original hardware devices, from which digital skeuomorphs were created, and thus the metaphors were found to be helpful. For novices who do not possess the background to interpret the metaphors, they are not useful and can possibly be detrimental as presenting emulated interfaces can clutter the screen and it is awkward to use tiny controls with mice [35].

Consistency of features and interaction methods (4)

Across tools (3)

- "I find the sample edit window (or the audio editor) in Logic to be practically useless—it's clunky and outdated when you compare it to say, Sound Forge."
- "I work faster on audio editing in Vegas (and Acid) than I do in anything else (i.e. Pro Tools, Logic)."
- " Logic is much better for music creating re: midi interface and internal libraries. Either one is good for editing. Basically ProTools is best for recording and Logic is best for interfacing (MIDI) and production."

Within a tool (1)

- "Logic seems to have a lot of old *legacy* code still attached to it [...] some of the features and menus are in rather strange places and not that intuitive to find."

These comments illustrate that consistency can be an issue within a single MMA system as well as among different systems. While it is probably inevitable that different systems provide different sets of features for various reasons (e.g., copyright, commercial competition), not having standardized interfaces and interactions can greatly affect users' ability to transfer their skills and knowledge from one task to another, or from one tool to another, even for advanced users (transfer of learning [25]). Again, while advanced users tend to have more resources to rely on, such as asking colleagues and general experience and knowledge of MMA tasks, inconsistency can cause significant difficulties for novice users.

Annotation features (4)

Existing and ideal annotation methods (4)

- "I want to be able to write everywhere, as if it's a scrapbook"
- "Soundscape allows you to do this but I never use this feature ··· too time consuming··· the old pen and paper is easier for me"
- "Being able to directly draw a circle around something would be pretty cool"
- "If you could have a sort of *marker pen* that you could scribble coloured notes on a *clear layer* over your project"

It is not surprising that many users expressed their desire to have different types of annotation features for their work environment. In early informal user observations, many advanced users kept notes on paper to keep track of the progress of their work, keeping a "to-do" list, notes on different data clips, and so on. While many existing MMA systems usually provide mechanisms to allow users to attach notes to data clips (e.g., naming clips, add a short description, change colours, etc), annotating their workspace in a flexible manner such as mentioned in their answers is yet to be realized. Although systematic usability testing will be required to support speculation that new styles of digital work-space annotations could possibly enhance their productivity over paper-based methods, having extra information about the task and data can likely be beneficial for generating and maintaining an appropriate task plan.

Constraints of interactions (1)

Lack of constraints (1)

- "flexibility is the key. unfortunately, flexibility also adds complexity"

This quote from one of the interviewees nicely describes the design principle of "constraints" [95]. As many of the MMA systems have been produced to accommodate advanced features and techniques [4,21,33,35,112], it can be difficult for novice users to learn and use these systems. Although this interviewee noted this difficulty of using these systems, as seen in the observation sessions (Section 3.1.1), it is the case that advanced users can usually narrow the search space for relevant actions effectively, while novice users often struggled with finding their way around in complex MMA systems.

Work-Flow Patterns and Approaches (7)

I was able to obtain answers from all the interviewees regarding typical work-flow. It is important to note that each of these advanced users was articulate about their work-flows, and their answers were detailed. For example:

> *"[...] first 'spot' what I'm going to write music for using Acid or Vegas to give myself audio cues, then I import that into Logic and compose music without the video. Then I keep a 'master' file in Vegas and add music cues*

> *to it as I go along. Since I can't see what I'm doing when I'm in Logic, I have to do a lot of back and forth referencing things."*

This was also recognized in the observation sessions (Section 3.1.1). These answers indicate that advanced users have their own routines at various stages of a project, meaning within a big work-flow, they also have a set of smaller work-flows, or subtasks, that are independent from a specific project. These subtask work-flows can often be reused and reconfigured at different stages of different projects. Discussion of MMA task decomposition is provided in Section 2.1 in more detail.

What is important in the observations here is that these subtask work-flows often help advanced users to quickly structure and plan new projects at higher levels of abstraction. This is consistent with prior findings that experts employ task-decomposition processes more often than novices [59].

These interview answers highlighted issues of existing systems: cluttered interface, use of metaphorical frameworks (often unfamiliar to novices), tiny controls, unclear feedback, and inconsistency of features within and between systems. All these issues can contribute to difficulties novices face during MMA tasks. Furthermore, all the advanced users were able to clearly state their work-flow patterns and typical approaches that they would normally employ. In other words, they have a concrete mental model of a task, which is essential for successful task performance [130].

3.1.3 Surveys of Novice Users

I conducted surveys to gather information from eight novice users. The purpose of these surveys was to understand how these users approach unfamiliar MMA tasks, to determine what types of support they might need and to explore how their approaches differ from those of advanced users. Each survey consisted of a set of questions shown in Appendix A.2.1, and they were administered through a web-site.

A total of eight novice users responded and answered the surveys. Two people had less than one year of experience in both video and sound authoring or editing, four had less than one year of experience only in video editing, two had no experience either in video or sound authoring.

3.1.3.1 Analyses of the Survey Answers

The following is a summary of the answers to the survey questions.

Answers for Question 1: Preparation and initial steps of MMA tasks

Question 1 asked about the ways that novice users plan their MMA tasks. Do they only identify the first few steps of the project as seen in the observations, or do they actually see a bigger picture? Although the respondents had only minimal experience in MMA tasks, the answers to this question indicate a rather interesting tendency. While novice users did not exhibit the ability to adequately plan ahead and successfully construct concrete work-flows in the contextual observation of users (Section 3.1.1), answers such as “try to think about the structure of the video and audio” and “review what is needed” illustrate that novices do have the intention and possibly the ability to conceptualize their project at an abstract level when they are not faced with the actual authoring tools. This implies that when they are explicitly asked to think about their plans (or provided with appropriate tools), novice users may also be able to plan explicitly for MMA tasks.

Answers for Question 2: Most likely scenarios

Question 2 was designed to see if there was any pattern in regards to which specific strategies or approaches are employed. The following description of each scenario will indicate how novice users may approach MMA tasks. Note that choosing any of the scenario options does not of course guarantee that one would actually perform those tasks exactly in the ways that are described, but it did provide some insights on patterns novices use to approach similar tasks if actually given a choice.

Scenario 1 describes a situation when the multimedia creator gets inspiration for a story-line by first collecting and analysing available data, instead of coming up with a story-line first. This approach is often referred to as the data-driven approach.

Scenario 2 starts with the more abstract level of coming up with an overall story, and then looking for suitable data. This approach implies a typical task-reduction process, as it reduces an abstract but large problem (a story-line) down to more manageable and smaller problems (find data clips that suit each part of the story-line).

Scenario 3 may sometimes be employed in bigger productions. The final timeline is divided into a number of sections in order to work on each section one at a time. While this scenario implies a kind of task-reduction process, and it does start with an abstract idea like Scenario 2, to come up with specific sectioning ideas (e.g., scenes), the user would actually need to begin with a more concrete storyline at the beginning compared with Scenario 2. This means that the process would likely require shifting the focus from abstract to more detail-specific levels and switching between different scenes and tasks.

These scenarios represent generic MMA tasks. They involve processes such as task-reduction and switching between different levels of abstraction relevant to a specific task. These processes are closely related to characteristics of the problem-solving and task-planning processes.

Six out of eight people answered that they either “would probably do that” or “would almost certainly do this way” for Scenario 1, and seven out of eight did for Scenario 2. However, six out of eight people answered that they would “unlikely to do that" for Scenarios 3. These results were expected as Scenario 3 would involve more careful planning than the other two scenarios, and without relevant experience, it might seem more demanding. These initial findings show that the first two scenarios were liked more or less equally by the novice users. Likely, the demands of Scenario 3 seemed too great without sufficient knowledge and understanding of MMA generally. This knowledge is especially pertinent to maintaining the overall process when switching between different sections or changing approaches.

Answers for Question 3: Information to be available

Question 3 asked users to investigate and evaluate informational lists that could be displayed when working on a project. These lists provided relevant informations about the workspace as well as a specific task itself. In addition, users were asked to make additions that they thought might be beneficial.

The lists that were provided were a compilation of informational lists common to most MMA tasks based on the insights gleaned from the interviews (Section 3.1.2) as well as the contextual user observations (Section 3.1.1).

The types of informational lists provided are identified in Table 3.1. The informational lists that were most highly rated were: "List of Goals/To-do list" (six out of eight), "List of actions performed (as in undo list)" (seven out of eight), "Project media and their original locations as well as where they are used in workspace" (five out of eight), and "Diagram indicating project stages and current stage" (five out of eight). Most participants identified these informational lists as ones they would "definitely would like to see" (six, seven, five, and five, respectively, out of eight responded). Of these four highly rated items, three are features that allow users to keep track of the current status of the work in relation to the goal(s) of the entire project. These results are also consistent with additions that are suggested by some participants: two requested some sort of progress indicator in the project: "view the completed parts" and "project so far (progress)," providing a progress indicator tool within an entire task structure would accommodate this request.

Table 3.1: Informational lists that could be displayed when working on a project. The numbers in parentheses indicate numbers of participants' answers to the question if they would like to see them during MMA tasks.

Information	definitely	maybe	not sure
List of Goals/To-do list (e.g., colour correction for the opening, re-mix audio, etc)	6/8	2/8	0/8
List of actions performed (as in undo list)	7/8	1/8	0/8
Project media and their original locations as well as where they are used in the workspace	5/8	1/8	2/8
Short notes for each editing action (e.g., readjusting clip 1, etc)	3/8	4/8	1/8
Diagram indicating project stages and current stage	5/8	3/8	0/8
Preview of actions before actually executing them (i.e., play short animation/video of how to perform certain things)	4/8	4/8	0/8
Review of actions to remind yourself what you did (play animation/video of what you did, similar to preview)	3/8	4/8	1/8
Description of each data/cut/clip (e.g., clip 1 used for the explosion scene, etc)	1/8	5/8	2/8

Answers for Question 4: Difficulties using existing tools

The following is the list of reasons for difficulties that the respondents had with the MMA tasks.

- "Things were not clear."
- "There were too many functions."
- "The program looks very crowded."
- "The interfaces for these tools are too disjointed (overly modular) with idiosyncrasies."

- “Limited documentation on how to use the software.”
- “Little prior experience on how we should edit the video step by step.”
- “Maybe requires a lot of new terms and knowledge to know how to use it.”
- “Not able to understand all the functionality.”
- “It’s too complex sometimes.”
- “The biggest obstacle was that it was hard to think creatively because I was just unaware of the options.”
- “Perhaps a list of suggestions or possible actions would be helpful.”

What these responses have in common is the lack of a foundation and a basic working knowledge, and users may not even know where to begin. Such conditions might also prevent them from forming an appropriate mental model of the task, which is a crucial step for successful task performance [130] as previously discussed.

3.1.4 Summary of the Preliminary User Studies

I conducted a series of user studies in order to compare the approaches of novice and experienced users, and to identify issues that might be preventing novice users from performing MMA tasks in the same ways as experts. Based on the results from these studies, there are in fact certain issues experienced by novices, positively answering RQ1(a): *Are there common issues experienced by novices?*

The observed behaviours of MMA users showed that the advanced user was able to utilize MMA systems effectively, and that novice users struggled or were not effectively performing operations in MMA systems. For some, the struggle began prior to actually performing any MMA tasks. I found that there are general tasks that advanced users perform more effectively than novice users. These tasks appear more obvious and/or explicit for advanced users in regards to when and how to perform them, while these tasks seem less explicit for and often obscured from novice users.

Likely causes of this phenomenon involve novice users' lack of knowledge and experience and complex interface designs.

The interview with advanced users illuminated some potential issues hampering novices from performing essential tasks. These issues include cluttered interface designs and workspaces, unfamiliar metaphors, lack of useful feedback, lack of constraints accommodating certain advanced features, inconsistency among and within MMA systems, and possibly lack of support to generate and maintain appropriate work-flow of MMA tasks. While advanced users are often able to overcome these issues by taking alternate approaches, it can be difficult for novice users to do so.

The analysis of the survey answers also provided insight into issues novices face during MMA tasks and possible approaches to address them. One of the findings was that, when novice users were explicitly asked to think about their plans for an MMA project, they did in fact show elements of an initial planning stage. The survey responses also indicated that many users were interested in a progress indicator tool. The responses suggested that novices may lack foundational and basic working knowledge, which would prevent them from forming an appropriate mental representation of the task. This may be one of the reasons why they were generally interested in a type of a feature that could help them generate and maintain a mental representation of the task such as would be represented in a progress indicator.

Having presented the observed issues experienced by novice users in User Study 1, in the rest of this chapter, I will further explore these results with the intention of identifying possible challenges underlying these observed issues.

3.2 Meta-Tasks in Multimedia Authoring

In this section, I will first discuss how known problem-solving strategies are employed during MMA tasks in relation to actual operations performed in authoring tools (Section 3.2.1), and then define important properties relevant to structuring an MMA task (Section 3.2.2). I will then introduce and define a new concept of *meta-tasks*, and provide examples of meta-tasks observed during User Study 1. Finally, I will describe potential novice-specific challenges in relation to these meta-tasks (Section 3.2.3). This section, therefore, proposes an answer to RQ1(b): *Are there common challenges underlying novice issues identified in RQ1(a)?*

3.2.1 Problem-Solving and Task-Planning Strategies in Multimedia Authoring tasks

I will now describe how users may employ known problem-solving strategies (top-down, bottom-up, and opportunistic strategies as discussed in Section 2.2) in relation to common MMA operations.

Top-Down

In a top-down strategy, one solves complex problems by first setting an overall goal, then dividing it into smaller, more manageable sub-goals. It is often employed when one can start with a general high-level plan of a large project [32]. For example, users may start by sketching out a very rough plan of the project they are working on. The users may then determine the various media materials that are needed. From there, they can look for particular media, think about how these media should be edited, and so on, thus working toward more detailed aspects of the project.

A top-down approach is also often employed when reviewing the progress of the project: users may look back to see where the current stage fits in relation to the overall goal state (i.e., by revisiting more abstract levels). They may then examine details of what needs to be modified (i.e., by investigating more details). Instead of storing each detail in working memory, the users can reduce the memory load by dealing only with abstract information during the planning phase or when reviewing work in progress [49].

Bottom-Up

In a bottom-up strategy, one first attempts to find a solution to a local problem and generalises this solution to address other problems. This strategy is often used to accommodate changes that happen during a project. Hoc [54] defines three key mechanism invoked in this strategy: *plan recovery*, *plan revision*, and *plan abstraction*. Plan recovery is a mechanism in which a pre-learned action sequence is triggered by new incoming data. Plan revision occurs when the task-performer finds and modifies flaws in the original plan upon examination of the finer details. In this way, the task-performer adapts the plan to the new data. In the absence of plan recovery (i.e., the task-performer is unable to find a suitable pre-learned action sequence), plan abstraction may be initiated. Plan abstraction means that a new plan is constructed to address the current problem at a detailed level, and then the task-performer applies it to address more generic problems.

For example, a plan revision may be observed when the editor finds a new media clip that may be more suitable for the current project. He or she might tentatively replace the old clip that is already in the workspace with this new one. This replacement consequently forces a sequence of operations to accommodate the new clip into the current workspace, such as adjusting the length and changing several parameters of both the new and the other existing clips. The task performer initially reacted to new information/data (i.e., the newly found clip), and then shifted focus to a more contextual level to see whether or not additional actions were necessary (as in this example).

The bottom-up strategy may also be employed during exploratory phases of MMA projects. Before the task performer decides on a concrete plan for the project, he or she may experiment with different types of video clips. In the previous example of inserting a new clip, the user may audition different clips before making a decision. For this process, the user may insert each clip into the workspace and perform a small set of actions to evaluate how this new clip fits. If the user is not satisfied, then the process is repeated many times. Again, the focus was initially at a detail level and then moved to more abstract levels.

Opportunistic

The opportunistic strategy is a combination of both the top-down and bottom-up strategies and it provides more flexibility on how users perceive and perform tasks. A task performer switches planning strategies and perspectives, seeking solutions to accomplish the project goals. Initiating these shifts in strategy and focus can be troublesome for users who are unfamiliar with MMA projects. In general, existing tools allow their users to perform tasks using the opportunistic strategy as they are designed to accommodate advanced features and processes that involve such shifts as is typically required by experienced users.

Although any combination of top-down and bottom-up approaches may be considered an opportunistic approach, there are common situations in MMA projects that compel shifts between these strategies and abstraction levels. Guindon [50] suggests that a shift can be caused either by recognizing a partial solution in another part of the problem space (Case 1), by dealing with inferred or newly added rules and requirements (Case 2), by moving among partial solutions (Case 3), or by alternating problem specification and solution development (Case 4).

In MMA tasks, for example, when a user is looking for a new video clip for the opening section of a video, the user may accidentally discover a number of clips that may be used for a different section that he or she was working on earlier. The user may then halt the current task and resume a task that was paused earlier. In this case, the user has found a partial solution in a different part of the problem space (Case 1). In another case in which a user is working on a production of a TV show, he or she may get some requests for additions from sponsors or the producers of the show. The video editor may then be forced to deal with these new requirements and may have to modify some parts of the video (Case 2). MMA projects usually require many steps to refine the products to achieve an ideal outcome: for example, a video editor typically works on multiple parts simultaneously such as the introduction and ending segments in which similar video effects are required (Case 3). In this refinement process, video editors repetitively alternate the processes of problem specification and corresponding solution development until they are satisfied with their work (Case 4). Note that while each of these situations may often employ known strategies of problem-solving and task-planning, they are actually independent from any particular MMA systems.

3.2.2 Properties of Task-View Construction

As discussed in Section 2.2, both generating and maintaining an appropriate representation of a task are two important processes for successful task performance [130]. Thus, I define a *task-view* to be a (full or partial) representation of the structure of a task. It consists of subtasks and specifies hierarchical and sequential relationships of these subtasks.

The notion of this hierarchical structuring of a task is consistent with *activity theory* [92,121,126], in which an activity can be conceived as an aggregation of actions. Each of these individual actions, in turn, can then be conceived by a set of lower level operations. Suchman [121] states that this organization of actions is defined as a result of moment-by-moment interactions between actions and the environments of their actions. For example, a very abstract task-view for a complete MMA project might consist simply of the steps:

1. Find media
2. Put them in sequence

A more refined but partial task-view might elaborate on Step 1 consisting of a set of subtasks as follows:

1.a) Find video clips used for the opening scene
1.b) Find sound effect clips for the opening title
1.c) Find a music track played during the opening credits

Task-views can include a complex combination of both abstract and low-level tasks as well as various hierarchical and sequential relationships between these tasks. Thus, a task-view can be thought as a cognitive snapshot of the user's conceptual model of the task. This snapshot can adapt slowly over time as the user refines her goals, but it can also change quickly as the user switches focus to a different aspect of the overall task.

Task-views possess the following properties, *scope*, *abstraction levels*, and *constraints*, which I describe below.

Scope As noted earlier, humans have difficulties generating and recalling the entire task structure when the task is large and complex [130]. To compensate for this limitation, people generally focus on only a small portion of a task at a time [32]. In other words, they narrow the *scope* of the task-view, whether implicitly or explicitly. The scope can vary: the full-scope may be useful when users need to examine the overview of the project structure. On the other hand, a partial or local scope is more efficient when it is necessary to focus on details such as when the user concentrates on the introduction segment of the video.

Abstraction Levels Humans also simplify a task-view by eliminating detailed information from it, and dealing only with its *abstract* representation [107]. For example, instead of specifying each action (e.g., *copy*, *paste*, *group*, *set pans*, *adjust volume*, etc), a sequence of actions may be grouped into a more abstract but contextually meaningful subtask unit (e.g., *Edit a video clip*, *Apply sound effects*, etc). By omitting detailed information from a task-view, one deals with less information at one time.

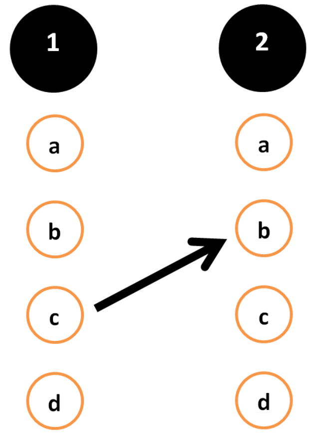


Figure 3.1: A task with two high-level steps (1 and 2, indicate by large black circles), where each step consists of a few subtasks (a to d, indicated by smaller circles).

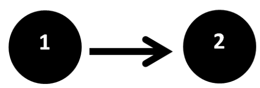


Figure 3.2: A high abstraction level of the task.

Constraints Large tasks have complicated networks of constraints. For example, certain low-level subtasks must be performed before others, which in turn may imply that certain high-level tasks must happen before others. Thus, for a task-view, there is an interaction between the abstraction level, the scope and the relevant *constraints* induced by the partial-order among the subtasks.

For example, consider a task with two high-level steps, where each step consists of a few subtasks, as shown in Figure 3.1. Suppose that a,b,c, and d can occur in any order within each step, but that 1(c) must come before 2(b). If the abstraction level is high such that the task-view only includes Steps (1) and (2), then the constraint that "Step 1 must precede Step 2" is discernible (shown in Figure 3.2). However, if the abstraction level includes (a)-(d) but the scope only includes Step 1, then no constraints will be discernible (shown in Figure 3.3).

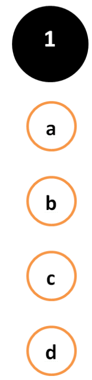


Figure 3.3: A narrow scope of the task with only Step 1.

3.2.3 Meta-Tasks in Multimedia Authoring

In User Study 1 (Section 3.1), I observed that novice users experienced difficulties that were different from those experienced by experts. Based on an analysis of MMA and a survey of the problem-solving literature (in Chapter 2), I hypothesise that novice difficulties can largely be explained in relation to what I define as *meta-tasks*.

> **Meta-tasks** are those tasks that users perform to (1) construct, (2) modify and (3) utilize an appropriate task-view. Task-view is a full or partial representation of the structure of a task. It consists of subtasks and specifies hierarchical and sequential relationships among these subtasks.

Meta-tasks relevant to MMA are related to human problem-solving more generally, and involve aspects of familiar problem-solving strategies (top-down, bottom-up, and opportunistic strategies) as conceptualized in standard cognitive models (discussed in Section 2.2). Consider the following sample scenario.

> **Scenario**. *A user needs to insert a new video clip. To do this, the user realizes that the clip must first be modified, then put on the timeline, and finally its exact placement must be reviewed. To modify the clip, the user*

will need to change some of its properties (e.g. opacity, speed and duration).

Task-View Construction: In this example, “modifying” the clip is a high-level task, and “changing the duration of the clip” is a lower-level task. A meta-task in this example, however, refers to the process whereby the user expands a high-level task (such as modifying the clip) into a partially-ordered set of lower-level ones (such as changing opacity levels, clip speed, and clip duration). Another example of a meta-task is determining the order in which to perform these lower-level subtasks.

Task-View Modification: Yet another meta-task is the (possibly unconscious) choice of whether and when one will switch between attending to the high-level tasks and the low-level ones: for example, at the beginning of the process, the user may be focused on the general need to modify the clip, while later on, he or she may be fully absorbed in details of a specific property such as opacity. This need to move between higher and lower levels of tasks is itself an example of how the task-view changes dynamically.

Task-View Utilization: Given a task-view, another common meta-task is to translate conceptual task goals represented in the task-view into actual commands within an MMA system. Once the user determines his or her focus is on modifying the opacity of the clip, then the user needs to figure out how it could be done within a particular MMA system (e.g., using a pop-up menu). The utilization of a task-view, therefore, involves processes of relating the given task-view to data and interface of a MMA system.

Users have different levels of fluency with these meta-tasks. For example, consider the meta-task of subdividing a task into more detailed steps. An experienced user will find this easy because they have already done this process so many times, that they no longer need to solve this as a problem: they already know the solution, i.e., the sequence of detailed steps that accomplish the task. They can recall the solution rather than needing to generate it. Thus, this meta-task appears more obvious for an advanced user than for a novice one.

Note that the term *meta-task* itself is not new and has been used in certain contexts such as scheduling computational jobs or general task management [18, 124]. The concept of meta-tasks in the current work is, however, defined specifically as the above processes that deal with users' task-views (i.e., task-view construction, modification, and utilization). Task-views can change dynamically as the user's perspectives of the environment change throughout an MMA project, and thus meta-tasks are carried out at different stages of the project.

Classes of novice behaviours

In the following list, I summarise and organize observations from User Study 1 into classes of related behaviours. The study results from which they are taken are indicated in parentheses. Each of these classes can be associated with a set of meta-tasks. I list here a proposed set of meta-tasks for each class of behaviours.

C1. Novices did not know what actions needed to be performed.

Observations

- When a desired action was unavailable, novices often did not know what actions needed to be undertaken to make the desired actions performable. (Observation)
- Even when they knew what they wanted to accomplish, novices often could not come up with an appropriate/efficient sequence of actions to accomplish it. (Observation)

- "The biggest obstacle was that it was hard to think creatively because I was just unaware of the options [of actions]. Perhaps a list of suggestions or possible actions would be helpful." (Survey)

Meta-tasks

M1. Determine possible next actions (a sequence of actions/subtasks).

M2. Translate a conceptual goal into a goal within the MMA system.

C2. Novices did not have concrete project plans

Observations

- Advanced users could state their typical work-flows clearly and promptly, but novices tended to be unclear and/or disorganized. (Interviews/Observation)
- Novices started tasks in a more exploratory manner, but advanced users planned what to do. (Observation)
- Many showed a tendency to choose more structured/well-planned approaches to authoring tasks, even when they have little or no experience. (Survey)
- Many expressed the need for an overview of the project in addition to the typical undo list. (Survey)
- When undoing several actions, the users typically undid one action at a time to see the effect of each action, instead of perceiving a group of actions that are constitutive of a certain task. (Observation)

Meta-tasks

M3. Parse larger goals into viable subtasks (task-decomposition).

M4. Generate overall task/work-flow.

M5. Determine/traverse contextual/abstraction levels and scope of tasks.

C3. It was difficult for novices to recall the current state of their project in relation to their goals.

Observations

- Task interruption led to disorientation in regards to where the users had left off. (Observation)

- When resuming work after leaving it for a certain period of time, novices took a long time to recall where they left off. (Observation)

Meta-task

M6. Maintain overall task/work-flow and be aware of the work-flow location in the task-view.

C4. Novices did not have many pre-learnt options and/or did not utilize them.

Observations

- For novice users, recovering from unwanted situations was not always smooth, and the "undo action" function was often unavailable. The advanced users, however, were quick to both diagnose problems and come up with suitable solutions. (Observation)
- Novice users had less pre-learnt recovery options given common problem situations compared to the advanced users. (Observation)
- Methods unique to the MMA paradigm are sometimes needed to perform certain tasks. (Observation)

Meta-task

M7. Recognize when and how to transfer and apply solutions from a familiar situation to a new situation (as often observed in the bottom-up strategy).[4]

C5. It was difficult for novice users to understand and obtain information about data in the workspace.

Observations

- Locations of data clips in workspace are often obscured and not clearly marked. (Observation/Interviews)
- The users often had to search for a target data clip by opening many clips in the workspace one by one identify details. (Observation)

[4] Although this task may not appear to be a meta-task at first, the process of applying solutions from one situation to another is equivalent to transferring a partial task-view of one project to a part of another task-view. Thus, this is a meta-task involving the construction and/or modification of a task-view.

- Many expressed that it would be nice if they could label/annotate their data in the workspace. (Interviews)

Meta-task

M8. Maintain contextual meaning of the data clips in relation to performed tasks.

C6. All the features provided by the MMA systems were not fully absorbed by novices.

Observations

- Novices only used a few functions and dealt with one or two main windows within a tool, while the advanced user even went further and explored multiple tools to find ways to maximize functionality. (Observation)
- "There were too many functions." (Survey)
- "Not able to understand all the functionality." (Survey)
- "The program looks very crowded." (Survey)
- "The things were not clear." (Survey)
- Once users found one way to accomplish a task, they tended to stick with it regardless of efficiency. (Observation)

Meta-task

M9. Understand and know common and specific features of MMA systems.

Novice-Specific Challenges

Drawing on the analyses of MMA tasks and tools (Section 2.1) and of problem-solving and task-planning strategies (Section 2.2) as well as the theoretical framework of explaining the observed novice-behaviours in regards to the meta-tasks described above, I identified a set of potential novice-specific challenges underlying the observed novice issues. Table 3.2 shows these novice-specific challenges and the underlying meta-tasks that users need to be able to perform to overcome the challenges. The relationships between the challenges and the meta-tasks are rather complex, and in order for a novice to overcome each challenge, more than one type of meta-task is typically involved. For example, one of the novice-specific challenges is "Identification of concrete task-flow" (Challenge-1), and this can be explained as difficulties of performing multiple meta-tasks (Meta-Tasks M1–M6): e.g., difficulties with determining appropriate sequence of actions (M1) or with performing task-decompositions effectively (M3) can pose a challenge of identifying a concrete task-flow (Challenge-1).

This section (3.2) focused on a discussion of meta-tasks, which are tasks that users perform to construct, modify and utilize an appropriate task-view. While advanced users effectively perform meta-tasks, I hypothesise that difficulty with meta-tasks can lead to challenges that often prevent novices from performing MMA tasks in the same ways as experts. These findings positively answer RQ1(b): *Are there common challenges underlying novice issues identified in RQ1(a)?*. Based on this theoretical framework of explaining novice-specific challenges in regards to meta-tasks, I will develop a set of new design guidelines for MMA support tools in the next Section (3.3).

3.3 Design Guidelines for Multimedia Authoring Support Tools

One of the goals for this work is to allow novice users to effectively utilize existing MMA tools, and specifically, to do so by assisting them with construction, modification, and utilization of their task-views. I hypothesise that identified novice-challenges have underlying explanations in terms of meta-tasks. If this is true, then helping with the meta-tasks would alleviate a significant obstacle for novices using MMA systems. I therefore:

1. propose guidelines aimed at addressing novice-challenges that stem from meta-task difficulty.
2. design a system that satisfies this set of guidelines.

Table 3.2: Novice-specific challenges and their underlying meta-tasks that need to be performed.

Novice-specific challenges	Underlying meta-tasks
ch1. Identification of concrete task-flow (natural for experts, but elusive for novices).	M1, M3, M4, M5, M6
ch2. Maintaining a mental model of a task and retaining contextual meanings of tasks (experts can do this, e.g. they utilize features to add notes, while novices struggle).	M5, M6
ch3. Understanding relationships between data clips and performed operations (not clear for novices).	M5, M8
ch4. Recovery from error (much smoother for expert users than for novices).	M1, M5, M7, M9
ch5. Recognizing dependencies: novice users had difficulties understanding what other actions needed to be performed prior to their target operation.	M1, M2, M5, M9
ch6. Recognizing combinations of actions needed to accomplish a certain goal (difficult for novices).	M1, M3, M5, M7, M9
ch7. The contextual information of the task is not available in current tools.	M2, M5, M6, M8
ch8. Task interruption could leave users disoriented when they return to the system.	M1, M5, M6

Thus, these guidelines and a new MMA environment designed based on them address RQ2(a): *What are possible solutions to provide viable support?* As part of the environment, I also propose an explicit and dynamic task-view visualization: this is not meant to replace the user's internal task-view, but simply to help them construct and maintain their internal ones. The list below shows these design guidelines, organized roughly in the order to help users with the task-view construction, modification, and utilization.

3.3.1 Design Guidelines

Guideline (G1): Provide additional contextual information where possible.

One important aspect of providing a comprehensive task-view is that it adds more contextual information about the task being performed than what the standard interface of MMA systems usually provide: existing tools typically allow a view of previously performed actions in a linear sequential form (e.g., undo list) and have traditional menu structures that group actions in a relatively context-free manner. Contextually meaningful perspectives of a task are essential for executing a top-down strategy, which capitalises on abstracted perspectives of a task to plan and proceed toward any goal(s) [58]. It is also crucial for a bottom-up strategy as task-performers need to gain some contextual information to successfully employ this strategy despite that it initially starts with a bottom-level action(s) [54].

For example, given a task goal such as creating a music video, one needs to be able to first decompose this high-level goal into smaller sub-goals such as media import, clip layout, apply transition effects, and so on: however, this decomposition process is not directly supported by existing interface designs — menu items are organized based on generic action types but not based on these sub-goals, and previously performed actions are presented in a linear sequence (i.e., an undo list) but not organized based on subtasks that were performed.[5] Thus, existing MMA systems do not provide task contextual information that can be useful for performing this task-decomposition process, which is one of the necessary steps to generate an appropriate task-view.

Types of context to be provided may also vary depending on individual work-flows, types of projects, and which subtasks are being performed at a given moment. There is probably no single context that will work for all users for all occasions, and thus designers of such tools will be required detailed task analyses with specific target user groups. One way to provide task contextual information is to classify actions into contextual groups, and this is the primary method used in this research. However, there are no restrictions on how to provide such information. For example, allowing users to freely annotate task-views may possibly increase the likelihood of adding some crucial information about tasks [24, 128]. Indicating the relationships of data and performed tasks (i.e., which actions affected which data in what way) can be an asset as well, as deciphering their relationships was found to be one of the challenges for users with different skill levels.

Guideline (G2): Show task-decompositions and structures.

As discussed in Section 2.2, when performing problem-solving tasks, an effective representation of a task structure can help improve task performance. Although it would be quite helpful to have a visual aid similar to Figure 2.15 for the ToH puzzle, it is practically impossible to create a comprehensive state-space for MMA tasks, as previously discussed. What could be done, however, is to provide information on the structure of a task being performed: we can provide an

[5] A more detailed discussion of current support in MMA systems in this regard is provided in Section 4.2.1.

explicit task-view visualization that displays an appropriate task structure for an MMA project. In this way, it is possible for users to obtain useful information about the current state of the task, previously performed actions/subtasks, possible future actions/subtasks, and their hierarchical relationships.

For example, certain techniques and notations such as ConcurTaskTrees [19, 101, 102] could be utilised to display a hierarchical structure of a task and indicate the order and the concurrencies of subtasks. It can be used to describe tasks at different levels of abstraction as well as their temporal relationships.

Guideline (G3): Provide explicit task-views that can dynamically change to show different levels of abstraction, scopes of tasks, and task contexts.

Due to the nature of MMA processes, task-views for MMA cannot be static, unlike well-formed problems such as a ToH puzzle [130]. The flexibility of a task-view is important for MMA tasks as the users need to traverse different scopes of the task and various abstraction levels, which can also dynamically change depending on the perceived contexts of the task. For complex tasks such as MMA, humans often employ the opportunistic approach [50,51,99] (discussed in Section 2.2), which involves both the top-down and the bottom-up approaches, switching between them to seek suitable means to accomplish their goals. Being able to investigate possible solutions by utilizing multiple strategies as well as looking at the task from different perspectives (i.e., contexts) can, therefore, be an important asset of support tools of this kind.

Guideline (G4): Provide common task-structures that allow integration of newly performed actions.

Generating and maintaining an appropriate task-view are important steps toward successful task performance [130], but were found to be difficult for novices.

Thus, providing support for novices to achieve these steps is crucial. One approach is to provide scaffolding for novice users by presenting initial task-views that are common to typical MMA projects.

For example, as observed in the preliminary studies (Section 3.1), advanced users typically have certain work-flows for their own work/projects. We may analyse and utilize these known work-flows to first generate a task-view consisting of common subtasks and operations such as discussed in Section 2.1 and specifying a hierarchical and sequential order of them, so as to address the initial issue of generating a task-view. Novice users can initiate MMA projects by first referencing this task-view. As the task progresses, newly performed actions could be integrated into the task-structure thus addressing the second issue of maintaining an appropriate task-view.

Guideline (G5): Indicate possible next subtasks/actions.

While most existing systems somewhat accomplish this by disabling (i.e., greying out) certain menu items or icons to indicate action commands that are unavailable at a given time, there remains the issue of narrowing the search space for target actions. In order to address this issue, several approaches [23, 43, 86] have been proposed to display only a subset of the full list of action commands in the menu system. There is also an issue related to visible constrains of the task-view property as described in Section 3.2.2. Within a task-view is a network of constraints induced by the partial order among the subtasks, and these relationships among subtasks are not accommodated by the above approach of disabling menu items or icons.

These issues (i.e., narrowing the search space and visible constraints) could be addressed by providing a task-view visualization that provides contextually meaningful views to help users first narrow the search space of actions, and then see constraints on the order of actions and subtasks.

Guideline (G6): Provide a means to carry out actions within contextually meaningful views.

When users are performing tasks in unfamiliar systems, they show the characteristics of exploratory learning [105], which includes searching for menu items. When users are performing MMA tasks in certain contexts, such as working on the introduction of a video, or mixing the audio tracks for the concert video, the users need to look for particular commands from the menu system (or from the list of tool-bar icons). However, the menu items and tool-bar icons are not grouped based on contexts of the task, and thus the users need to be able to switch from the context of the task to the context of the menu groups. While advanced users can easily switch between these two contexts and narrow the search space of the menu items, novices often need to search through many menu items due to unfamiliarity with the system. This can likely be prevented if there is a mechanism to provide an alternate approach that maintains the task context from which users can find and execute target actions.

Lafreniere et al. [69] addressed this issue by providing a customizable interface based on the keywords that users specify. Instead of modifying the familiar standard interface designs, the approach in the present dissertation involves an additional support tool that integrates a dynamic task-view to provide the contextual information about the task so that the users can select and execute operations that are specific to the given task context.

Allowing direct execution of actions from a contextually meaningful view may not only help users to narrow the search space of menu items, but could also address the issues caused by the tool-specific idiosyncrasies such as different terms to describe the same operations and different ways to perform them. As well, as noted earlier, the context to be provided should be adjusted accordingly, thus planning and performing the same actions can be accomplished from different perspectives, and in this way, it is likely able to direct users to seek the best solutions opportunistically.

Guideline (G7): Show sets of common operations.

There were many cases in which novice users were not aware of common patterns of operations. For example, advanced users were quick to come up with a sequence of actions to diagnose unwanted situations and then try another sequence of actions to solve the error(s). This observation, together with the analysis of general MMA tasks (Section 2.1) indicates that there are in fact common patterns or sequences of actions that are executed as a group, but not apparent or known to novice users. These users could, therefore, benefit from a tool that shows a sequence of necessary operations as a group that are executed to accomplish a certain goal.

Furthermore, during MMA tasks, users often perform a mental simulation of future actions [9, 98]. If a tool can show what target actions look like when actually executed, it could potentially eliminate this mental simulation but still provide a similar effect. Note that this guideline is different from the other guideline “Indicate possible next subtasks/actions” in that this guideline suggests an element that shows specific sequences of actions grouped together to accomplish common goals, as opposed to showing all the actions that are available given a goal and a context of the task. The two guidelines are, of course, interrelated in the sense that satisfying one of the guidelines will possibly affect the other, depending on how such support features are designed and implemented.

3.3.2 Design Guidelines, Novice-Specific Challenges, and Meta-Tasks

Table 3.3 summarises how each of the guidelines is aimed at addressing potential novice-specific challenges and meta-tasks.

Table 3.3: Design guidelines aimed at addressing novice-specific challenges and meta-tasks.

Design Guidelines	**Novice-specific challenges**								**Meta-tasks**								
	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	m1	m2	m3	m4	m5	m6	m7	m8	m9
G1. Provide additional contextual information where possible.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
G2. Show task-decompositions and structures.	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
G3. Provide explicit task-views that can dynamically change to show different levels of abstraction, scopes of tasks, and task contexts. notes, while novices struggle).	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
G4. Provide common task-structures that allow integration of newly performed actions.	✓	✓		✓	✓	✓			✓	✓	✓	✓	✓	✓	✓		✓
G5. Indicate possible next subtasks/actions.				✓	✓	✓			✓	✓	✓		✓		✓		✓
G6. Provide a means to carry out actions within contextually meaningful views.		✓		✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
G7. Show sets of common operations.		✓		✓	✓	✓			✓	✓	✓		✓	✓	✓		✓

In Section 3.2, I observed both low-level and high-level issues that novice users experienced and explained the challenges underlying these observed issues by considering associated meta-tasks at both levels. Issues and meta-tasks are intricate and interdependent both in their own right and taken together. It would therefore be highly impractical to address numerous issues separately, and more importantly, it could quite possibly be misguiding as it is the collective outcome that is of concern. Addressing an individual issue at one point could cause another issue to arise at another point.

Further, there is no indication of, and no reason to expect, that the potential underlying causes (i.e., the meta-tasks) are orthogonal factors in explaining the novice-specific challenges. Conceptually, the meta-tasks could be further organized into a partial "hierarchy" of abstraction, but it is not clear that this would be helpful in discerning better guidelines. Rather, the goal is to identify both specific and meta-tasks that can naturally be addressed with clear guidelines. In some cases, it is easier to address the specific task. In other cases, it can be more effective to address a meta-task that impacts several of the novice-specific challenges than addressing a specific task. *More effective* is meant to suggest that a single guideline can address multiple specific issues at once. Typically, when explaining a set of specific issues with a hierarchical underlying factor, I am looking for the most concise possible explanations. In this case, however, I am simply trying to understand a variety of issues in whatever way leads to reasonable and useful guidelines.

3.4 Conclusion

In order to address RQ1(a): *Are there common issues experienced by novices?*, I have conducted a series of preliminary user studies, which consisted of contextual observations of users, interviews, and surveys. These studies were conducted in order to observe how differently novice and advanced users would approach and perform general MMA tasks, and to understand what might be preventing novice users from performing MMA tasks in ways similar to advanced users. The results from these preliminary studies indeed revealed that there are common work-flow patterns and approaches that advanced users employ but novices do not. These patterns and approaches were independent from any specific MMA systems, and they often involved tasks that are closely related to problem-solving and task-planning in general. Novices understandably had issues with employing these patterns and approaches.

The definition of a task-view along with important properties was also presented. A task-view is a full or partial representation of the structure of a task. It consists of subtasks and specifies hierarchical and sequential relationships of these subtasks. Task-views will change dynamically along with the task-view properties (scope, abstraction levels, and constraints) as the user works through the stages of an MMA project.

Based on the theoretical background presented in Chapter 2 and the analysis of the observed novice issues, I hypothesised that there are some recurring themes to the observed novice issues. Based on this hypothesis, I then categorized these issues into classes of related behaviours, and associated each of these classes with what I call meta-tasks.

Meta-tasks are tasks that users perform to construct, modify, and utilize an appropriate task-view. Although the exact definition of meta-tasks can be elusive, certain characteristics and properties of meta-tasks, such as traversing abstraction levels and task scopes, and translating conceptual task goals into actual commands within an MMA system, were identified. By considering these meta-tasks, I identified common novice challenges underlying the observed issues, thus positively answering RQ1(b). I explained that each of these challenges could be a result of difficulty with one or more meta-tasks.

The findings in this chapter culminated in a set of design guidelines for developing MMA support tools for novice users to effectively utilize existing MMA systems (addressing RQ2(a)). These guidelines are developed to aim at addressing novice-challenges that stem from meta-task difficulty.

Note that as there are numerous possible meta-tasks and by using a different set of meta-tasks from those listed in this work, the observed novice-issues could be categorized differently. This may also result in a different set of novice challenges, and consequently, resulting in a different set of design guidelines than the ones developed in this chapter. The theoretical framework that I developed in this work is not meant to guarantee particular sets of novice challenges or guidelines, but it is an approach to identify underlying challenges in regards to the notion of meta-tasks so that one could develop viable guidelines for their target novice user groups.

In the next chapter (Chapter 4), I will describe the design process of a new MMA support system that was developed based on the guidelines presented in this chapter.

Chapter 4

Design Development of the New Multimedia Authoring Environment

In this chapter, I will first propose a framework for a multimedia authoring (MMA) environment (Section 4.1) that satisfies the design guidelines developed in Chapter 3. I will then describe support features of this new environment (Section 4.2), developed through an iterative design process. As part of the design process, I conducted User Study 2 (Section 4.3) to explore and compare different possible design configurations.

4.1 A Framework for a Multimedia Authoring Environment

While altering existing interface designs to address novice-specific challenges [4, 41, 45, 69] (Section 2.1) has been shown to have certain benefits by bridging the gap between the system model and users' mental models [95], I intend to provide support for users who struggle to develop an appropriate mental model: the guidelines in this research collectively attempt to provide support for generating and maintaining task-views. Thus, the current approach is not meant to be an alternative to the existing methods, but it is meant to add further support that can presumably strengthen these approaches to address issues experienced by novice users. Appropriately generated task-views can provide additional contextual information that may be beneficial for effective meta-task performance. Moving toward the goal of providing such support, I propose a framework for an MMA environment that consists of the standard MMA system in conjunction with an additional support tool, called the *Interactive Task-Viewer (ITV)*.

In this framework, the ITV has two distinct roles:

1. It provides visualization of task-views that are constructed based on a given MMA project.
2. It provides alternate ways to interact with the MMA system (i.e., users can also execute actions and receive feedback through the ITV).

In the first role, the ITV allows the user to view the relationship between low-level actions and their higher level—and usually more structured—context. This multi-resolution view allows the task-view to change dynamically by varying its properties; scope, abstraction levels, and constraints, as defined in Section 3.2.2. It also supports planning and execution of operations at multiple levels. In the second role, the ITV provides an additional input channel through which the user can control the actual authoring system. That is, the ITV provides a view of the high-level context; from this the user selects an action, which in turn gets sent as a command to the MMA system. Similarly, any actions performed in the MMA system are represented in a dynamically updated view in the ITV.

The ITV itself consists of a window with interactive components (on a second display such as a monitor or tablet) and a semi-transparent overlay component on the main MMA system as shown in Figure 4.1. In this environment, the window displays the interactive visualization of the task-view while the overlay component helps establish the connection between the ITV and the MMA system.

Figure 4.2 shows the interaction/feedback cycle between the user, the MMA system, and the ITV. The user interacts with and receives feedback from the MMA system conventionally (Figure 4.2-a and b), while he or she interacts with the ITV (Figure 4.2-c and d) to change its views and select actions to execute through the ITV. The ITV then sends these selected actions as commands to the MMA system (Figure 4.2-f). Actions can thus be executed both through the ITV (Figure 4.2-f) and through the MMA system (Figure 4.2-a), and in each case, the other component will

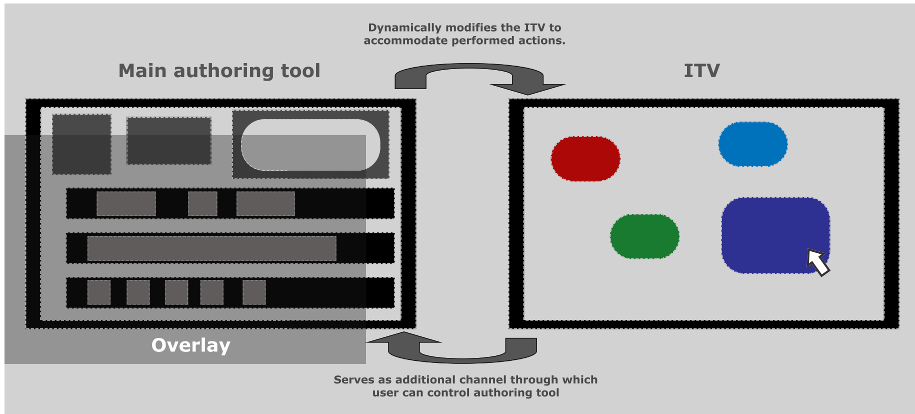


Figure 4.1: A schematic diagram of the ITV, which consists of a primary window with interactive components (shown on the right) and a secondary component which appears as a semi-transparent overlay superimposed on the main MMA system (shown on the left).

be updated accordingly (Figure 4.2-e and b).

4.2 Establishing Features

I will now describe a set of features of the ITV, designed based on the guidelines developed in Chapter 3. I initially started with a larger feature set and the list shown here is the resulting set after several iterations of evaluations (through cognitive-walkthroughs described in Section 4.2.2), pruning, and refinement of initial design ideas.

I present the following possible features, each of which would provide the support indicated by one or more of the guidelines, and thereby aiming at indirectly addressing one or more of the novice issues found in User Study 1:

- Contextual grouping of actions (contextual views)
- Multi-level/Hierarchical structure of task representation
- Dynamic views (multi-level abstraction and scope)
- Show possible subtask/actions within a context
- Operation previews
- Visualization of relationships between performed actions and affected data

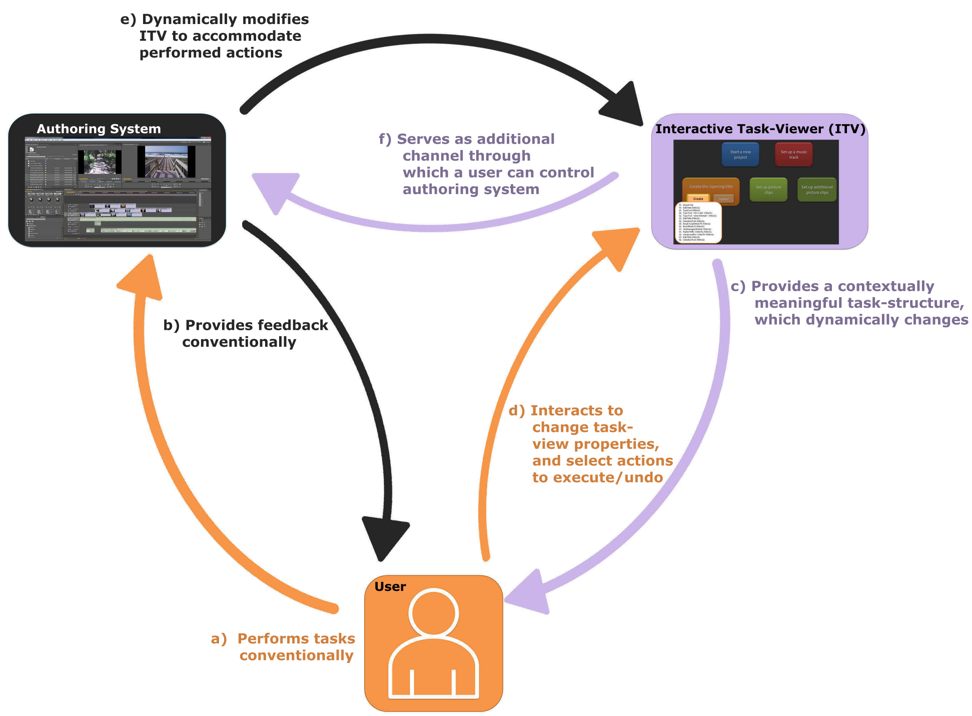


Figure 4.2: A framework for an MMA environment and its interaction/feedback cycle between users, the Interactive Task-Viewer (ITV), and the MMA system. The user executes actions (orange arrows) on both the MMA system and the ITV. The MMA system sends its output (black arrows) both to the user, in the form of visual feedback, and to the ITV, in the form of internal messages allowing the ITV to update its display. The ITV sends its output (magenta arrows) to both the user, in the form of visual feedback, and to the MMA system, in the form of internal messages that the MMA system can then execute as actions (e.g. open a file, trim the endpoint of a video, etc). Thus, each of the three elements—the user, the MMA system, the ITV—interacts in a bi-directional way with the other two.

- Annotation of task-view

I will first describe each of these possible features (Section 4.2.1) in conjunction with the guidelines that the features were designed to address, and then describe cognitive walkthroughs with early design implementations of these features (Section 4.2.2).

4.2.1 Features

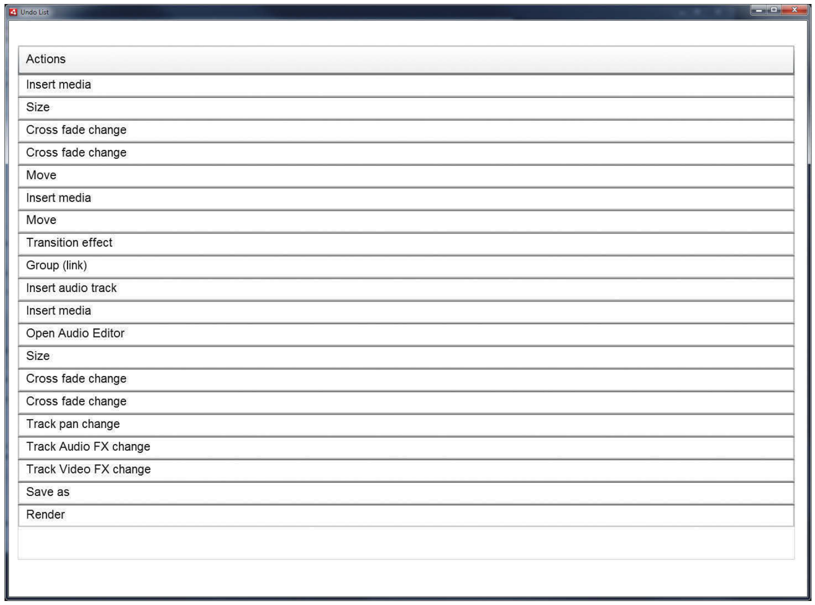
Contextual grouping of actions (contextual views)

Toward the goal of constructing a comprehensive task-view, I considered organizing actions into contextually meaningful groups. For example, traditional menu systems group actions based on their types: the *File* menu includes actions such as *Open*, *New*, and *Close*, all of which involve file operations, while the *Edit* menu includes *Copy*, *Paste*, and *Cut*, all of which modify or deal with data in the workspace. While this organization has been used by many applications for many years and existing users are familiar with it, it is limited when it comes to reviewing tasks. Contexts of the task such as what sort of project it is (e.g., wedding video editing), kinds of subtasks being performed (e.g., colour correction), and which data are involved (e.g., video clips for the ending segment), are not available in the action organization of the menu systems. The following example illustrates this limitation.

> **Scenario.** *We are working on an MMA project to create a music video. Figure 4.3 shows the list of actions that have been performed so far*[1]*, and we are trying to see what actions have been performed for the music track so far so that we can determine what needs to be done next.*

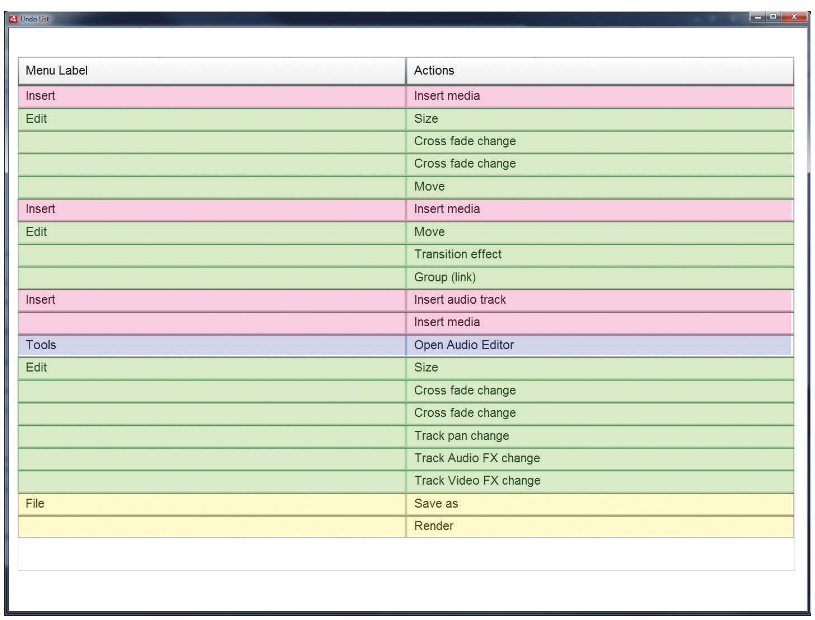
To achieve this task effectively, the user would need to understand more context about the actions in this list. For example, using the standard menu organization,

[1]These are taken from an actual undo list during a video production project performed by the author.



Actions
Insert media
Size
Cross fade change
Cross fade change
Move
Insert media
Move
Transition effect
Group (link)
Insert audio track
Insert media
Open Audio Editor
Size
Cross fade change
Cross fade change
Track pan change
Track Audio FX change
Track Video FX change
Save as
Render

Figure 4.3: A list of actions that have been performed for an MMA project. This list is the relevant excerpt from a longer list.



Menu Label	Actions
Insert	Insert media
Edit	Size
	Cross fade change
	Cross fade change
	Move
Insert	Insert media
Edit	Move
	Transition effect
	Group (link)
Insert	Insert audio track
	Insert media
Tools	Open Audio Editor
Edit	Size
	Cross fade change
	Cross fade change
	Track pan change
	Track Audio FX change
	Track Video FX change
File	Save as
	Render

Figure 4.4: Actions with Menu labels: actions are grouped based on the menu which they belong to. For example, "Size," "Cross fade change," and "Move" all belong to the "Edit" menu.

we would group actions under menu labels whenever one or more consecutive actions are from the same menu label, as shown in Figure 4.4. This grouping is, however, highly limited in the information it conveys about task contexts because actions are organized based on very generic action types.

In contrast, consider the list shown in Figure 4.5. Here, I have used descriptions of subtasks[2] to group the same list of actions. With this single change to the grouping scheme, the list of past actions becomes much more contextually meaningful. I call this presentation of grouped actions a *contextual view*.

Subtask Description	Actions
Setting up picture clip "beach.jpg"	Insert media
	Size
	Cross fade change
	Cross fade change
	Move
Setting up picture clip "ocean.jpg"	Insert media
	Move
	Transition effect
Link clips "beach.jpg" and "ocean.jpg"	Group (link)
Setting up music track ("How deep is the ocean.mp3")	Insert audio track
	Insert media
	Open Audio Editor
	Size
	Cross fade change
	Cross fade change
	Track pan change
	Track Audio FX change
Transition effects between "beach.jpg" and "ocean.jpg"	Track Video FX change
Saving the work space	Save as
Creating output file "ocean-video.mp4"	Render

Figure 4.5: Actions with subtask headings: I have used descriptions of subtasks to group the same list of actions. With this organization, the list of past actions becomes much more contextually meaningful.

Table 4.1 shows the corresponding guidelines that this feature is designed to address.

Multi-level/Hierarchical structure of task representation

Another approach to further organize these grouped actions is to create a hierarchical structure. For example, we can organize the action groups based on the type of

[2]These subtasks were manually generated for the purpose of this demonstration, based on common subtasks that are performed during general MMA tasks.

Table 4.1: The guidelines supported by the contextual views.

	Design Guidelines: Show/provide...						
	G1 ... additional contextual information	G2 ... task-decompositions & structures	G3 ... explicit task-views that can dynamically change	G4 ... integration of newly performed actions	G5 ... next subtasks/actions	G6 ... executable actions within contextual views	G7 ... common operations
This feature addresses:	✓	✓					

Table 4.2: A multi-branch representation of actions.

Image operations	Sound operations	Master operations
Setting up picture clip "beach.jpg" ⇓ Setting up picture clip "ocean.jpg" ⇓ Link clips beach.jpg and "ocean.jpg" ⇓ Transition effects between "beach.jpg" and "ocean.jpg" ⇓ Creating output file "ocean-video.mp4"	Setting up music track ("How deep is the ocean.mp3")	Saving the work space

data that are manipulated, e.g., we have subtasks that deal with video/image data (setting up, linking, and applying effect to pictures), sound data (setting up music track), and overall/master data (save and render). In this way, the task-view consists of three branches as seen in Table 4.2. Within each of these branches are subtasks which themselves consist of one or more low-level actions, thus creating a three-level hierarchical structure in this particular example.

Alternately, instead of the types of data, the branches could represent the types of data manipulations, such as (1) preparation, (2) modification, and (3) externalization, in which case, we will have the structure shown in Table 4.3.

Table 4.3: A multi-branch representation of actions based on the types of data manipulation.

Data preparation	Data modification	Data externalisation
Setting up picture clip “beach.jpg” ⇓ Setting up picture clip “ocean.jpg” ⇓ Link clips beach.jpg and “ocean.jpg”	Transition effects between “beach.jpg” and “ocean.jpg” ⇓ Setting up music track (“How deep is the ocean.mp3”)	Saving the work space ⇓ Creating output file “ocean-video.mp4”

Table 4.4 shows the guidelines that this feature is designed to address.

Dynamic views (multi-level abstraction and scope)

In order to effectively support users performing various meta-tasks, the task-view needs to adapt based on the information the user needs. For example, when a user employs the top-down approach, the ITV could initially provide a task-view with high-level subtask descriptions, and when the user proceeds down the hierarchical

Table 4.4: The guidelines supported by representation of a task hierarchical structure.

	Design Guidelines: Show/provide...						
	G1 ... additional contextual information	G2 ... task-decompositions & structures	G3 ... explicit task-views that can dynamically change	G4 ... integration of newly performed actions	G5 ... next subtasks/actions	G6 ... executable actions within contextual views	G7 ... common operations
This feature addresses:	✓	✓					

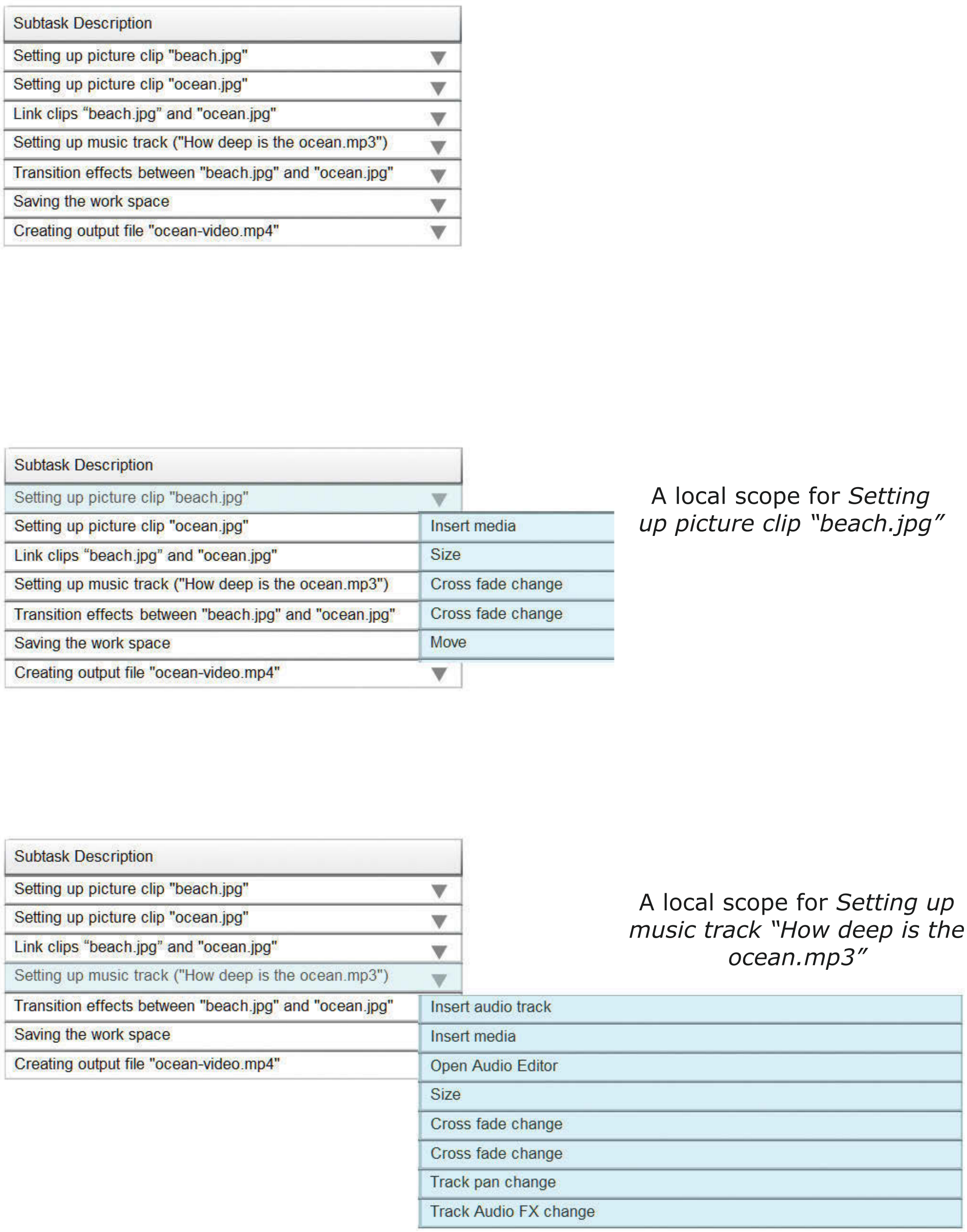


Figure 4.6: Subtask headings with partially revealed actions. The user interacts with the ITV to reveal more detailed information when needed.

levels to investigate the detailed information of the task, the ITV could reveal these lower-level actions. These interactions are illustrated in Figure 4.6.

Table 4.5 shows relationship of this feature to the corresponding guidelines.

Table 4.5: The guidelines supported by the dynamic views.

	Design Guidelines: Show/provide...						
	G1 ... additional contextual information	G2 ... task-decompositions & structures	G3 ... explicit task-views that can dynamically change	G4 ... integration of newly performed actions	G5 ... next subtasks/actions	G6 ... executable actions within contextual views	G7 ... common operations
This feature addresses:	✓	✓	✓				

Show possible subtask/actions within a context

In addition to reviewing past actions using the dynamic view of the task hierarchical structure, the user can also use the same approach for browsing future actions. That is, the user can consult a view that lists only those actions relevant to a given context. For example, when a user is performing a subtask labelled "Setting up picture clip beach.jpg," the ITV shows operations such as "modify the picture clip" or "apply visual effects" (as sketched in Figure 4.7).

While this approach can be more effective for narrowing the search space of target actions than the familiar menus and tool-bar icons, it should not imply the removal of latter items from existing MMA systems. Instead, it could augment the existing

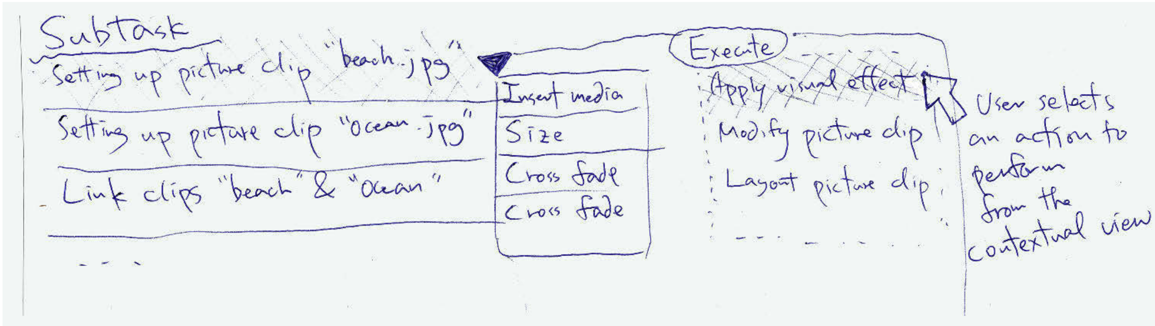


Figure 4.7: An initial design idea of a feature that allows the action execution from a contextual grouping of actions. In this sketch, as an extension to the view similar to Figure 4.6, the ITV also displays a list of possible actions (on the right) relevant to the given context, from which the user can select an action to be executed.

UI by adding optional methods for browsing possible actions.[3]

Table 4.6 shows the guidelines that this feature is designed to address.

Execution of actions directly from the task-view

As described in Section 4.1, the ITV provides an additional input channel to execute actions. This could be implemented as an extension of the previous feature: that is, from those possible actions that are relevant to a specific contextual group, the user can select an action that gets sent as a command to the MMA system.

Allowing action executions from contextually meaningful groups, rather than the standard menu, can help users narrow the search space of particular action commands when they do not know the terminology associated with action commands. For example, as seen in User Study 1 (Section 3.1), terminology is indeed a point of confusion and inefficiency for many novice users: the term *Overwrite* used in Adobe Premiere Pro refers to the action of bringing a new data clip into the timeline, while the same action is referred to as *Media drop in tracks* in Sony Vegas Pro. When a clip is moved to a new location, Premiere Pro refers to this action as *Lift & Overwrite Selection* while Vegas calls it *Move event*. In these cases, users could select

[3]Section 5.2.3.1 will describe how this feature is implemented.

Table 4.6: The guidelines supported by the feature to show possible subtask/actions within a context.

	Design Guidelines: Show/provide...						
	G1 ... additional contextual information	G2 ... task-decompositions & structures	G3 ... explicit task-views that can dynamically change	G4 ... integration of newly performed actions	G5 ... next subtasks/actions	G6 ... executable actions within contextual views	G7 ... common operations
This feature addresses:	✓				✓		

and execute actions from a smaller subset consisting of only those relevant to a local scope of the task-view (e.g., importing or layout media) regardless of which standard menu item encapsulates the target actions. The users can also initiate a cluster of essential subtasks without knowing which subtasks are necessary within this scope, plan a viable course of action, and execute each individual subtask.

Table 4.7 shows the corresponding guidelines that this feature is designed to address.

Expand task-views with new actions

When actions are executed (either directly by using the MMA system's UI or through the ITV), the task-view will need to incorporate these new actions into its representation. Each executed action is categorized into one or more contextual groups and added to the end of the list of previously performed actions in those groups. For example, suppose the user applied a video transition effect on a picture clip "beach.jpg."

Table 4.7: The guidelines supported by the execution of actions directly from the task-view.

	Design Guidelines: Show/provide...						
	G1 ... additional contextual information	G2 ... task-decompositions & structures	G3 ... explicit task-views that can dynamically change	G4 ... integration of newly performed actions	G5 ... next subtasks/actions	G6 ... executable actions within contextual views	G7 ... common operations
This feature addresses:	✓					✓	

From the previous example (Figure 4.5), this action should belong to the subtask "Transition effects between beach.jpg and ocean.jpg," so instead of appending to the end of the entire list of actions, it should be appended to the relevant group(s) of action lists. In this case, for example, after the third last action "Track Video FX change" in Figure 4.5 instead of after the last one "Render."

This approach in part addresses the non-linear nature of MMA tasks: we usually deal with multiple media simultaneously, perform different actions on different parts of these clips, and we may go back and forth between the current subtask and the previously paused ones, as is typical of the opportunistic strategy. Furthermore, in MMA, a slight change in one data clip has repercussions throughout the workspace, and the initially planned tasks need to be altered to accommodate necessary changes. Therefore, a linear representation of actions alone cannot appropriately express the structure of common MMA tasks.

The idea of using multiple paths to represent performed actions is not new. Undo branching, for example, has been proposed in several studies [38, 52, 67, 88] and it is implemented as part of some programs such as Vim, a popular text editor [77]. Performed actions are sequenced as in the regular undo list until the user undoes some of the actions to a point in the past, where a new branch starts. Undone action branches remain visible and in most implementations, the user can revisit any of those branches to redo the actions to recover the previous state(s). A typical style of this type of multi-branching of actions is depicted in Figure 4.8.

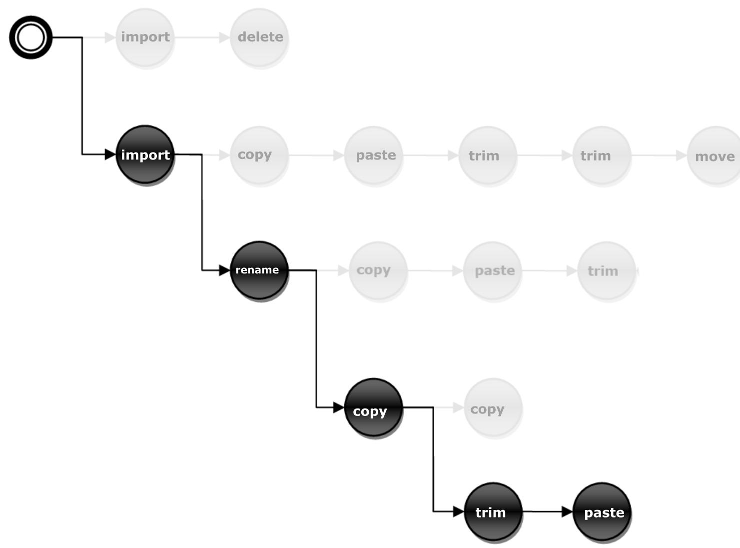


Figure 4.8: Multi-branch representation typically used for undo-list of previously performed actions. The greyed-out circles indicate the actions that are undone, and the solid circles indicate the current path. This representation does not reflect non-linearity and concurrency of MMA. In this example, the user first performed “import” and “delete” actions, but undid both actions. The user then performed the import action again, and then proceeded to perform a sequence of actions: “copy,” “paste,” “trim,” and so on. He or she then decided to undo all but the first action.

Although this approach supports multiple paths of actions, the particular sequence of actions to reach the current state from the beginning still remains linear (as indicated by the solid circles/path in Figure 4.8). It still does not, therefore, appropriately reflect the non-linear nature of MMA tasks. Instead, the ITV represents multiple paths as previously depicted in Figures 4.2 and 4.3. For example, in

Figure 4.2, when a new action to apply a video transition is performed, it is added to the corresponding subtask group ("Setting up picture clip beach.jpg"), which is in the branch of "Video operations" as this action modifies the video data.

When the task progresses and the hierarchical and sequential relationships of the task structure become complex, the corresponding task-view also becomes intricate. This growth of the task-view makes it difficult to maintain it mentally, especially when the task is interrupted by a long period of inactivity. Indeed, task interruptions are known to cause problems [57,108,111] and they frequently happen in MMA tasks as completing authoring projects often takes anywhere from hours to months. In these cases, having external representations of task-views associated with the project and stored for later use (as opposed to mentally maintaining them) could be helpful.

Table 4.8 shows the corresponding guidelines that this feature to allow expansion of task-views with new actions is designed to address.

Table 4.8: The guidelines supported by task-views that can expand with new actions.

	Design Guidelines: Show/provide...						
	G1 ... additional contextual information	G2 ... task-decompositions & structures	G3 ... explicit task-views that can dynamically change	G4 ... integration of newly performed actions	G5 ... next subtasks/actions	G6 ... executable actions within contextual views	G7 ... common operations
This feature addresses:	✓			✓			

Operation previews

When the task-view becomes large, the dependencies between the subtasks/actions must be carefully handled. As discussed in Section 3.2.2, certain actions have prerequisite actions, but these constraints may not be fully visible to inexperienced users. For example, when a user wants to record an audio clip, the user must know all the things he or she needs to do first in order for that step to be possible (e.g., check microphone levels, search for a suitable location on timeline, and arm the target track). In other words, just as users might need to select a task-view based on abstraction level or scope, they might sometimes need to select the scope and abstraction level so as to view all the constraints associated with a particular subtask. While constraints and dependencies between subtasks/actions can make task representation more complex, the current and the next few features are intended to mitigate this complexity.

Although certain kinds of previews are common in MMA systems (e.g., a coarse render of how a cross-fade effect will look), the ITV framework opens the possibility for other types of previews.

An *operation preview* is essentially a demonstration of an operation which consists of one or more actions. That is, when the user wants to execute a certain operation (e.g., "record an audio clip") but faces potential problems (e.g., the user does not know how to perform it, or cannot perform due to unknown action dependencies), the ITV will display a short demonstration of a sequence of actions. Thus, this feature will address two of the common novice-specific challenges: recognizing dependencies (Challenge-5), and recognizing combinations of actions needed to accomplish a certain goal (Challenge-6).

Table 4.9 shows the corresponding guidelines that this feature is designed to address.

Table 4.9: The guidelines supported by operation previews.

	Design Guidelines: Show/provide...						
	G1 ... additional contextual information	G2 ... task-decompositions & structures	G3 ... explicit task-views that can dynamically change	G4 ... integration of newly performed actions	G5 ... next subtasks/actions	G6 ... executable actions within contextual views	G7 ... common operations
This feature addresses:	✓				✓		✓

Visualization of relationships between performed actions and affected data

Another feature to help mitigate the complexity of the task structure is to visualize relationships between performed tasks and affected data in the workspace. In User Study 1, this information was found difficult to obtain: a user needed to inspect individual video clips to investigate exactly which colour correction was applied to which clip. Standard UIs provide a few ways to decipher this information: (1) undoing actions one by one to see which data have been affected by each undo, (2) visually or sonically inspecting properties of data (e.g., colour and timbre) by playing back media, or (3) opening a special property window for a target data clip. All these approaches can be tedious, and certain actions such as trimming or splitting are still not always discernible with the last two approaches. Thus, providing a mechanism to show which data are affected by which actions can be helpful.

One way to accomplish this is to highlight or show connections between previously performed actions in the ITV and data clips in the workspace, as sketched in Figure 4.9.

Table 4.10 shows the corresponding guidelines that this feature is designed to address.

Annotation of task-views

Annotation of task-views is another possible mechanism that could add useful contextual information. In addition to features often provided in typical MMA systems such as changing names and background colours of clips, the ITV could allow users to attach annotations to the performed tasks. As there is no limitation on what types of information can be used, this feature can potentially address multiple guidelines. For example, the user could annotate about the next viable action plan (e.g., "will need to normalize the volume before rendering"), which may imply task-decompositions and

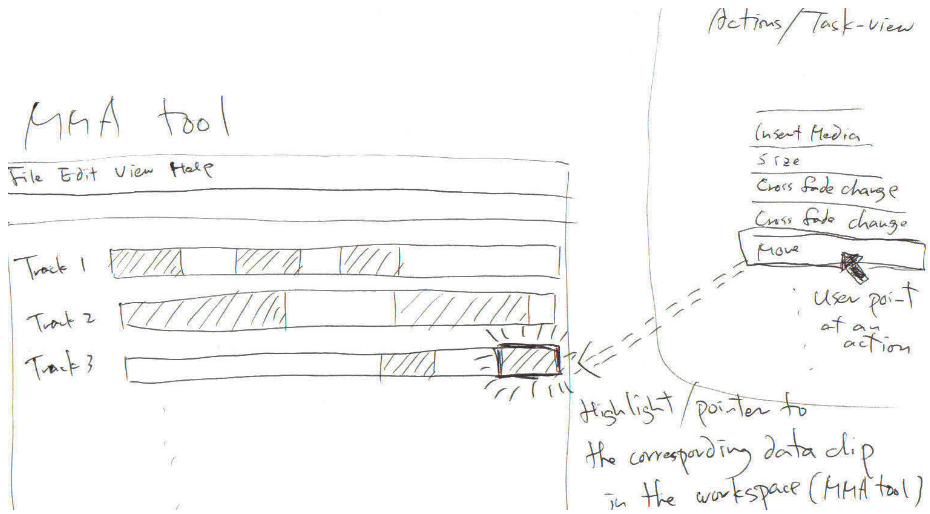


Figure 4.9: An initial design idea for indicating the relationship between performed action and affected data in the workspace. It shows a task-view on the right similar to the one presented in Figure 4.5, and the MMA system on the left. When the user points at one of the actions (“Move”), the ITV will highlight/point to the corresponding data clip in the workspace of the MMA interface.

Table 4.10: The guidelines supported by visualization of relationships between performed actions and affected data.

	Design Guidelines: Show/provide...						
	G1 ... additional contextual information	G2 ... task-decompositions & structures	G3 ... explicit task-views that can dynamically change	G4 ... integration of newly performed actions	G5 ... next subtasks/actions	G6 ... executable actions within contextual views	G7 ... common operations
This feature addresses:	✓						

at the same time indicate next suitable actions. Figure 4.10 shows an initial sketch depicting this feature.

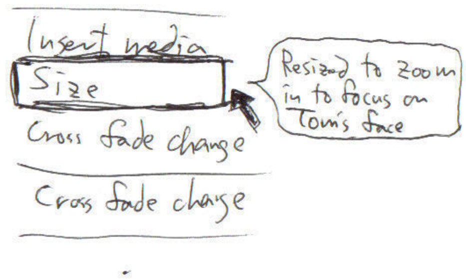


Figure 4.10: An initial design idea for annotation of the task-view. The sketch shows the annotation of an action ("Size"), which is revealed (e.g., a pop-up window) when the user points to an action in the ITV.

Table 4.11 shows the corresponding guidelines that this feature is designed to address.

Table 4.11: The guidelines supported by annotation of task-views.

	Design Guidelines: Show/provide...						
	G1 ... additional contextual information	G2 ... task-decompositions & structures	G3 ... explicit task-views that can dynamically change	G4 ... integration of newly performed actions	G5 ... next subtasks/actions	G6 ... executable actions within contextual views	G7 ... common operations
This feature addresses:	✓	(✓)			(✓)		(✓)

As is often the case, integrating these features into a single interface was a challenging process, since addressing one particular issue could introduce other issues. For example, providing too much contextual information could clutter the interface itself, and each feature could, both positively and negatively, affect the way other features address corresponding issues. Thus, I sought to find a feature set that most optimally address identified issues.

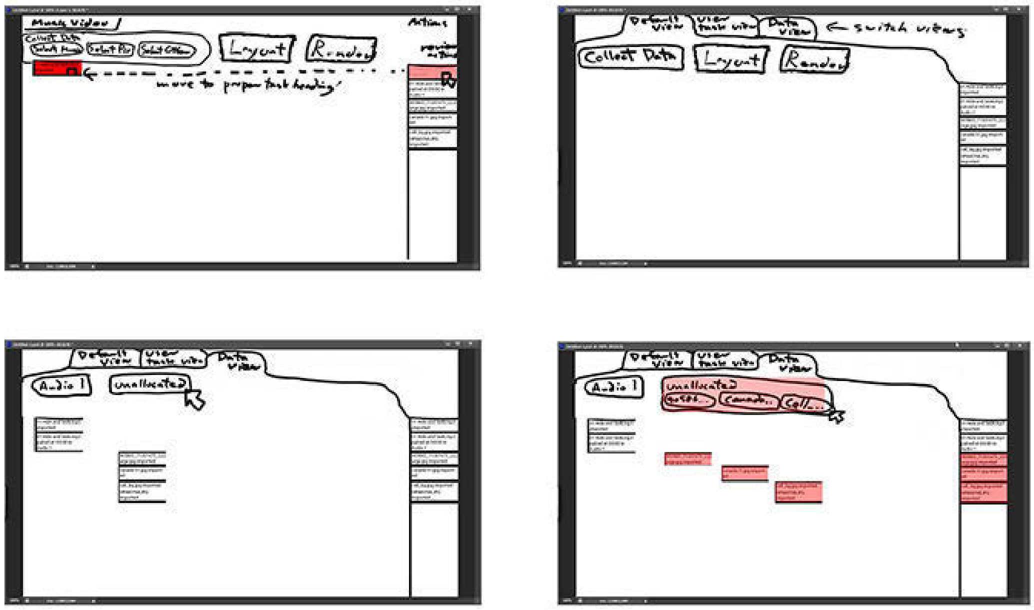
The design process was iterative, but can be coarsely grouped into three stages: cognitive walkthroughs (Section 4.2.2), exploration of design features (User Study 2, Section 4.3), and finally User Study 3 (Section 5.2). Different features were explored and tested at different points in the design cycle. Implementing and testing all possible features at every stage was simply not feasible. Therefore, subsets of the feature set had to be selected for testing at different stages. These subsets were chosen based on various criteria:

- feasibility and meaningfulness of testing a given feature with a given prototype "depth" (e.g., Can testing different context clues be done with paper mock-ups?)
- the amount by which testing a particular feature could help narrow the subsequent design space (e.g. by testing an organizational structure early on, does this significantly help focus the next iteration of the design?)
- the priority of the issues that they addressed (e.g. how important is it to test a feature multiple times that addresses only one of the least significant user issues?)

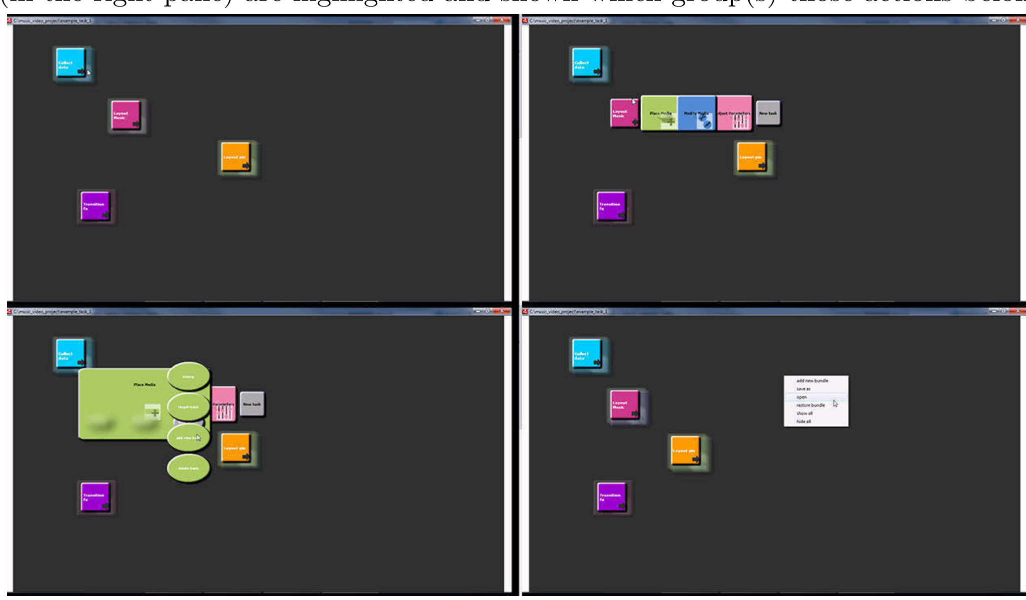
Thus, some features were tested using a cognitive walkthrough with a low-fidelity prototype, some as part of User Study 2 to narrow the design space, and some as part of User Study 3 (and some were tested in two or more settings).

4.2.2 Cognitive Walkthroughs with Low-Fidelity Prototyping

Prior to a larger-scale evaluation with users, I evaluated some of these prospective features by several rounds of informal cognitive walkthroughs using paper-based mock-ups and low-fidelity implementations. The users were given steps to create a storytelling video from a set of raw media as outlined in Table 4.12. The evaluations



(a) A paper mock-up for the early design concept. In the left pane are objects representing action groups. When the user interacts with them, the corresponding actions in the linear list (in the right pane) are highlighted and shown which group(s) those actions belong to.



(b) A low-fidelity prototype for the early design concept. The task-view initially displays a number of objects representing action groups (top-left figure). When the user interacts with them, they reveal their child objects displaying the hierarchical structure of the task.

Figure 4.11: Samples of paper mock-ups and low-fidelity implementations. These early design ideas were evaluated and refined through several rounds of cognitive walk-throughs.

were based on several versions of the mock-up, a few samples of which are shown in Figure 4.11.

Table 4.12: List of the provided media and the task procedures.

Provided media set	Basic steps
• Narration clips • Video clip with no sound (some needed to be trimmed, sliced, or grouped) • 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 the prototypes were primitive, the results of these evaluations were consistent with my prior observations of users as discussed in Chapter 3. These cognitive walkthroughs also allowed me to gain insights into how this tool could support characteristics of problem-solving, which are closely related to meta-tasks.

As expected, the information provided by the contextual views (e.g., a limited set of actions based on the current context) helped to narrow the search space of actions. One of the issues observed in User Study 1 was that novices are unable to escape from unwanted situations. Reducing the number of possible actions by classifying them into contextually more meaningful groups can decrease the chance of attempting inappropriate actions in the first place, and potentially increase the chance of choosing appropriate recovery actions when needed.

Dynamic views were helpful when users needed to change perspectives of the task-view: for example, task overview corresponds to an abstract perspective typically employed at the initial stage of a top-down strategy, while the grouped actions can provide a local perspective often utilized for a bottom-up strategy. By allowing different abstraction levels and scopes, the interface could accommodate the requirements of an opportunistic strategy. As well, indicating relationships between the task and the data in the workspace helped users by clarifying which actions were performed on which data clips.

The cognitive-walkthrough also highlighted issues to address before further testing. For example, early versions of the mock-up used only a few colours (black, white, and red), and this highlighted the importance, in future, of consistent colouring schemes for clarity of the visualization. Poor use of text (e.g., long and similar headings) also led to confusion.

In Section 4.1, I proposed a framework for an MMA environment that is based on the developed guidelines from Chapter 3. Initial design ideas were discussed and evaluated in a set of informal cognitive-walkthrough sessions with low-fidelity mock-ups (Section 4.2). The results from each evaluation session led to the refinement for the next stage of a design development cycle. Table 4.13 summarises the list of features that I described above, the guidelines they address, and the point in the design cycle at which they were tested/explored. I will next present a set of user studies conducted to further explore and refine these design features.

Table 4.13: This table summarises the list of features that I described, the guidelines they address, and the point in the design cycle at which they were tested/explored.

Features	Tested in			Design Guidelines: Show/provide...						
	Cognitive walkthroughs	User Study 2	User Study 3	G1 ... additional contextual information	G2 ... task-decompositions & structures	G3 ... explicit task-views that can dynamically change	G4 ... integration of newly performed actions	G5 ... next subtasks/actions	G6 ... executable actions within contextual views	G7 ... common operations
Contextual views	✓	✓	✓	✓	✓					
Hierarchical structure	✓	✓		✓	✓					
Dynamic views	✓		✓	✓	✓	✓				
Show possible actions	✓		✓	✓				✓		
Action execution from the ITV	✓		✓	✓					✓	
Expansion of task-view	✓		✓	✓			✓			
Operation previews		✓	✓	✓				✓		✓
Visualization of task/data	✓		✓	✓						
Annotation of task-view		✓		✓	(✓)		(✓)			(✓)

4.3 Exploration of Design Features (User Study 2)

4.3.1 Purpose of The Study

In this user study (User Study 2), I evaluated prospective design ideas for the ITV in order to determine their usability. The evaluations presented in this section also helped to determine whether or not a particular design option would have certain advantage over others, thus allowing me to choose an appropriate option for each prospective feature (addressing RQ2(b): *Which solutions are worthwhile implementing initially?*).

Answering this question was a necessary step in this research, as implementing and testing numerous parameters (e.g., hierarchical levels, contextual grouping schemes, types of projects to be tested, etc) of all possible features separately was simply not feasible. Such an undertaking would quickly grow out of hand requiring a very large number of participants, each of whom would need to perform experimental tasks for several hours at a time. In order to address this issue, I have conducted several rounds of evaluation studies (cognitive walkthrough sessions and User Study 2) to refine the design space and prioritise certain design decisions.

In this phase of the design cycle, I primarily focused on design options and styles for task-view representations, and the evaluation tasks for mainly reviewing previously performed actions. Thus, the aspects evaluated in User Study 2 involve visualizing task-structures (i.e., the first role of the ITV as described in Section 4.1). Based on the representation styles chosen as a result of User Study 2, I focused on interactive aspects of the ITV (i.e., its second role) in User Study 3 with a more functional prototype (Section 5.2).

4.3.2 Study Settings

4.3.2.1 Study Design

In User Study 2, I conducted evaluations to compare design options for the following aspects/features of the ITV:

- Eval.1) Different types of contexts for grouping actions (i.e., Contextual Views) (Section 4.3.3.1)
- Eval.2) Different styles of hierarchical representations and abstraction levels (Section 4.3.3.2)
- Eval.3) Different annotation retrieval methods (Section 4.3.3.3)
- Eval.4) The visualization of prerequisite actions (Section 4.3.3.4)

I will describe details of evaluation procedures and their results in respective sections, but the general study design can be summarised as follows: for each evaluation task, the participants rated the usability of different design options/ideas. All the activities on screen and conversation with experimenters were recorded. I used a repeated-measures design to collect the data, and a within-subjects design to perform the comparisons. I analysed user behaviours and task performance and used statistical data as general guidance in accordance with the qualitative observations to support the discussions and conclusions. For all the statistical analyses used in User Study 2, the significance was tested at a level of $\alpha = .05$.

4.3.2.2 Participants

Eleven participants (seven males and four females, 18 to 39 years of age) took part in User Study 2. The participants are from a variety of technological backgrounds (five in engineering and science, and six in non-technical disciplines), but they all had no experience in multimedia content development, except one, who did have minimal experience with an entry-level video editor (i.e., less than ten times for simple video editing).

4.3.2.3 Study Equipment

The participants performed all the evaluation tasks on a personal computer. The ITV features were implemented using the Apache Flex framework[4] that allowed creation of a light-weight tool appropriate for the purposes of this phase of the research.

4.3.3 Evaluations Tasks

4.3.3.1 Evaluation 1: Comparison of Contextual Views

While providing contextual information for a task can help users in a number of ways (Section 4.2), there are different types of contextual clues that could be provided. In this evaluation, I compared grouping schemes of actions to see if any indication that novice users prefer (or dislike) one of these schemes relative to the others.

Grouping schemes for contextual views

I considered the following four grouping schemes (Views A to D):

View A (data/media types) grouped actions based on the data/media types: video, audio, still images, and text (title). In addition, *master* operations referred to those actions affecting the all media rather than individual data clips (e.g., master volume controls). In this view, all actions performed on a specific data clip are first collected to form action groups, and the data clips of the same media type are then grouped to form media groups. That is, an action group is a collection of all actions performed on a specific clip, and a media group is a collection of action groups for all

[4]Initially developed by Macromedia and then acquired by Adobe, before the project was handed to Apache Software Foundation [123].

media of a specific type. This view is shown in Figure 4.12.

Figure 4.12: View A: a contextual view based on the data/media types. In this example, actions performed for a still image clip called "usa03.jpg" are first grouped into one group. All other groups consisting of actions performed for still images are then grouped into one big group called "Still image data operations," creating a 3-level hierarchical structure.

View B (subtask headings) grouped actions based on subtask headings in a similar manner as discussed in Section 4.2.1. This grouping is depicted in Figure 4.13. Each subtask group consisted of further smaller tasks (sub-subtasks) to construct a 3-level hierarchical structure of the task-view.

View C (Time-line) grouped actions according to the final product's time-line. It first groups actions performed for particular sections (e.g., opening scene, restaurant scene, ending, etc). Each section consisted of sub-sections, creating a 3-level hierarchical structure as in all the other views. This view is captured in Figure 4.14. Note that this grouping scheme assumes that such sectioning is already known as is often the case of many MMA projects (i.e., at least rough story-lines are typically determined before main editing processes begin).

Figure 4.13: View B: a contextual view based on the subtask headings. In this example, a subtask group called “Create the opening title” would include two sub-subtasks “Create” and “Layout,” each of which groups relevant actions.

Figure 4.14: View C: a contextual view based on the final product’s time-line. In this example (a music video production project), a typical music structure segmentation was used, such as an introduction, verses, and an ending.

View D (Menu structure) used the grouping scheme based on the original menu structures of the MMA system, similar to the one described in Section 4.2.1. Each menu is further divided into sub-menus creating a 3-level hierarchical structure as in all the other views (Figure 4.15). The structure of the menu used for this task-view was based on that of Adobe Premiere Pro.

Figure 4.15: View D: a contextual view based on the menu structures. In this view, actions are grouped based on the menu grouping used in a standard MMA system (e.g., File, Clip, etc). Each menu group contains sub-menus creating a 3-level hierarchical structure.

In addition to these contextual views, I also provided a linear list of actions, which is typically provided as an undo-list in standard MMA systems (Figure 4.16). This linear list was the control case in this evaluation, comparing to the treatment cases of the contextual views. Note that MMA systems often limit the number of actions shown in this type of list, but the current implementation of this linear list allowed all actions to be visible, making it comparative to the contextual views.

1. New/Open
2. Create sequence 01
3. Import Clip
4. Overwrite
5. Lift & Overwrite Selection
6. Import Clip
7. Import Clips
8. Overwrite
9. Lift & Overwrite Selection
10. Trim
11. Lift & Overwrite Selection
12. Trim
13. Apply Transition
14. Trim
15. Apply Transition
16. Lift & Overwrite Selection
17. Trim
18. Apply Transition
19. Lift & Overwrite Selection
20. Trim
21. Lift & Overwrite Selection
22. Trim
23. Lift & Overwrite Selection
24. Trim
25. Lift & Overwrite Selection
26. Trim
27. Lift & Overwrite Selection
28. Trim
29. Lift & Overwrite Selection
30. Trim
31. Lift & Overwrite Selection
32. Trim
33. Trim
34. Lift & Overwrite Selection
35. Trim
36. Lift & Overwrite Selection
37. Lift & Overwrite Selection
38. Trim
39. Lift & Overwrite Selection
40. Trim
41. Trim
42. Apply Transition
43. Apply Transition
44. Apply Transition
45. Apply Transition
46. Apply Transition
47. Apply Transition
48. Apply Transition
49. Import Clips
50. Overwrite
51. Lift & Overwrite Selection
52. Lift & Overwrite Selection
53. Trim
54. Lift & Overwrite Selection
55. Trim
56. Lift & Overwrite Selection
57. Trim
58. Lift & Overwrite Selection
59. Trim
60. Lift & Overwrite Selection
61. Trim
62. Trim
63. Delete
64. Apply Transition
65. Apply Transition
66. Apply Transition
67. Apply Transition
68. Apply Transition
69. Slide Transition
70. Move Parameter Segment
71. Apply Transition
72. Trim
73. Trim Transition
74. Trim Transition
75. Trim Transition
76. Trim Transition
77. Trim Transition
78. Import Clip
79. Edit Title (Title 01)
80. TypeTool (Title 01)
81. Type Tool: "All I want" (Title 01)
82. Type Tool: "(Joni Mitchell)" (Title 01)
83. Edit Title (Title 01)
84. Selection Tool (Title 01)
85. Brush Script Black 75 (Title 01)
86. Birch Black 80 (Title 01)
87. Charlemagne Bold 49 (Title 01)
88. Poplar Puffy White 51 (Title 01)
89. Garamond Pro White 50 (Title 01)
90. Edit Title (Title 01)
91. Selection Tool (Title 01)
92. Overwrite
93. Trim
94. Apply Transition
95. Apply Transition
96. Trim Transition
97. Trim

Figure 4.16: A simple linear list of actions, typically employed in standard MMA systems as an *undo-list*.

Procedure

The participants were given a set of contextual views representing common MMA projects. These projects represented "works in progress" and I prepared the task-views that grouped performed actions based on the grouping schemes as described above.

The participants repeated each of common operations listed in Table 4.14 using the contextual views (Views A to D). The objective of these tasks was to locate relevant action(s) within these task-views. The participants selected a view that was most helpful for performing each operation.

Table 4.14: A list of tasks for comparison of the contextual views.

- Find which data clip(s) were imported to the workspace but not used in the project.
- Replace the transition effects applied to the pictures for the 2nd verse (Water tunnel scene–P1210426.JPG, Water tunnel #2–P1210381.JPG, and Cherry trees–sakura 002.jpg) with additive dissolve transition.
- Revert a clip called "sakura 002.jpg" to its original state as when it was brought to the workspace for the first time (i.e., undo all the actions performed to the clip).
- Undo all the transition effects only applied to the picture clips.
- Replace a music clip with a new one but maintain all the actions performed on it.
- The additional pictures that you've imported did not turn out how you wanted. Undo all the actions performed on them.
- Replace a clip(s) used for the ending scene with a new one(s) and perform the same operations (such as modifications and moving) performed to the original clip.
- Undelete the clips that were deleted from the workspace.
- The 3rd verse is one picture-clip shorter than the other verses, add another picture and apply exactly the same modifications as the ones applied to the 2nd picture (Mt. Fuji) in the 3rd verse.
- The current font setting of the title (clip) is not ideal. Modify it so that it will have the previous font that was tested but abandoned.
- The transition effect of clip (P11000240.JPG) was slightly moved but it turned out that it did not need to be moved. Undo the action that slid the transition effect.
- Undo all the clip modifications (such as trimming and deleting) done to all the clips, but keep the transition effects as they are.

After the participants completed all the operations with the contextual views, they were also asked to perform the same set of operations using the linear list. The participants then rated the difficulty (on a five-point Likert scale, much easier to much harder) of using the linear list compared to the contextual views (Views A to D) for performing these tasks.

Results and Discussions

Table 4.15 shows the results of the comparison among the contextual views. Although there was statistically no significant difference in their preference among these various views (Friedman test: $p = .07$), participants often seemed to have problems finding operations with the menu-based grouping (View D, which was rated as a preferred view 17.71% of the times). This could possibly be due to the fact that it requires specific knowledge of which menu item encapsulates which actions, but none of the participants had the experience with the tool (Premiere Pro) from which this menu structure was derived. As noted earlier in Section 4.1, the menu based grouping does not provide much contextual information about the task compared to the other grouping contexts. This implies that the standard organization to present actions in the menu structure can be complemented by providing alternate ways to organize them and that it might potentially reduce a gulf of executions to improve task performance. This hypothesis was tested in User Study 3, presented in Section 5.2.

Table 4.15: Preferences of different views. The numbers in parentheses indicate the numbers of times each view is selected out of the total of 132 answers (12 operations rated by 11 participants).

View A (media type)	View B (subtask)	View C (time-line)	View D (menu)	(Does not matter)
33.33% (44/132)	22.73% (30/132)	25.00% (33/132)	17.42% (23/132)	1.51% (2/132)

When comparing the linear list of actions with the contextual views, the majority of participants expressed that using the linear list was harder to perform operations

(56% of answers indicated “much harder“ while 25% of answers were “harder”, as shown in Table 4.16). The participants stated that the action group views were informative about the kinds of tasks performed, that after using these contextual views to perform tasks, they were able to perceive the original list of actions as a sequence of more contextually meaningful tasks, rather than as a (seemingly) random sequence of actions. This finding implies a learning effect, of sorts, by the use of the contextual views.

Table 4.16: Difficulty rating of performing operations using only the linear list of actions, compared to the contextual views. The numbers in parentheses indicate the number of answers out of the total of 132 answers (12 operations rated by 11 participants).

much easier	easier	about the same	harder	much harder
0.76% (1/132)	4.55% (6/132)	13.64% (18/132)	25.00% (33/132)	56.06% (74/132)

Overall, each view had its own benefits that may be more suitable for some tasks than others. For example, when trying to find clips that were used for a specific section of the final product (e.g., “Replace a clip(s) used for the ending scene with a new one(s) [. . .]”), ten out of eleven participants understandably chose View C (based on the final product’s time-line) as a preferred view, while in another similar operation (e.g., “Revert a clip called “sakura 002.jpg” to its original state [. . .]”) eight out of eleven chose View A (based on the media type). This observation, together with the fact that the participants rated contextual views easier to use than the linear list, implies that a support tool should allow different contextual views to accommodate differences in styles of authoring and how users perceive task contexts.

4.3.3.2 Evaluation 2: Comparison of Task Hierarchical Representations and Abstraction Levels

While the participants expressed their preference toward the contextual views over a linear sequence of actions in the previous evaluation, there are different ways to present hierarchical structures of a task. This evaluation sought to identify differences in user preference, if any, among representation styles for hierarchical structures of a task-view and in abstraction levels provided by those views.

Styles of Hierarchical Representations

In this evaluation, I compared the following three styles of hierarchical representations of a task.

I) A text-based list of action/command names within each subtask headings, similar to standard undo-lists with a major difference being each list shows actions only relevant to each specific context, as shown in Figure 4.17. In this representation, each column represents a subtask, each of which itself contains one or more sub-subtasks consisting of associated actions.

II) Graphical objects with subtasks being included within a parent task (i.e., space filling hierarchical representation [60, 127]), in the same styles as View B in the previous evaluation (Figure 4.13).

III) The same style as II), but instead of the space-filling style, it used the tree-style (node-link) hierarchical representation shown in Figure 4.18, slightly similar to the style of ConcurTaskTree notation [19, 100–103] but without extensive use of its notations.

Procedure

Similarly to the comparison of contextual views, I asked the participants to compare the hierarchical representation styles (i.e., representations I to III). As a control case, I also provided the linear list of actions (as was previously shown in Figure 4.16). Thus, there were four different representation styles to be compared: a typical linear list of

[Starting a new project]	[Preparing clips]	[Cutting clips]	[Adding an SE]
<overall project setting> 1. New/Open <adding a sequence> 2. Create Sequence 01	<Import, trim and layout> 3. Import Clips 4. Mark Out 5. Mark In 6. Overwrite 7. Mark In 8. Mark Out 9. Overwrite 10. Mark Out 11. Overwrite <Synchronizing video and audio tracks> 12. Lift & Overwrite Selection 13. Lift & Overwrite Selection 14. Lift & Overwrite Selection 15. Lift & Overwrite Selection 51. Lift & Overwrite Selection <Adjusting audio tracks> 16. Disable Track Output 17. Disable Track Output 18. Move Parameter Segment 19. Apply Transition <Adjusting video tracks> 20. Lift & Overwrite Selection 21. Lift & Overwrite Selection	<Finding edit points> 22. Add Marker 23. Add Marker 24. Add Marker 29. Add Marker 32. Add Marker 33. Add Marker 41. Add Marker 42. Add Marker 46. Add Marker 47. Add Marker 48. Add Marker 56. Add Marker 57. Add Marker 61. Add Marker 62. Add Marker 66. Add Marker 67. Add Marker 71. Add Marker <slice> 25. Razor 26. Razor 27. Razor 30. Razor 34. Razor 35. Razor 43. Razor 44. Razor 49. Razor 50. Razor 58. Razor 59. Razor 63. Razor 64. Razor 68. Razor 69. Razor 72. Razor <Move cuts to track> 28. Lift & Overwrite Selection 31. Lift & Overwrite Selection 36. Lift & Overwrite Selection 37. Lift & Overwrite Selection 38. Lift & Overwrite Selection 39. Lift & Overwrite Selection 40. Lift & Overwrite Selection 45. Lift & Overwrite Selection 52. Lift & Overwrite Selection 53. Lift & Overwrite Selection 54. Lift & Overwrite Selection 55. Lift & Overwrite Selection 60. Lift & Overwrite Selection 65. Lift & Overwrite Selection 70. Lift & Overwrite Selection 73. Lift & Overwrite Selection <Trim and fades> 74. Apply Transition 75. Trim 76. Trim 77. Disable Track Output 78. Enable Track Output 79. Trim 80. Trim 81. Trim	<Adding audience laughter> 82. Insert Clip 83. Overwrite 84. Lift & Overwrite Selection 85. Lift & Overwrite Selection 86. Apply Transition 87. Trim Transition

Figure 4.17: View: Grouped actions in text format. Each column represents a subtask, each of which itself contains one or more sub-subtasks consisting of associated actions.

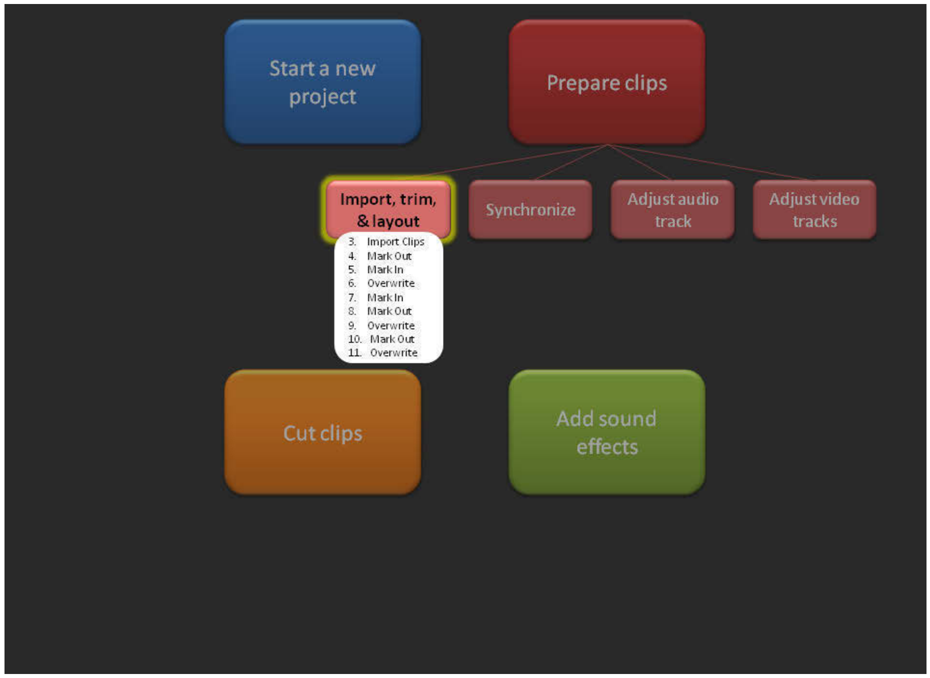


Figure 4.18: View: Grouped actions based on the subtask headings using the tree (node-link) style instead of the space-filling approach of View B (Figure 4.13) of the previous evaluation

actions (in text but no action groups), a text-based representation of grouped actions (Figure 4.17), Contextual View B (Figure 4.13), and a tree (node-link) representation (Figure 4.18). The first two views had all the actions visible at all times while the last two styles hid lower level actions until participants explicitly clicked through hierarchical levels to reveal them.

The participants evaluated usefulness of each of these styles in the situations shown in Table 4.17. The participants rated each view for its usefulness on a five-point Likert scale (very useless to very useful) in each of these situations.

Results and Discussions

The result of this comparison (shown in Table 4.18) revealed that (1) the ungrouped action list view was rated less useful than all the grouped views (all the pair-wise comparisons using Wilcoxon Signed-Rank Test resulted in $p < .001$), (2) there was no significant difference between the rated preference among the grouped actions

Table 4.17: List of situations for comparing task hierarchical representation styles.

- Overview the project to examine what kinds of tasks have been performed.
- Determine whether or not video clips had been trimmed from the original size.
- Determine the cause of the problem that we do not hear any sound.
- Determine what tasks are left in order to complete this video editing project.
- Examine kinds of operations performed on an audio clip called "audience laughter" (so you can perform the same operations to other audio clips).

(Friedman test: $p = .14$), and (3) there was no significant difference between abstraction levels (the pair-wise comparisons using Wilcoxon Signed-Rank Test, Style 2) vs. Style 3) $z = -1.85$, $p = .65$, and Style 2) vs. Style 4): $z = -1.73$, $p = .84$) .

Table 4.18: Comparisons of Grouped Views (Average rating out of 5).

no grouping	groups in text	groups (abstraction)	groups (abstraction, tree-style)
2.05	4.00	3.47	3.58

While this result is consistent with the result from Eval. 1 that contextual grouping of actions are more preferred to the views without grouped actions, it also showed that styles of hierarchical representations or abstraction levels did not affect the participants' preference: however, the number of performed actions were relatively small (87 performed actions) in this evaluation, and thus all the actions could be displayed on a screen without scrolling. When the task-view becomes larger and the user needs to scroll to browse all the actions, the text-based views with no abstraction could potentially present an issue of cluttering the screen estate. The same issue may also arise with the tree (node-link) style representation as it usually takes up more space than the space-filling style of hierarchical representation. For these reasons, I chose the space-filling style for representing task hierarchical structures for future implementations.

4.3.3.3 Evaluation 3: Comparison of Annotation Retrieval Methods

As a step toward the goal of integration of task-view annotations, in this evaluation, I compared different annotation retrieval methods. As discussed earlier, task-view annotations differ from features that allow users to add notes to data clips (as often seen in standard MMA systems) in that this current feature allows the annotation to be attached to operations in the task-views.

Types of Annotation Retrieval Methods

In this evaluation, I compared three different methods to provide annotation feedback:

1. Text-based annotation (displayed only text information)
2. Voice-based annotation (played back a voiced annotation)
3. Combination of text and voice annotation (displayed text as well as played back voiced annotation)

Figure 4.19 shows a text-based annotation retrieval method. Although there are other possible ways to provide annotations for task-views (e.g., drawing), these methods were chosen as they could provide less ambiguous annotations for all users. For example, seeing a line connecting two visual objects of subtasks can be less intuitive than explaining their relationships in words.

Figure 4.19: A screen-capture of the annotation retrieval methods (showing text-based annotation). When the user moves the cursor to an action (or a group of actions), it displays the annotated information in a pop-up window.

Procedure

I annotated a task-view with the following three pieces of information for this evaluation (i.e., the participants did not create the annotation):

1. description of previously performed tasks (e.g., "[these actions were for] preparation of audio Clip022.mp3")
2. editing decisions (e.g., "deleted because of unwanted lens flares")
3. future plans (e.g., "will need to normalize the volume once sequenced")

The participants obtained the above information (i.e., retrieved the relevant annotation) using the three methods described above, and rated each of the methods on a five-point Likert scale (strongly dislike to strongly like).

Results and Discussions

The results (shown in Table 4.19) show the participants' preference for text annotation over voice annotation (Wilcoxon: $z = -4.78$, $p =< .001$), but the rating for the combination of both the text and voice was neutral, and it did not show clear preference one way or the other (Chi-Squared Test: $p = .228$). Two participants stated that the text information was necessary and should not be omitted, while the voice feedback might be still good for some cases but was often distracting, thus there should be an option to toggle the voice on and off.

Table 4.19: Comparisons of annotation retrieval methods. The numbers in parentheses indicate the number of answers out of the total of 33 answers (3 annotation retrievals by 11 participants)

	strongly dislike	dislike	neutral	like	strongly like
Text	0% (0/33)	3.03% (1/33)	6.06% (2/33)	54.55% (18/33)	36.36% (12/33)
Voice	21.21% (7/33)	54.55% (18/33)	15.15% (5/33)	9.09% (3/33)	0% (0/33)
Text & voice	9.09% (3/33)	33.33% (11/33)	18.18% (6/33)	24.24% (8/33)	15.15% (5/33)

As this evaluation was an initial step toward integrating task-view annotations, there are still various aspects to be investigated. For example, I had observed (in User Study 1) that some advanced users used paper-based annotations for their task annotations, but how much a user can actually benefit from a digital tool for the same purpose is unclear. As well, some existing MMA systems (e.g., Ableton Live [1]) allow their users to add notes to clips and time-line markers, and how much improvement the task-view annotation can provide beyond these existing features is yet to be determined. Furthermore, the effectiveness of annotation greatly depends on users' communication skills, thus poor-quality annotations or inability to interpret them can confuse the outcome of such evaluations.

Although annotation can provide a variety of information that could address one or more proposed design guidelines as shown in Table 4.13, the other features were implemented to specifically address them as also shown in Table 4.13. As well, utilizing users' own annotation has already been shown to be effective for learning tools with complex UI's [24, 128]. Therefore, I focused on the rest of the features to be evaluated in User Study 3 (Section 5.2) in order to avoid possible redundancies.

4.3.3.4 Evaluation 4: Evaluation of Visualizing Prerequisites of Actions

For all the evaluations conducted in User Study 2, I restricted information on the precedence of actions to the lowest level to simplify the process of evaluating various design ideas. Actions were numbered in the order that they were performed for each project. When grouped, some actions appeared in more than one group: a user can perform a single action of *import clips* of both video and audio types at once, and thus this action appears in both the action groups for video processing and sound processing (for contextual views based on media types).

Visualization of Prerequisite Actions

As an initial step toward visualizing slightly more complex action precedence orders, I implemented a mockup of a feature to visualize prerequisite actions for a particular operation. This feature shows an appropriate set of actions/operations that should be performed in order to enable a particular target action. For example, when a user wants to change the speed of a clip but finds out that the menu item is greyed (i.e., disabled). In this case, the tool shows which action(s) needs to be performed to enable the disabled action when the user clicks on a greyed menu item. This feature is depicted in Figure 4.20.

Procedure

I asked the participants an open-ended question about this type of feature in helping them when they encounter similar situations.

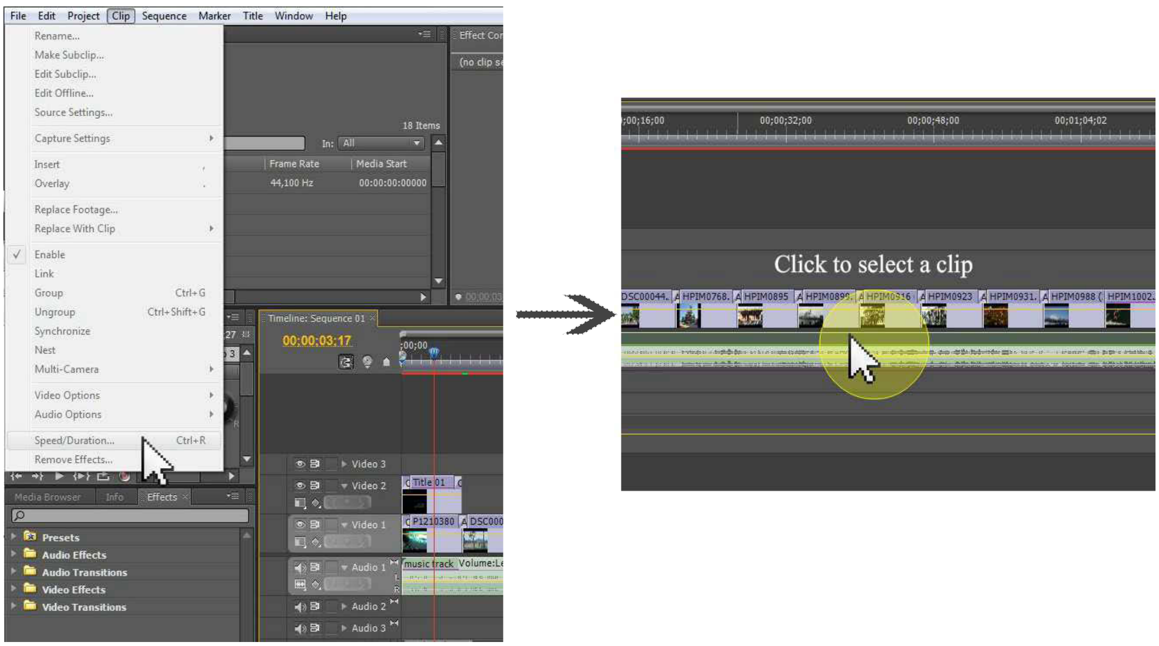


Figure 4.20: Visualization of prerequisite actions in order to enable an action for changing speed and duration of a clip. When the user clicks on a greyed (disabled) menu item in the MMA system's UI (shown on the left), the overlay component of the ITV displays which action(s) needs to be performed to enable the disabled action (shown on the right).

Results and Discussions

The responses from the participants were all positive, suggesting that this feature would potentially support users to quickly escape from situations in which they try to identify necessary solutions; however, a fully functional implementation of this feature might be very complex at many levels. For example, when the user's intention is at a more contextual level (e.g., "add a cutaway shot to avoid jump cuts"), it is difficult for the tool to correctly *guess* users' intention as there is no clear way for them to indicate it unlike the case of disabled actions. Even suppose if their intention was known somehow, it would be still challenging to visualize a complex *chain* of actions effectively so that the users can understand and remember what they need to do. This observation led me to design and implement a related feature (*operation previews* as described in Section 4.2.1) that displays a set of actions to complete an operation. The feature of operation previews was tested in User Study 3 (Section 5.2.3.2).

4.3.4 Summary from the Exploration of Design Features (User Study 2)

In User Study 2, I explored possible features that were designed based on the guidelines (developed in Chapter 3). More than to provide rigorous or detailed results, the purpose of User Study 2 was to: (1) indicate which design directions are promising, and (2) provide red flags, if any, about which design directions not to pursue, and thus address RQ2(b): *Which solutions are worthwhile implementing initially?*.

The study consisted of a set of tests to evaluate different design ideas of support features: different types of contexts for grouping actions (i.e., Contextual Views), different styles of hierarchical representations and abstraction levels, different annotation retrieval methods, and the visualization of prerequisite actions (Section 4.3.3.4).

When compared with a linear sequence of actions, the participants rated the contextual views more helpful, and they expressed that the contextual views enhanced the understanding of the performed tasks in spite of their lack of experience with MMA tasks. It was also found that appropriate context to be provided could vary depending on various factors such as work-flow patterns and subtasks being performed, thus an ideal support tool may need to accommodate these variations through different stages of an MMA project.

For the comparison of hierarchical representation styles and abstraction levels, the result indicated that participants preferred grouped actions while the difference in the task hierarchical representations and the abstraction levels of actions did not affect the participants' preference.

Among the three different annotation retrieval methods (text-based, voice-based, and the combination of both), the text annotation was most preferred. Two participants stated that the text information was necessary and should not be omitted, while the voice feedback might be still useful for some cases but was often distracting. The difficulty of evaluating the effectiveness of task-view annotations was also discussed.

I also implemented and tested a feature to visualize prerequisite actions for performing a particular operation. While the responses from the participants were all positive, the actual implementation of this feature may pose several difficulties such as properly determining which actions the users intends to perform and presenting a complex series of prerequisite actions.

The results from these evaluations allowed me to narrow the design space for certain features and provided a basis for determining designs for other features. In particular, the following features were selected for a prototype of the proposed MMA environment and evaluated in Chapter 5:

Interactive Contextual Views: Contextual views provided more preferred and informative representation for a task-structure than the linear sequence of actions (in User Study 2). By incorporating interactive components of the ITV (dynamic hierarchical views, execution of and expansion with new actions), interactive contextual views can support the majority of the proposed design guidelines. One important aspect that interactive contextual views do not provide, however, is a visible connection between the task-views and the corresponding workspace. For this reason, I integrated the two additional features for the prototype MMA environment (operation previews and visualization of task/workspace, discussed below), which were found effective during the cognitive walkthrough sessions and in User Study 2.

Operation Previews: Operation previews display a short demonstration of operations superimposed on the MMA system's UI. This superimposition allows users to view the relationship between the target operations and the corresponding UI windows and controls, and to make connection between the task-view and the workspace.

Visualization of Task/Workspace: Another support that I chose to incorporate is the visualization of the relationships of performed tasks and corresponding data in the workspace. This information was difficult for users to recall (in User Study 1). While users can use operation previews to make connections between the task and the MMA system's UI, they can utilize this visualization to make connections between the task and the data in the workspace.

I will present more detailed description of the prototype MMA environment in Section 5.1, and its evaluation in Section 5.2.

4.4 Conclusion

In this chapter, I have described the process to design an MMA environment with a support tool called the Interactive Task-Viewer (ITV). This new MMA environment augments an existing environment with additional task contextual information and interactions provided by the ITV. In this framework, the ITV has two distinct roles: it provides visualization of task-views that are constructed based on a given MMA project, and it provides alternate ways to interact with the MMA system (i.e., users execute actions and receive feedback through the ITV). In this new environment, the user interacts both with the ITV and the MMA systems. Interaction with the authoring system is as before while interaction with the ITV is designed to provide support for users performing meta-tasks by allowing the user to view the hierarchical and sequential structures of a task and control the main authoring system.

The initial design ideas were developed and evaluated through several iterations, including cognitive walkthroughs with low-fidelity implementations of various design features and User Study 2 that allowed me to further explore and compare different possible design configurations in order to select a set of ITV features (addressing RQ2(b)).

The results of User Study 2 indicated that contextual views (i.e., actions organized into contextually meaningful groups) were preferred over a standard undo list (i.e., a liner sequence of action commands) representation of the performed task. I also found that appropriate contexts could vary depending on various factors such as work-flow patterns and subtasks being performed, while differences in the hierarchical representation styles and abstraction levels did not influence participants preferences for the tested tasks.

I also identified certain issues to be addressed in future studies: while the text-based annotation retrieval method was most preferred, evaluating the effect of task-view annotations could be difficult as the quality in annotation itself may likely influence the results more than the difference of annotation methods. Implementation of a feature to visualize prerequisite actions may pose some difficulties to overcome such as properly determining intended actions and presenting a complex series of prerequisite actions.

As a result of this iterative design process presented in this chapter, I selected and implemented a set of features as parts of the ITV. The next chapter (Chapter 5) will provide the details of these features as well as their evaluations.

Chapter 5

Evaluating the New Multimedia Authoring Environment

In this chapter, I will first describe features that I selected for a prototype of the new multimedia authoring (MMA) environment (Section 5.1): interactive contextual views, operation previews, and visualization of task/data relationships. I designed these features based on the guidelines developed in Chapter 3, and selected them based on the results of the iterative design process described in Chapter 4.

I will then present the evaluation study of the new MMA environment (User Study 3 in Section 5.2). In User Study 2 (Section 4.3), I mainly focused on presentation approaches of the task-view itself (i.e., the first role of the Interactive Task-Viewer (ITV) as defined in Section 4.1). This was necessary in order to build a foundation before introducing and testing more interactive aspects of the ITV (i.e., its second role) in User Study 3. For this purpose, this prototype environment provides more interactive components than those explored in User Study 2, allowing testing with longer sequences of action commands.

5.1 Prototype of the New Multimedia Authoring Environment

The new MMA environment consists of a standard MMA system and the ITV with an overlay component that displays additional information on top of the MMA system's UI. This overlay helps users to make visible connections between the task-view and the workspace (e.g., video clips that were affected by a certain group of actions, or which UI window to use to change the speed of a clip), and it can manipulate how users access the MMA system itself (e.g., it can create a layer visually and functionally blocking direct access to certain clips).

I implemented this prototype environment using the Apache Flex SDK [123] for the ITV and its overlay. I selected Adobe Premiere Pro as the standard MMA system for this prototype environment for three main reasons: (1) it is an ideal target MMA system for this research because it uses complex interface designs and interactions such as those observed to be difficult for novices to use, (2) it is one of a few popular tools of this type that are available for both Mac and Windows, allowing for larger audience of the new environment, and (3) there is an abundance of rich help resources available. Figure 5.1 shows this prototype environment in use.

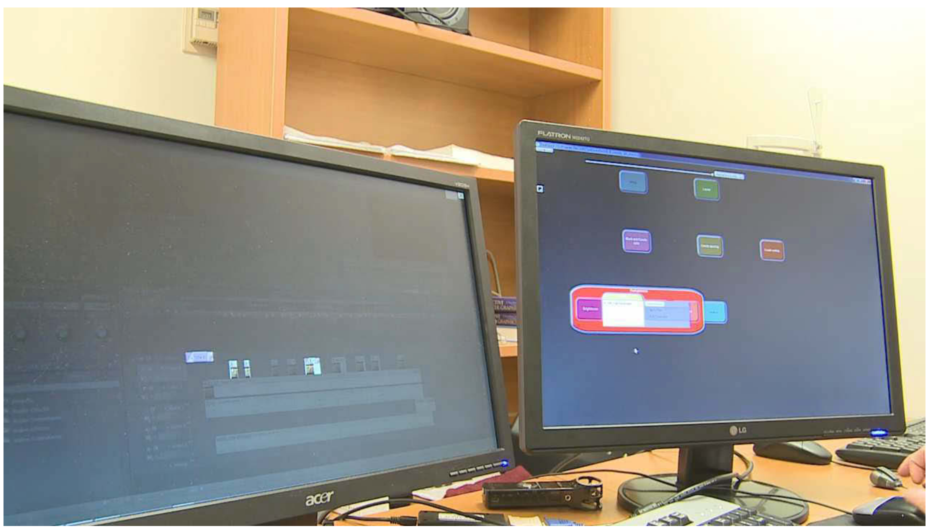


Figure 5.1: A prototype of the new MMA environment, consisting of a standard MMA system (Adobe Premiere Pro, on the left) , the ITV (on the right), and an overlay component that runs on top of the MMA system.

I will now describe each of the three features of the ITV for this prototype environment: interactive contextual views, operation previews, and visualization of task/data relationships.

5.1.1 Interactive Contextual Views

I chose the contextual views (Figures 5.2 to 5.4) as the basis of the task-view visualization for this prototype environment. User Study 2 (Sections 4.3.3.1 and 4.3.3.2) showed that, while there was variation between tasks, users generally found the contextual views preferable and informative. I did not, however, include the menu-based view (View D, as previously shown in Figure 4.15) in this prototype for two reasons. First, in User Study 2, the users had difficulties locating target actions in the menu-based view more often than the others. Second, as the new environment integrates an MMA system (Premiere Pro) with its menus, an additional interface providing the same context would have been redundant.

Each of these views represents a task-structure of the same project (e.g., editing of a comedy skit) with a different context, allowing multi-views of the same project. As the details of these views were described in Sections 4.2 and 4.3, I will only provide brief descriptions here.

1. View A: types of media (e.g., video and audio data) on which operations were performed. Users navigate in the task-view first by types of media, and within each media group (e.g., video data group), they can look at an action group for a particular clip (e.g., “clip002.mp4”), as shown in Figure 5.2.

2. View B: subtask headings. In this view, users first navigate in the task-view based on types of subtasks that are performed such as “Create cuts” and “Setup (of project/media).” Each subtask contains one or more sub-subtasks (e.g., “Project setup” or “Media setup”), creating 3-level hierarchical structure, as shown in Figure 5.3.

3. View C: final product’s time-line. In this view, users navigate in the task-view

based on the segmentation of the final video (e.g., the introduction segment or specific scenes). This view also presents 3-level hierarchy, as each segment is further divided into smaller segments, as shown in Figure 5.4.

Note that these task-views were prepared based on common task structures by the author and initially provided for novice users. The prototype system provides an editing environment (see Appendix B) for the task-structure, allowing addition and removal of action groups and manipulation of associated actions for these groups.

Figure 5.2: A contextual view based on types of media on which operations were performed. Users navigate in the task-view first by types of media (e.g., sound, video, etc.), and within each media group (e.g., a group of video data), they can look for an action group for a particular clip (shown here is for "Tape 2 - Clip 002.mp4").

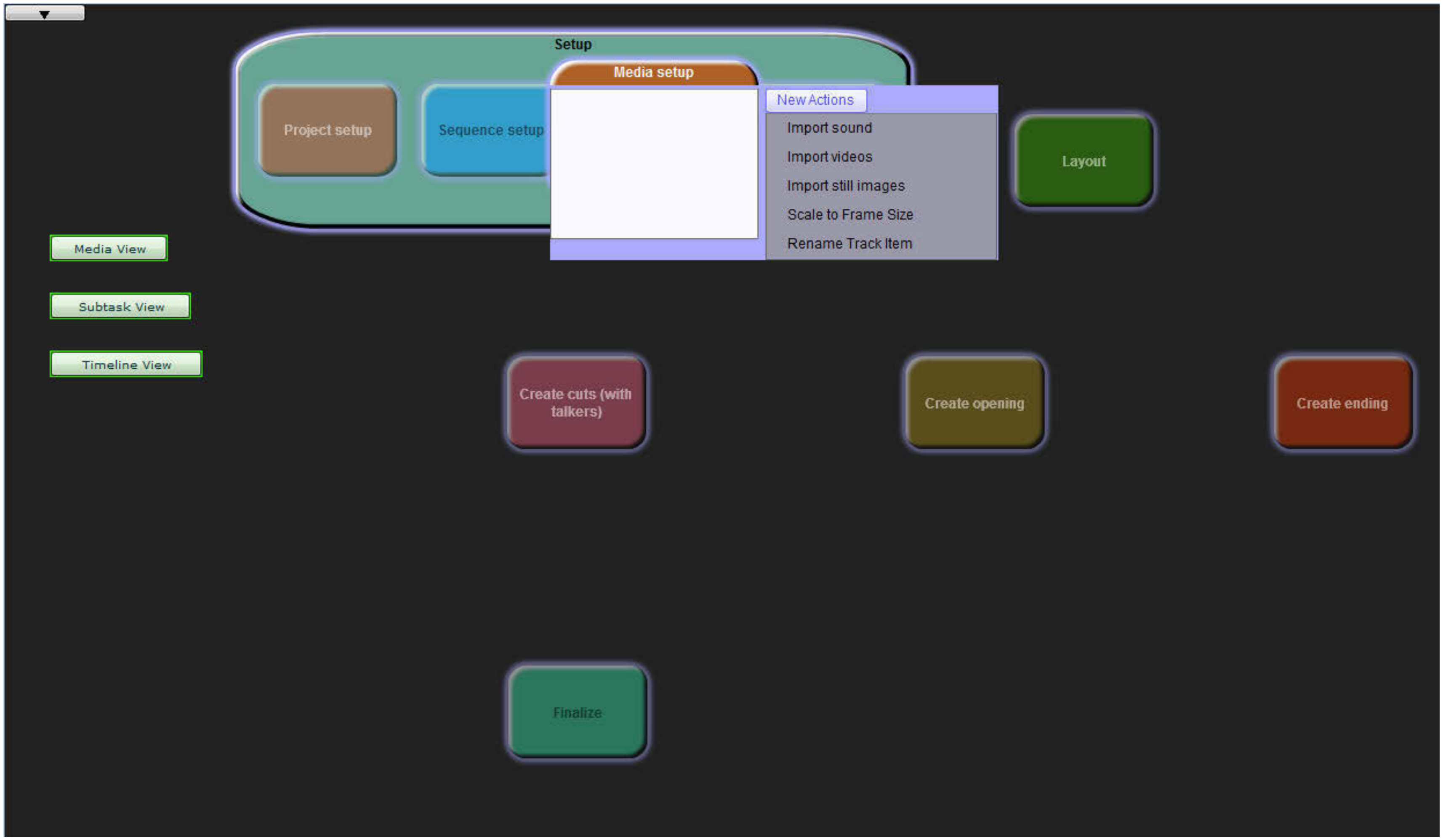


Figure 5.3: A contextual view with subtask headings. In this view, users first navigate in the task-view based on types of subtasks that are performed (e.g., “Setup”), each of which contains one or more sub-subtasks (e.g., “Project setup,” “Media setup,” etc) that groups relevant actions.

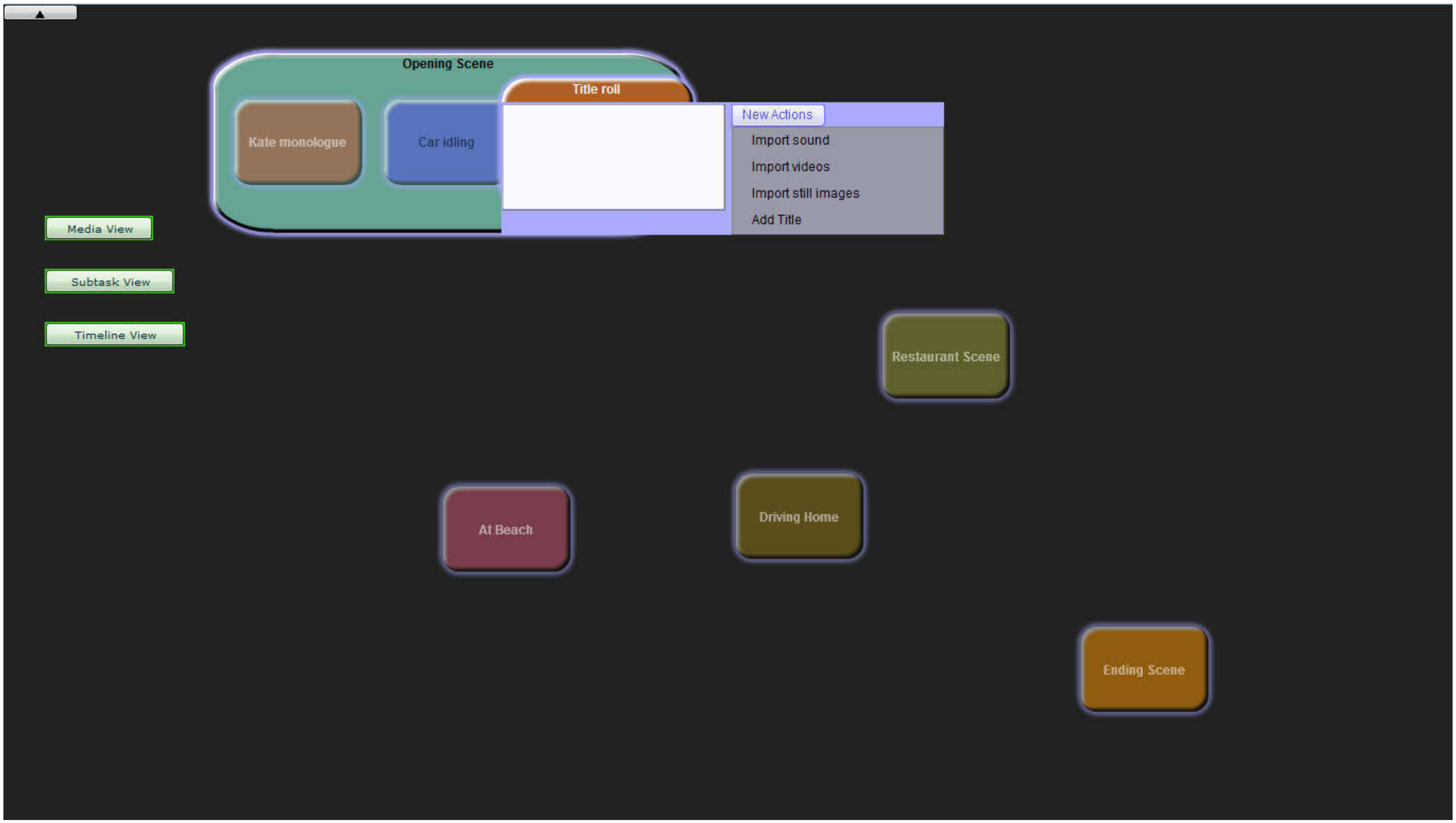


Figure 5.4: A contextual view based on final product's time-line. In this view, users navigate based on the segmentation (scenes) of the final video. In this example, they explored the opening scene, and then its sub-segment ("Title roll").

Users can switch between these contextual views interactively: allowing the users to choose a preferred view from a number of options is important, as I found in User Study 2 that users preferred certain contexts over others depending on various factors such as work-flow patterns and subtasks being performed. Within a selected view, the user can browse and navigate through a hierarchical structure, and then locate and execute actions within a contextually meaningful group (as described in Section 4.2.1).

5.1.2 Operation Previews

I incorporated the feature of operation previews in this prototype environment as well. An operation preview (as described in Section 4.2.1) is essentially a demonstration of an operation which consists of one or more actions. In this sense, the term *preview* itself refers to a preview of the *process* of executing actions, and thus, it is different from the previews showing the potential outcome of actions, commonly seen in MMA systems (e.g., a coarse render of how a cross-fade effect will look).

As interactive contextual views alone do not always provide support for making explicit connections between the task-views and the corresponding workspace, operation previews compensate this lack of support: operation previews are displayed on the overlay layer superimposed on the MMA system's UI, and thus they allow users to view the relationship between the target operations and the relevant parts of the UI windows and controls. Operation previews and the related feature of visualizing prerequisites of actions were found useful during the cognitive walkthrough sessions and in User Study 2.

With this feature, when a user selects a target action, the ITV can play a short demonstration of execution of the action on top of the UI (i.e., as a programmed sequence of action commands). This utilizes two layers of overlay (Figure 5.5): the first layer employs a filter to blur the background (i.e., the MMA system's interface) and the second layer displays the demonstration on top of the blurred layer. This keeps only the relevant part of the workspace in focus while maintaining sense of the overall configurations of the UI windows and controls (as opposed to completely masking the background).

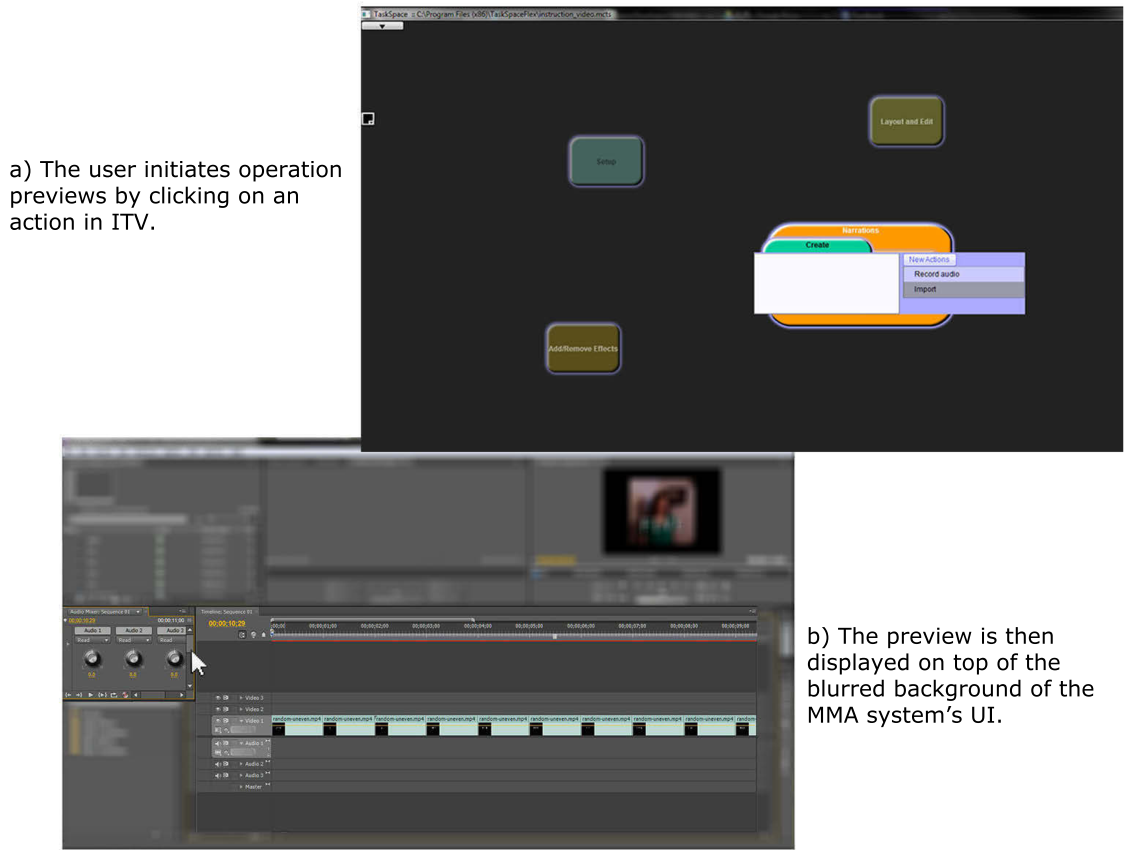


Figure 5.5: An operation preview in play. a) A user initiates an operation preview by clicking on one of the possible actions in the ITV (shown at the top), and b) the preview is then displayed on top of the MMA system's UI (shown at the bottom). Only the relevant part of the workspace was in focus and the other parts are blurred to make the previews stand out from the rather cluttered interface of the existing tool.

The second layer may also display constructed menu items as a part of the operation preview, superimposed on the real menus (Figure 5.6). These duplicated menu items on the overlay only simulate their behaviours to show the operations (e.g., open/close menus) without actually sending commands to the MMA system.

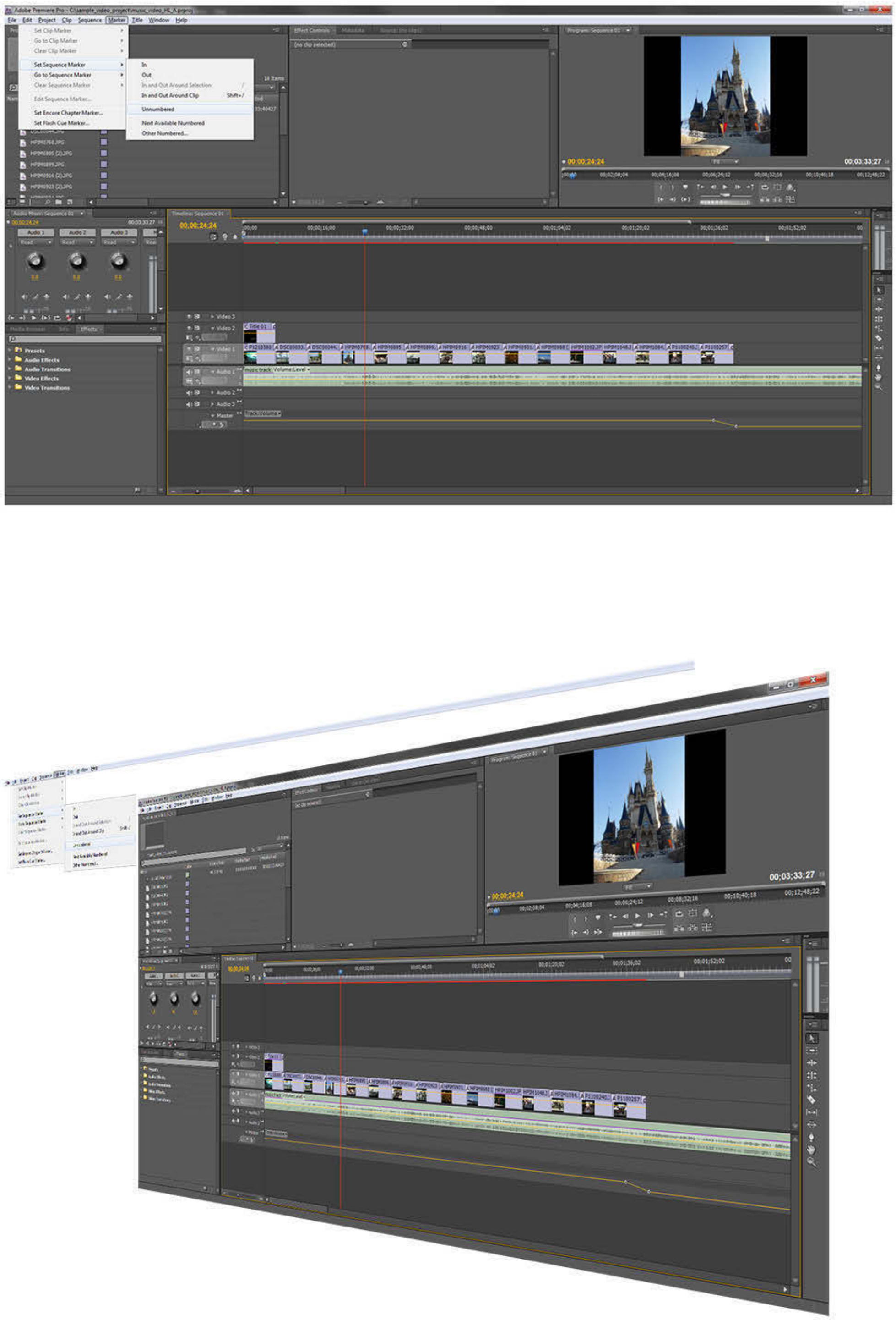


Figure 5.6: Menu overlay superimposed on the actual menus of the MMA system. The figure shows a static image of the interface as seen by the user (shown at the top), and the exploded view shows distinct layers (shown at the bottom). The menu overlay only simulates their behaviours (e.g., open/close menu) thus showing previews of action execution.

When combined with the contextual views, this feature also helps users to translate high-level goals into viable goals within MMA tasks, which is one of the identified meta-tasks (M2). For example, when a user has no idea what sequence of actions to perform in order to record a video, they typically either explore different functions within the MMA system to see if they can find appropriate actions, or they seek external help resources such as instructional videos or help files. In the proposed MMA environment, the user can first narrow the search space of actions by utilizing the contextual views, and then initiate previews of selected operations to see which actions need to be performed and in what order, indicating sequential constraints on certain actions.

5.1.3 Visualization of Task/Data Relationships

The last feature that I incorporated in this prototype is the visualization of relationships between performed tasks and their corresponding data. This feature also provides visual connections between the task-view and the workspace, but its focus is on previously performed actions and data in the workspace. This is in contrast to operation previews that focus on the connections between new/possible actions and the MMA system's UI. The results from User Study 1 (Section 3.1) showed that keeping track of these relationships was difficult even for advanced users, and making this information more accessible was useful for understanding a task in the cognitive walkthrough sessions (Section 4.2.2), as it provided additional contextual information.

For this feature, I utilized the overlay framework that was already implemented for operation previews: when the user interacts with relevant contextual views in the ITV, data that were affected by an action (or a group of actions) are highlighted in the workspace while all the other parts of the workspace are greyed out as shown in Figure 5.7. The overlay component enables this effect by creating a layer with transparent areas that expose relevant data and a semi-transparent area that greys out the rest of the rather cluttered workspace (Figure 5.8). A similar overlay approach has been also used for menu items (e.g., StencilMaps [110]); however, in this implementation, it is utilized for highlighting a subset of numerous data clips, and the overlay is triggered by the users interacting with the ITV in order to visualise the connection between previously performed actions and corresponding data in the workspace. If needed, the overlay on the MMA system can also limit the users' direct interaction with data; e.g., by only allowing them to manipulate highlighted data and blocking access to greyed ones.

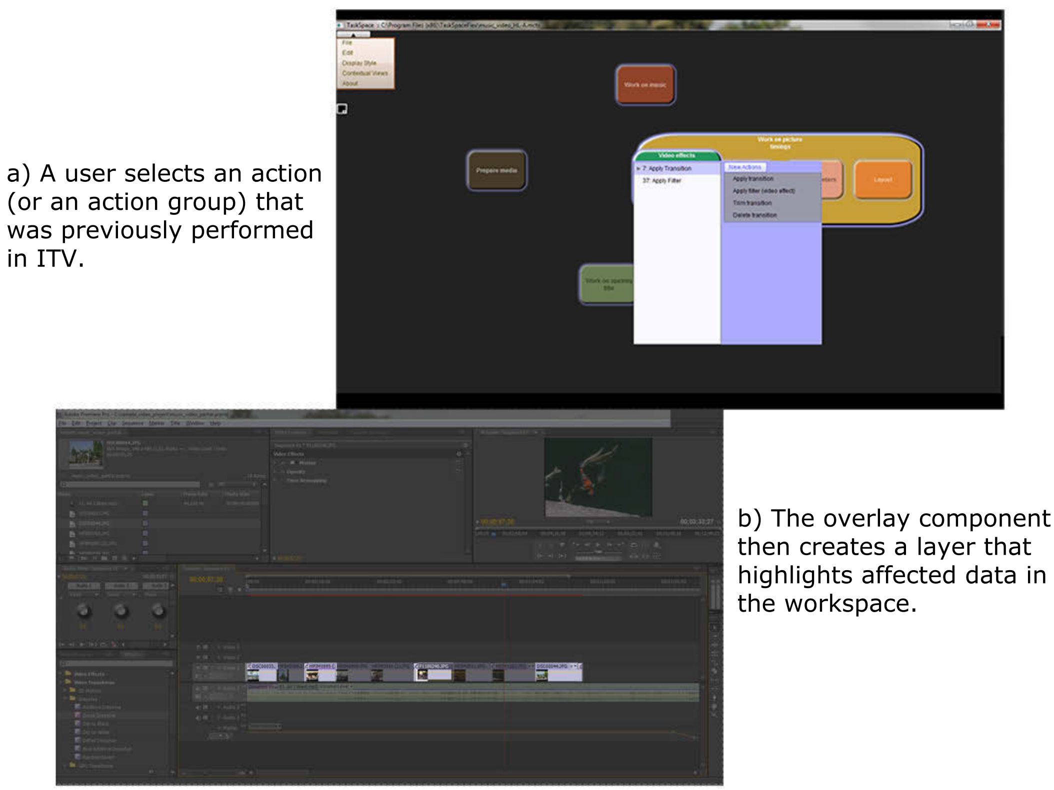


Figure 5.7: Premiere Pro with the highlighted data. a) A user selects an action group of "Apply Transition" (shown at the top), and b) the overlay highlights the data that are affected by the group of actions by creating a semi-transparent filter that greys out the rest of the rather cluttered workspace (shown at the bottom).

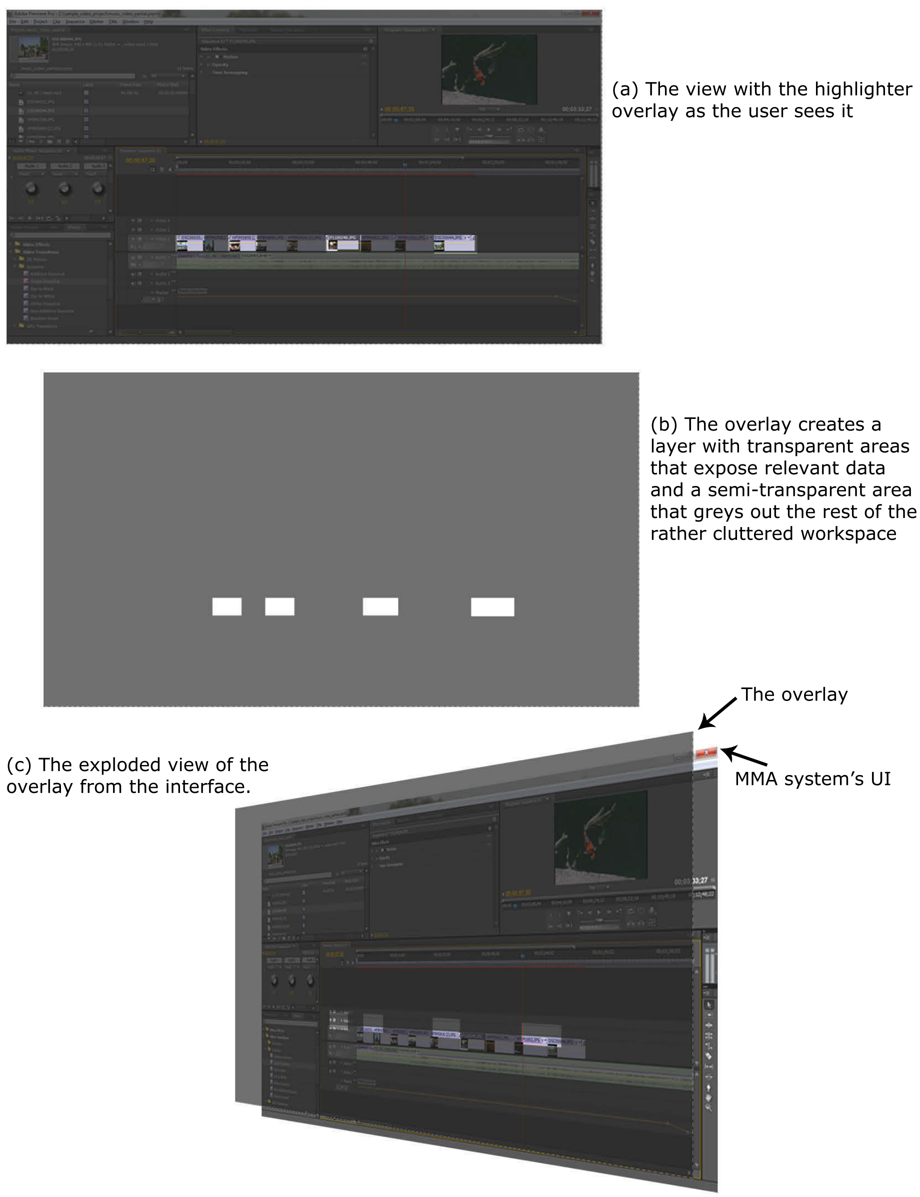


Figure 5.8: The highlighter overlay on an existing MMA tool. (a) The transparent areas highlight relevant data while the grey area masks the rest of the cluttered workspace. (b) The overlay creates a layer with transparent areas that expose relevant data and a semi-transparent area that greys out the rest of the rather cluttered workspace. (c) The image at the bottom shows the exploded view of the overlay from the interface. Note that the printed version of the image appears much darker than on screen; in practice, the greyed-out area is still clearly legible, but simply lightly greyed out.

5.1.4 ITV features, Novice-Specific Issues, and Design Guidelines

I identified the challenges experienced by novices in User Study 1 (shown in Table 5.1) based on the framework I developed in order to explain these challenges in regards to meta-tasks. Based on this framework, I then developed a set of design guidelines (shown in Table 5.2) aimed at addressing these challenges that stem from meta-task difficulty. I designed the above ITV features based on these guidelines and selected them based on the results of the iterative design process described in Chapter 4. Table 5.3 shows the implemented features as well as the design guidelines and novice-challenges that these features are designed to address. As seen in this table (Table 5.3), the interactive contextual views are the core features of the ITV, addressing the majority of the guidelines.

As discussed in Section 3.3, Guideline 1 ("Provide additional contextual information where possible," G1) is perhaps the most abstract guideline, and it may result in various design ideas. Other than the interactive contextual views, I chose two other features in this prototype implementation: operation previews and the visualization of task/data relationships. These features complement the interactive contextual views by visualizing connections between the task-view and the corresponding workspace; operation previews mainly focus on the connection between new/potential actions and the MMA system's UI while visualization of task/data relationships focuses on the connection between previously performed actions and data in the workspace. They also together address several frequently observed novice-specific challenges (Challenge-3 to Challenge-6) that are not fully addressed by the contextual views. For example, operation previews display appropriate combinations of actions needed to accomplish a goal (Challenge-6) while the visualization of task/data relationships can help users to understand the relationships between data clips and operations that affected them (Challenge-3). As well, providing a number of options to *address* the same issues can presumably increase the chance of actually *solving* them, as addressing an issue does not guarantee that the user can actually solve it.

Table 5.1: A list of novice-specific challenges identified in Section 3.2.3.

challenge-1)	Identification of concrete task-flow.
challenge-2)	Maintaining a mental model of a task and retaining contextual meanings of tasks.
challenge-3)	Understanding relationships between data clips and performed operations.
challenge-4)	Recovery from error.
challenge-5)	Recognizing dependencies: Novice users had difficulties understanding what other actions needed to be performed prior to their target operation.
challenge-6)	Recognizing combinations of actions needed to accomplish a certain goal.
challenge-7)	The contextual information of the task is not available in current tools.
challenge-8)	Task interruption could leave users disoriented when they return to the system.

Table 5.2: A list of design guidelines developed in Section 3.3.

G1.	Provide additional contextual information where possible.
G2.	Show task-decompositions and structures.
G3.	Provide explicit task-views that can dynamically change to show different levels of abstraction, scopes of tasks, and task contexts.
G4.	Provide common task-structures that allow integration of newly performed actions.
G5.	Indicate possible next subtasks/actions.
G6.	Provide a means to carry out actions within contextually meaningful views.
G7.	Show sets of common operations.

Table 5.3: The implemented ITV features, the corresponding design guidelines that the features were based on, and the novice-specific challenges that they are designed to address.

ITV features	**Guidelines**							**Novice-specific challenges addressed**							
	G1	G2	G3	G4	G5	G6	G7	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8
Interactive contextual views	✓	✓	✓	✓	✓	✓		✓	✓	✓[1]	✓			✓	✓
Operation Previews	✓				✓		✓				✓	✓	✓	✓	
Visualization of Task/Data Relationships	✓									✓				✓	

[1]Partially supported by utilizing the contextual view based on media-types.

This section described features of the new MMA environment. In the next section (5.2), I will present the usability study (User Study 3) to evaluate the proposed environment.

5.2 Evaluation of A Multimedia Authoring Environment (User Study 3)

5.2.1 Purpose of the Study

While User Study 2 mainly focused on presentation approaches of the task-view in order to build a foundation, in User Study 3, I investigated the usability of the new MMA environment with the prototype of the ITV features described in the previous section, with more focus on the interactive aspects of the ITV.

In User Study 3, I aimed to answer RQ3: *Do the proposed solutions help novice users?*

(a) *Do they help novices to complete MMA tasks faster?*

(b) *Do novices find the new approach easier to use?*

(c) *Does the new approach successfully address the challenges identified in RQ1?*

Therefore, I evaluated the new MMA environment on the following three key aspects:

(a) **Task performance**: to see if and how the new environment would improve novice users' task performance over the standard environment. To evaluate task performance, I primarily focused on task completion time, but I additionally measured errors (e.g., failed attempts while completing tasks). There are several other common ways to measure task performance such as quality of outcome, frequency of use of support, and learning effect [47, 55, 113]; however, for evaluating task performance of novice users who are unfamiliar with an authoring system, I was primarily interested in whether or not the new environment could enable users' performance of fundamental operations (e.g., find and execute basic actions). Shorter task completion time suggests that the new support helps users to complete tasks faster than the standard environment without ITV support, and fewer failed attempts suggest that the new support guides users in the correct direction.

The other measures such as those listed above may be useful once I confirm this primary support of the new environment. For example, the quality of outcome is useful information but it can only be meaningfully evaluated after these novice users become familiar with basic MMA tasks: e.g., in the preliminary studies that I conducted with two novice users prior to User Study 3, without ITV support, neither participant could properly proceed to the point where they produce sufficient outcome to be evaluated. As well, how the new environment affects the learning process of existing MMA systems is not a primary focus of the current research; my initial focus is to allow novice users to start using existing tools by capitalizing on the support provided by the ITV. Similarly, the frequency of use of support is an interesting measure to investigate but such an investigation makes sense only after we establish that the new support approach does in fact help novices use MMA systems.[2]

(b) **Ease of use**: to see if novice users would find the methods provided by the new environment easier to perform tasks compared to those in the standard environment.

(c) **Meta-task support**: to explore kinds of meta-task support provided by the new environment for addressing novice challenges while performing certain MMA tasks. Unlike the previous two aspects, this requires a qualitative analysis of participants' behaviours as well as their comments.

[2]Based on the survey of current practice in measuring usability of interfaces [55], task completion time, accuracy, and error rates were measured in 57%, 31%, and 26% of the studies reviewed, respectively, while quality of outcome, frequency of use, and learning effect were measured in 16%, 13%, and 3% of the same reviewed studies.

5.2.2 Study Settings

5.2.2.1 Study Designs

I conducted the following five evaluations (Eval.1 to Eval.5) in User Study 3, which are divided into three parts[3] and are summarised in Table 5.4.

I) Within-subjects comparisons against the standard environment

Evaluations of ITV support for:

- Eval.1) tasks with local scope such as finding and executing basic actions and recovering from unwanted situations (Section 5.2.3.1).
- Eval.2) tasks with slightly broader scope than those evaluated in Eval.1. These tasks involve multiple actions such as performing an operation with a sequence of actions and recognizing action dependencies (Section 5.2.3.2).
- Eval.3) understanding relationships between data clips and performed operations (Section 5.2.3.3).

II) Case-study

- Eval.4) An evaluation of ITV support in general MMA, requiring maintenance of a task-view, retaining contextual meanings of tasks, and resuming tasks from an interruption (Section 5.2.4.1).

III) Qualitative video analysis

- Eval.5) An analysis and summary of participants' comments and behaviours from all the above four evaluations with respect to support for meta-tasks and addressing novice-specific challenges (Section 5.2.5.1).

5.2.2.2 Participants

Sixteen participants (8 males and 8 females, 19 to 40 years of age), recruited through posters and email advertisements, took part in User Study 3. All participants were novices (less than one year of experience in general MMA, and no experience with

[3] This division is based on types of experimental designs of the studies for organizational purposes.

Table 5.4: Study designs and the ITV support evaluated in User Study 3.

Study Design	**Evaluations (Sections)**	**Support for/in ...**
Part I: Within-subjects comparisons	Eval.1: Locating and Executing Basic Action Commands (Section 5.2.3.1)	tasks with a local scope
	Eval.2: Executing a Sequence of Actions (Section 5.2.3.2)	tasks with a slightly broader scope (e.g., performing a group of actions)
	Eval.3: Understanding Task/Data Relationships (Section 5.2.3.3)	understanding relationships between data clips and performed operations
Part II: A case-study	Eval.4: Using the New MMA Environment in General Authoring Tasks (Section 5.2.4.1)	general MMA, requiring maintenance of a task-view, retaining contextual meanings of tasks, and resuming tasks from an interruption.
Part III: Qualitative video analysis	Eval.5: Summary and Discussions on Participants' Comments and Behaviours (Section 5.2.5.1)	performing meta-tasks

Premiere Pro). The participants are from diverse backgrounds: six participants had a background in computer science or engineering, and the other participants had a background in non-technology related areas of studies and/or professions.

5.2.2.3 Study Equipment

The participants performed the evaluation tasks on a personal computer and as described, Adobe Premiere Pro was selected as a standard authoring system that ran in conjunction with the ITV in this new environment.

5.2.3 Part I: Within-subjects comparisons

The first three evaluations (Eval.1 to Eval.3) were comparisons of the new environment against the standard environment (Sections 5.2.3.1 to 5.2.3.2). The description of an MMA project (shown below) was given at the beginning of each 1-hour study session, and the participants examined the corresponding raw video and audio clips mentioned in the description. They also watched the completed sample output of this project prior to performing any tasks.

> **Project 1: Comedy skit video editing**
>
> *In the next hour or so, you will be working on a video editing project to create a comedy skit. There are 2 video footages, which were recorded from 2 separate video cameras. Each camera was stationary at one location and recorded each of the 2 characters who appear in this skit.*
>
> *These footages are long-takes, meaning each clip shows one character for the entire duration of the skit.*
>
> *There is also an audio clip that captured the entire dialogue using a separate audio recording device. This audio recording contains higher quality data and it is to replace any audio recorded on the video cameras.*

> *We ask you to edit these videos so that when played back, all the scenes will show a character who is currently talking. Also, as described above, the high-quality audio data should replace the one that comes with the video footages.*
>
> *You can first browse the raw data (2 video and 1 audio clips) to get an idea of what kind of data you will be dealing with, as well as the sample output of the final product of this project.*
>
> *If at any time you have a question, either regarding the video editing project itself, or about the authoring tools that you will be using, please don't hesitate to ask me.*

For each session, the experimenter first showed the participants how to use relevant features of both the standard UI and the ITV by asking them to perform two example tasks for each tool. The participants then performed a set of tasks for each of the evaluations. I chose these tasks based on common tasks that occur during common MMA projects, and particularly those observed in User Study 1 (Section 3.1). In these evaluations:

(a) I measured and compared task performance (e.g., time to complete the tasks and errors)

(b) I measured and compared ease of methods provided in the environments for performing the tasks (participants rated on a Likert scale)

(c) I observed user behaviours for any indication of types of meta-tasks performed and how they would be supported

I evaluated the above aspects in two conditions: the new proposed environment with ITV support, and the standard environment without ITV.

I collected repeated-measurements and used a within-subjects design to perform the comparisons. I analysed user behaviours in relation to meta-tasks. I report both qualitative observations and statistical analyses: for all the statistical analyses used in User Study 3, the significance was tested at a level of $\alpha = .05$. I will now describe each of these first three evaluations (Eval.1 to Eval.3) in more detail.

5.2.3.1 Evaluation 1: Locating and Executing Basic Action Commands

I evaluated the effect of the ITV support for tasks with a local scope (e.g., finding and executing basic actions as opposed to examining the overview of the project) by comparing the user's performance of locating and executing basic action commands in two conditions:

1. Using the interactive contextual views that grouped actions as described in detail in Section 5.1.1 (Figure 5.9)
2. Using Premier Pro's standard UI (such as menus, tool-bars, right or double-clicks, and keyboard shortcuts)

Figure 5.10 shows this second method of performing the same action (done here by a right-click menu).

Procedure

In each condition, the participants performed a set of eight common actions listed in Table 5.5. Ordering effects were minimized by randomizing the order of actions with a Latin square and by having half of the participants start the task in the new environment while the other half started with the standard UI. The duration of time to complete each task and the numbers of failed attempts were measured and compared. The participants also rated the ease of each method on 10-point Likert scales (very difficult to very easy). I also qualitatively observed general task performance behaviour.

If a participant indicated that they were unable to locate target actions after five attempts[4], the experimenter showed the participant the answer and asked them to

[4] This number was estimated to be reasonable from the earlier observations.

Table 5.5: A list of basic actions that each participant was asked to perform.

A) You found a part of the clip that shows a character who is talking. You will need to cut out this part using the “razor” tool (cut at current time indicator).

B) You are finding parts of the clips that show the characters talking. You will need to add sequence markers to indicate locations to be cut.

C) When you are first preparing the project, you find that an additional video track is needed. Add a video track.

D) The transition effect, which you have applied to the audio clip of the audience laughing and applauding used in the ending part of the video, turned out to be not ideal. Remove this transition effect.

E) You have created a title clip for the opening of the video. Now apply (any) video transition effects to this clip.

F) You are starting to work on the opening scene of the video. You want to add the title of this skit (“Dentist”) at the beginning. Create a new opening title.

G) You have created a new project and now are trying to bring in all the media clips that you intend to use. First, you will need to import video clips.

H) The font used for the opening title was hard to read with the background of the video. Change the font of the opening title.

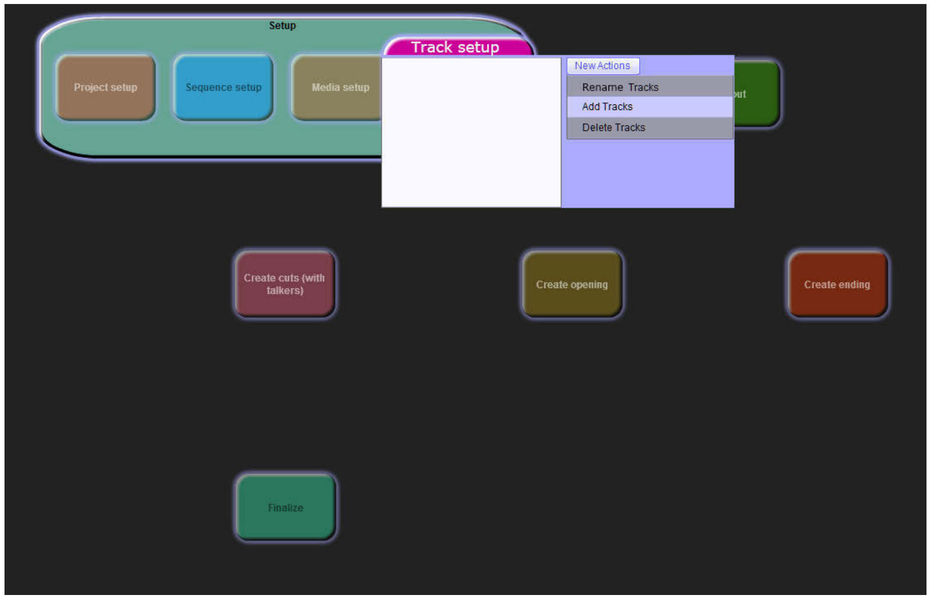


Figure 5.9: A contextual view used to find an action *Add tracks*. Actions are grouped into a hierarchical structure to convey task contextual information. In this example, users first explore the subtask "Setup," and then sub-subtask "Track setup," in which they can select the target action (*Add tracks*).

move on to the next action. This intervention was carried out in order to mitigate the possibility of the participant becoming frustrated by not being able to perform the task, a situation that was commonly observed in User Study 1.

The research questions for this evaluation were:

(a) *In what ways would the new environment improve task performance for finding and executing actions compared with the standard UI?*

(b) *Would the participants find the new environment easier to perform these tasks?*

(c) *Would the participants' behaviours in each environment indicate any particular meta-tasks and would there be any difference in meta-task performance?*

Results and Discussions

Task performance

Table 5.6 and Figure 5.11 show the results of the comparisons between the new environment and the standard interface for execution of basic actions. As the collected data were non-parametric (by Shapiro-Wilk test) and exhibited non-homogeneous

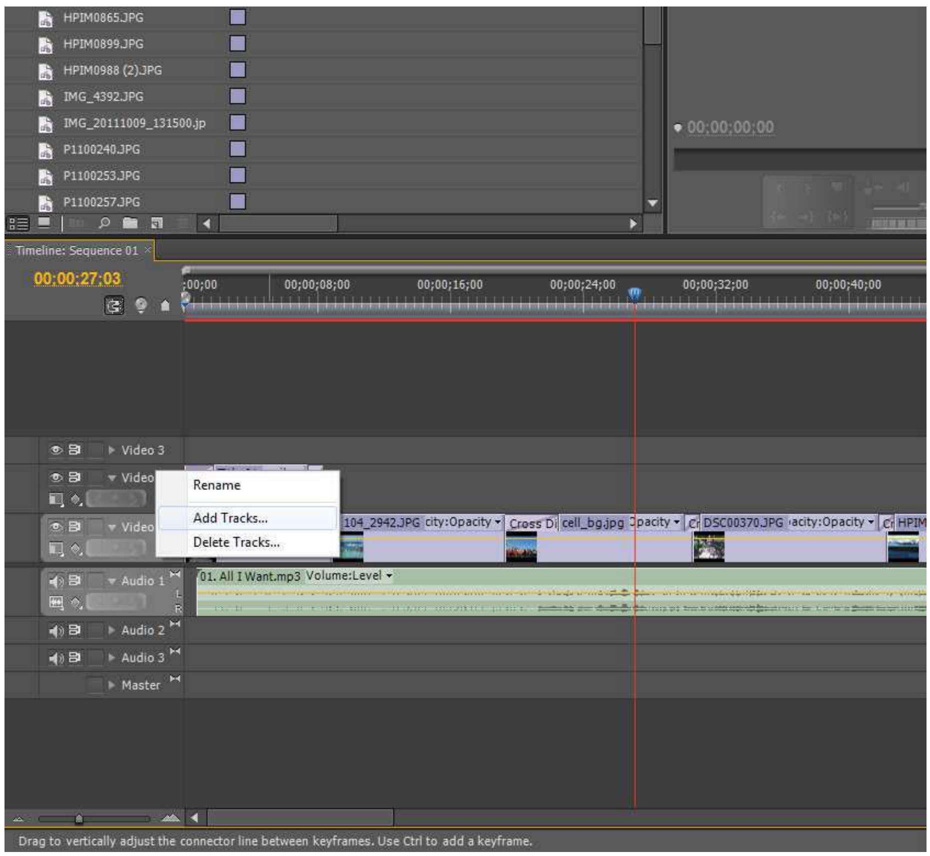


Figure 5.10: One of standard methods to find and execute the same action *Add tracks*. In this example, the user right-clicks on one of the existing tracks to open the pop-up menu, from which the user selects the target action.

variances (by Levene's test), Wilcoxon signed-rank test was used for the data analyses in this evaluation.

Table 5.6: Mean performance-time (in seconds) for executing actions in the new environment and the standard environment. The p-values were calculated using the Wilcoxon signed-rank test. The new environment yielded significantly faster task completion time for half of the actions (*Group I*: Actions A, C, E, and H, indicated by the shaded rows in the table), but there was no statistically significant difference between the two conditions for the rest of the actions (*Group II*: Actions B, D, F, and G). Note these actions were grouped and highlighted based on the results for the clarity of discussions, but this grouping was not hypothesised prior to the evaluation.

	Mean (μ) performance-time (in seconds)		
Action performed	New environment	Standard environment	p-value (bold when $p < 0.05$)
A) Razor (cut at current time indicator)	18.32	69.50	**<.001**
B) Add a sequence marker	28.60	16.52	.06
C) Add a video track	35.58	67.81	**.049**
D) Remove an audio transition effect	27.66	27.44	.30
E) Apply video transition effects	21.66	74.85	**<.001**
F) Create a new opening title	16.24	18.41	.84
G) Import video	25.86	15.20	.16
H) Change the font of the opening title	16.58	80.72	**<.001**

While the new environment yielded faster task completion time for half of the actions (Actions A, C, E, and H), there was no statistically significant difference between the two conditions for the rest of the actions (Actions B, D, F, and G). I call the first group of actions (A, C, E, and H) *Group I*, which benefited from the ITV support (indicated by the shaded rows in Table 5.6), and the other group of actions

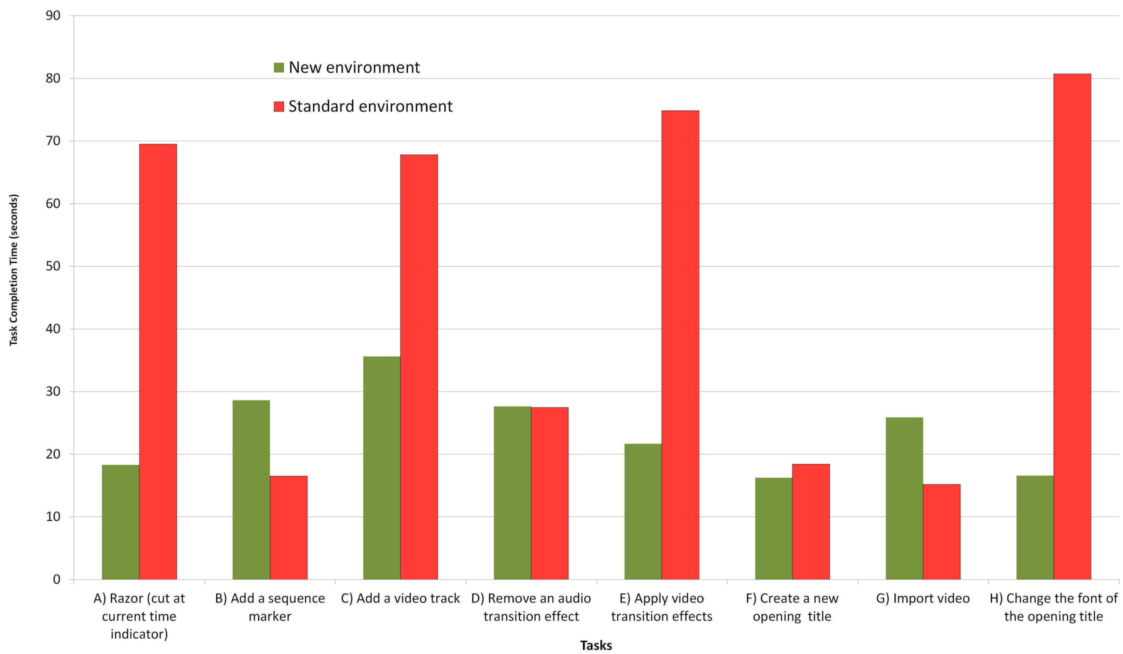


Figure 5.11: Mean performance-time (in seconds) for executing actions in the new environment and the standard environment.

(B, D, F, and G), *Group II* which did not benefit from the ITV support.

In the standard environment, the participants performed the actions in Group II significantly faster than those in Group I (pair-wise comparisons between action completion time using the Wilcoxon signed-rank test), and they generally had less difficulty with finding the Group II actions. This can be confirmed by the average numbers of failed attempts in the standard UI (shown in Table 5.7): for the Group I actions, the average numbers of failed attempts (2.00, 2.19, 1.44, and 3.06, respectively) were larger than those of the Group II actions (0.31, 0.44, 0.25, and 0.31). Comparing these numbers measured in the new environment (Group I: 0.19, 1.31, 0.31, 0.13, and Group II: 0.75, 1.00, 0.66, 0.63), the participants made significantly more errors in the standard UI for three out of four of the Group I actions, and this error difference can be one plausible reason that the participants took significantly longer time to complete the Group I actions in the standard UI.[5]

[5]Pearson product-moment correlation coefficient between the performance-time and the numbers

Table 5.7: Average numbers of failed attempts measured in the new environment and the standard environment. The p-values were calculated using the Wilcoxon signed-rank test. The participants made significantly more errors in the standard UI for three out of four of the Group I actions.

	Average numbers of failed attempts		
Action performed	New environment	Standard environment	p-value (bold when $p < 0.05$)
A) Razor (cut at current time indicator)	0.19	2.00	**0.004**
B) Add a sequence marker	0.75	0.31	0.23
C) Add a video track	1.31	2.19	0.20
D) Remove an audio transition effect	1.00	0.44	0.39
E) Apply video transition effects	0.31	1.44	**0.03**
F) Create a new opening title	0.66	0.25	0.27
G) Import video	0.63	0.31	0.50
H) Change the font of the opening title	0.13	3.06	**0.001**

A closer look at the Group II actions illuminates possible reason that the participants had less difficulties with finding the Group II actions in the standard environment: two actions in Group II were "Import video" and "Create a new opening title": "Import" can be found in the menu item called "File," which, in almost any type of application, includes actions to open/import/insert existing files. "Create a new title" can be accomplished by selecting "new " in the "Title" menu, which was the first menu item that 10 out of 16 participants tried. The other two actions in this group, "Remove an audio transition effect" and "Add a sequence marker" can be accomplished from the right-click menu. Therefore, the participants appeared to be able to find and perform the Group II actions quickly even in the standard environment, thus the ITV support is typically unnecessary for these actions.

Table 5.8 shows the results of task-by-task comparisons of the variances (σ^2) using Levene's test. These results show that (1) for the Group I actions (shaded rows in Table 5.8), the variances of the performance time in the new environment were consistently smaller than those in the standard environment, and (2) for the Group II actions, there were no significant differences in the variances between the conditions. Friedman tests between actions within each environment reveal that there was no statistically significant difference in performance time among actions when performed in the new environment ($p = .16$) but there was in fact significant difference when performed in the standard environment ($p < .001$). This result is complemented by Levene's tests among the variances showing that there was no statistically significant difference in variances of the performance time in the new environment ($F = 1.648$, $p = .13$) but there was significant difference in the standard environment ($F = 6.419$, $p < 0.001$). This difference is also likely attributed to the difference in the numbers of failed attempt (80 failed attempts in the new environment, and 160 in the standard

---

of failed attempts were 0.691 (new environment) and 0.725 (standard environment).

environment). Thus, based on all these tests (i.e., in the new environment, no significant difference in the task completion time itself or in variances of task completion time between the actions), it can be concluded that the contextual views provided a more consistent approach for finding target actions regardless of the difficulty of actions that were tested. As well, for the Group I actions (which benefited from the ITV support), the new environment led to much more consistent task completion time, but it did not negatively impact the performance of those actions in Group II in a significant way, for which the ITV support is not generally needed.

Table 5.8: Task-by-task comparisons of performance-time variances (σ^2). The p-values were calculated using Levene's test. The new environment provided a more consistent approach for finding target actions regardless of the difficulty of the actions that were tested.

	Task-by-task comparisons of performance-time variances (σ^2)		
Action performed	Contextual views	Standard UI	p-value (bold when $p < 0.05$)
A) Razor (cut at current time indicator)	400.22	3955.39	**<.001**
B) Add a sequence marker	635.31	216.95	.44
C) Add a video track	622.80	2714.83	**<.001**
D) Remove an audio transition effect	354.19	1599.52	.29
E) Apply video transition effects	204.59	2336.56	**<.001**
F) Create a new opening title	219.63	298.26	.95
G) Import video	534.21	180.62	.38
H) Change the font of the opening title	97.79	2332.05	**<.001**

Based on these results, I conclude that utilizing the new environment's support successfully improved the user performance for tasks that they had more difficulty in the standard environment while it still provided a comparable method for performing tasks that the users had less difficulty in the standard environment. The new environment also allowed more consistent task performance than the standard environment.

Ease of use

Overall, the participants found it easier to perform the tasks in the proposed environment (mean ratings of the contextual views and the standard UI were 8.20/10 and 6.49/10, respectively. Wilcoxon signed-rank test: $z = -6.05$, $p < .001$).

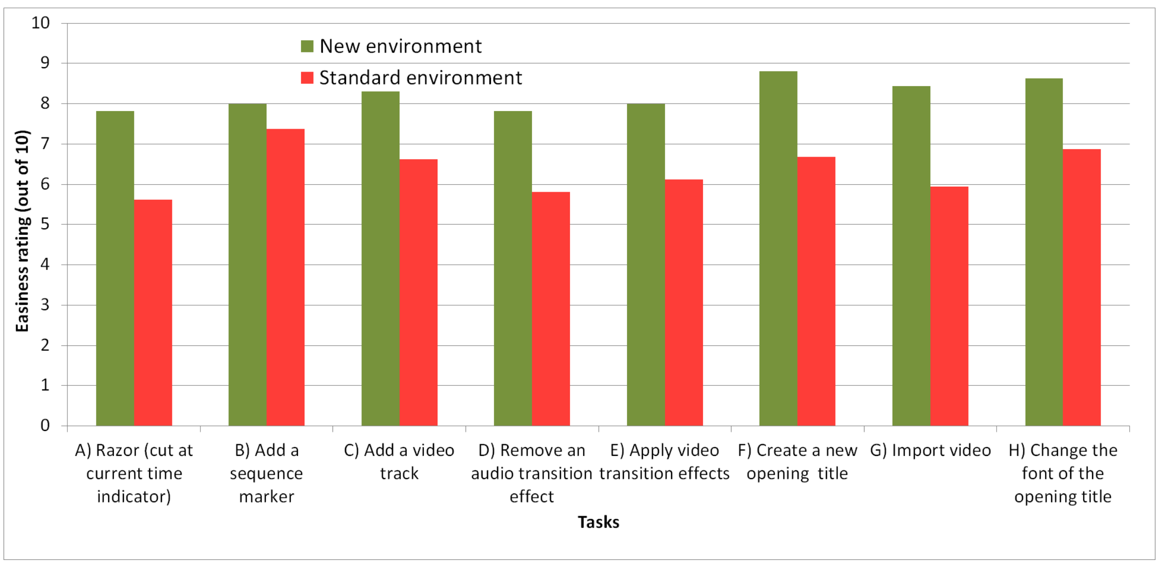


Figure 5.12: Ratings for ease of use (out of 10) for executing actions in the new environment and the standard environment.

Although no order effect was found in either condition, it appeared that the participants needed extra time to get used to the concept of the contextual views despite there being the same number of practice tasks (two) for each condition: nine out of sixteen participants at least showed the behaviour of scanning through different action groups in the first few tasks, while I did not observe this behaviour in the later tasks. The likely explanation is that while all the participants knew the standard menu approach, the contextual views were new for them and they needed to familiarise themselves with this new UI. Given that the participants performed tasks faster with the contextual views on average, however, it can safely be assumed that the participants adapted to this new approach by the end of the evaluation sessions. Thus, some of these results might be conservative: if the participants had more practice prior to the evaluation tasks, their performance in the new environment might have improved further.

Meta-task support

Because the task scope was relatively small for performing single atomic actions, I anticipated observing user behaviours typical of classes such as C1 and C6, as listed in Section 3.2.3. I associated these classes with three primary meta-tasks in the standard environment: "Determine possible next action(s)" (Meta-Task M1), "Translate a conceptual goal into a goal within the MMA systems" (Meta-Task M2), and "Understand and know common and specific features of MMA systems (Meta-Task M9) (e.g., participants need to figure out how to cut a clip in the MMA system, which is typically done by using a "razor tool" or using the menu item "Razor at the current at current time indicator"). The participants did have difficulty with these meta-tasks, which is consistent with the previous findings in User Study 1.

The results show that the new environment helped the users in these situations by allowing direct execution of actions from the ITV, especially for Group 1 actions (i.e., the actions for which the users made more failed attempts in the standard environment). The contextual views encapsulated action commands (e.g., cut a clip) with subtasks (e.g., "Create cuts with talkers") so that the users were able to locate and execute actions without prior knowledge of how it could be done with the standard UI, thus reducing the difficulty of the above meta-tasks (M2 and M9). In the new environment, the participants had to first "determine a relevant contextual group" for each action, and "traverse abstraction levels" in order to reach the target action command. These behaviours imply the involvement of the meta-task M5 ("Determine and traverse contextual/abstraction levels"). Table 5.9 summarises the above findings from Eval.1 in regards to the key aspects that were evaluated.

Table 5.9: The result summary of Eval.1 in regards to the key aspects evaluated.

(a) Task performance	(b) Ease of use	(c) Meta-task support
The new environment allowed significantly faster and more consistent performance time for the Group I actions (i.e., Actions A, C, E, and H, i which the participants had more errors when using the standard UI) than the standard environment.	Overall, participants found the new environment easier for performing the tested tasks than the standard environment.	Participants performed Group I actions significantly faster in the new environment by "determining a relevant contextual group" for each action and "traversing abstraction levels" in order to reach the target action command (meta-task M5). The interactive contextual views provided the support to reduce the difficulty of the meta-task that users had difficulty with in the standard environment (i.e., M2 and M9).

5.2.3.2 Evaluation 2: Executing a Sequence of Actions

In Eval.2, I evaluated the ITV support for tasks with slightly broader scope than those in Eval.1 (i.e., tasks involving multiple actions such as performing an operation with a sequence of actions and recognizing action dependencies in contrast to execution of individual actions in Eval.1) by analysing user performance for identifying and executing a sequence of relevant action commands. When a user has difficulty identifying a sequence of actions to complete a task, they typically seek external resources such as instructional videos and online help [128]. I investigated the effect of the operation previews used as an alternate help resource to identify a sequence of actions.

Procedure

In this evaluation, the participants attempted to complete a set of four general operations listed in Table 5.10 using Premiere Pro's UI. These were common operations seen during previous observations of novice users and they required multiple steps to complete (in contrast to individual action commands in Eval. 1).

Table 5.10: A list of operations that each participant was asked to perform.

A) Change the opacity of the opening title video clip (called "Title 0") to be 80% of its original (i.e., slightly dimmed).
B) Add a new audio clip for the ending scene by recording your own narration.
C) Change the speed of the video clip named 'I', which is in the ending scene of the project, to be 50% of its original speed.
D) Decrease the volume of an audio clip named "laughter-1" (the audience laughing and applauding), which is in the ending scene, down to approximately -3db.

For each operation, the participants were also asked to find help resource necessary to complete the operation in two conditions:

1. Using the operation previews (described in detail in Section 5.1.2, and shown in Figure 5.13)

2. Using help resources available on the Internet (such as instructional videos on Adobe's own website and tutorial materials).

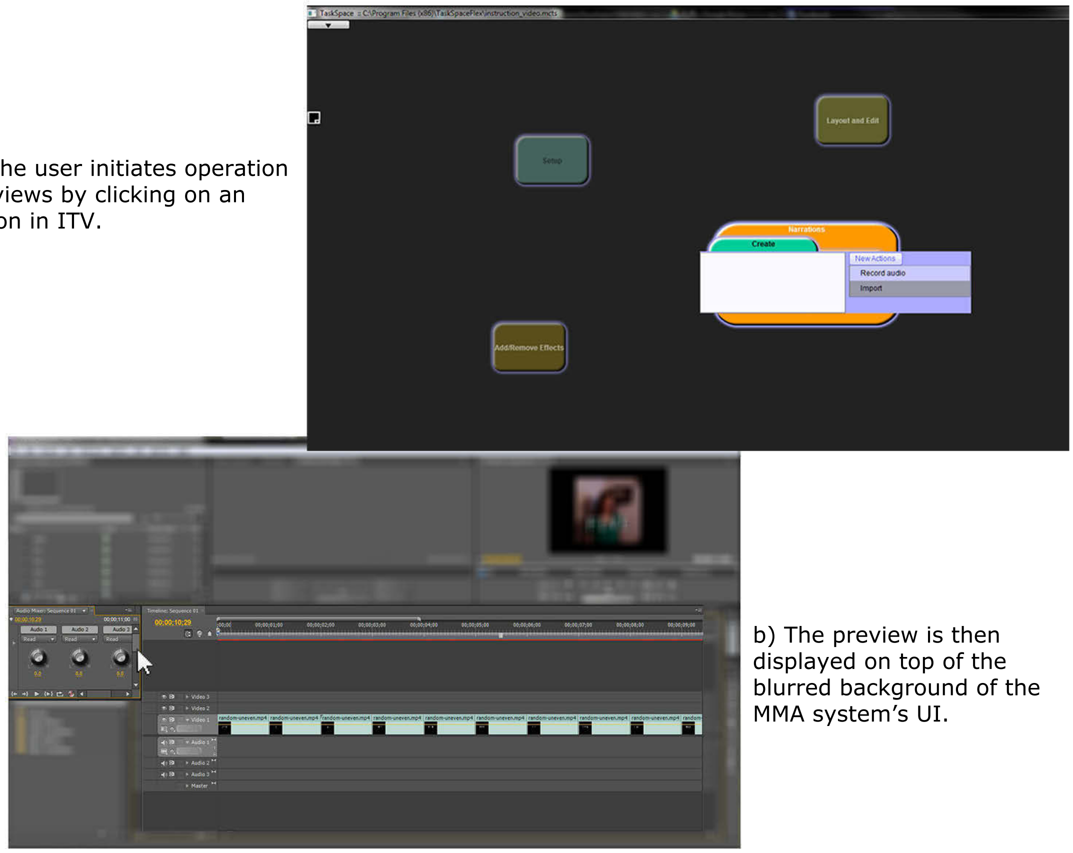


Figure 5.13: A preview of an action is in play. (a) A user initiates an operation preview by clicking on one of the possible actions in the ITV (shown at the top), and (b) the preview is then displayed on top of the MMA system's UI (shown at the bottom).

For each operation, there were two distinct cases that needed to be considered:

Case 1) A participant was able to complete an operation without any help resources (i.e., help was optional)

Case 2) A participant needed help resources to complete an operation (i.e., help was needed)

For Case 1), I first measured the duration of time to complete the operation. While help was not needed in this case, nevertheless, the participant was still asked to locate help resources in the two conditions described above. The duration of time to find

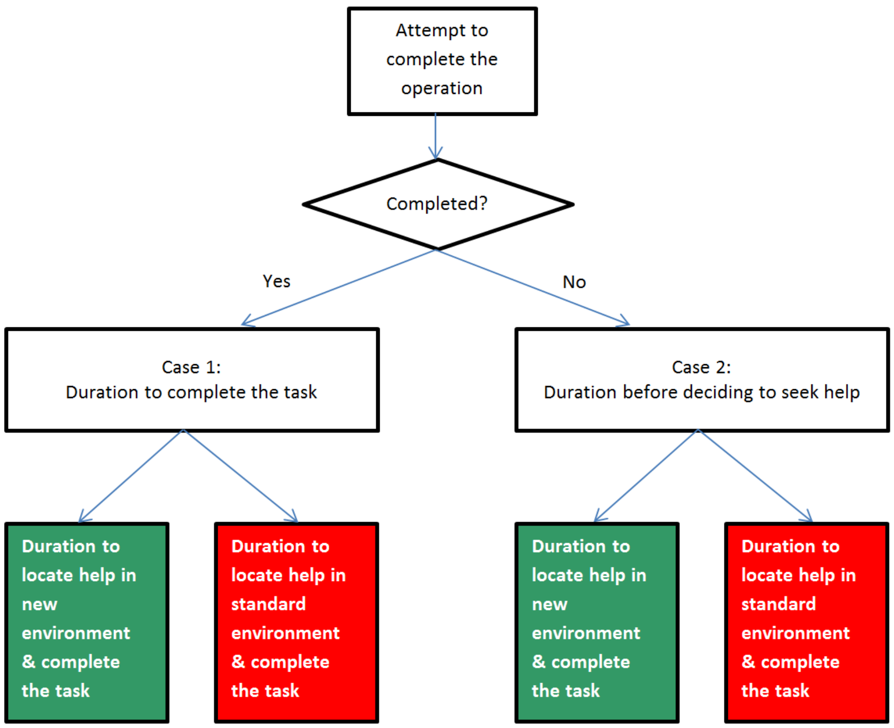


Figure 5.14: Durations of time measured in each of two distinct cases observed in the evaluation tasks: Case 1) a participant was able to complete an operation without any help resources (i.e., help was optional), and Case 2) a participant needed help resources to complete an operation (i.e., help was needed). These steps are repeated for each of the four operations.

the corresponding help and complete actions in each condition was then measured. For Case 2), I first measured the duration of time before the participant decided to seek help. Then, as in Case 1), the duration of time to find the corresponding help and complete actions was measured for each condition. Figure 5.14 summarises these cases and the durations of time measured in each case. The participants also rated the ease of each method on 10-point Likert scales (very difficult to very easy).

Ordering effects were minimized by randomizing the order of operations with a Latin square and by having half of the participants start the task in the new environment with operation previews while the other half started them in the standard environment. In addition to the time measurement mentioned above, I also observed general task performance behaviour for qualitative analysis.

As in Eval.1 (on p.159), I analysed the results with the three primary research questions in mind: task performance, ease of use, and meta-task support.

Note that while task completion time could be one of the indicators of the effect of this support, for this evaluation, I was particularly interested in how the users would utilize the operation previews in this new environment, and how it would (or would not) affect involvement of certain meta-tasks.

Results and Discussions

Task performance

Table 5.11 the task-by-task comparisons of the performance-time for locating help resources and executing operations in the two conditions: the new environment allowed participants to identify a necessary sequence of actions to complete all the operations significantly faster than locating and using help resources on the Internet. The qualitative observation, however, provides insights about user behaviours that are of much greater interest than this statistical difference implies, as described below.

All of the sixteen participants completed at least one of the tasks without any

Table 5.11: Mean performance-time (seconds) for locating help resources and executing operations using the new environment's operation previews vs. the standard environment. The p-values were calculated using Wilcoxon signed-rank test. The participants performed all the tasks faster in the new environment.

	Average time to locate help and execute the operation (in seconds)		
Action performed	Operation previews	Standard methods	p-value (bold when $p < 0.05$)
A) Change the opacity of a video clip	22.89	133.22	**<.001**
B) Decrease volume of an audio clip	20.15	95.49	**<.001**
C) Record voice	30.05	154.10	**<.001**
D) Change the speed of a video clip	19.40	86.60	**<.001**

help resources, and one participant completed all four tasks without assistance. This was unsurprising as the tasks were not equal in difficulty, and this is clearly shown in the results: the task completion rates without help resources (i.e., Case 1) were, Task A:62.50%, Task B:75.00%, Task C:18.75%, Task D:87.50%, and Overall:60.94%.

Table 5.12 and Figure 5.15 show the durations of time in the two cases. The results show that even when successful (Case 1), participants struggled to find ways to perform some operations (mean time duration to complete actions without any help resources was 77.50 seconds, indicated by the white bar for Case 1 in Figure 5.15). This result can be compared with the mean time duration invested for locating help resources and executing operations for the same tasks in the two conditions (i.e., the second column of Table 5.12): with the operation previews (18.25 seconds, Wilcoxon signed-rank test: $z = -2.18$, $p < .001$, indicated by the green bar for Case 1) and in the standard environment (104.88 seconds, Wilcoxon signed-rank test: $z = -4.24$,

Table 5.12: Durations of time (in seconds) for locating help resources and executing operations when tasks were completed successfully (Case 1), and when tasks were not completed without help (Case 2). For example, in Case 1 (in the second column), after successfully completed tasks (77.50 seconds on average indicated by the white bar in Figure 5.15), the participants sought help resources in 2 conditions: in the new environment (18.25 seconds on average indicated by the green bar) and in the standard environment (104.88 seconds on average indicated by the red bar). In both the cases, a user could improve the task performance by consulting the operation previews as the first choice of method to learn the appropriate sequence of actions regardless of the task difficulty.

	Case 1	Case 2
(Attempt to) Perform an operation	77.50 (duration to complete the task)	98.89 (duration before deciding to seek help)
Condition 1 (in the new environment)	18.25 (locate & execute)	29.74 (locate & execute)
Condition 2 (in the standard environment)	104.88 (locate & execute)	141.71 (locate & execute)

$p < .001$, indicated by the red bar). Thus, had the ITV preview feature been used before they attempted to figure it out on their own, they could have saved some time on these tasks, and, in the standard environment, they would have spent more time locating help resources than actually figuring out how to perform the tasks on their own.

Further, in Case 2), the mean time duration before they decided to seek help resources (98.89 seconds in indicated by the white bar for Case 2 in Figure 5.15) was larger than the time duration to locate necessary help resources and executing operations in the new environment (29.74 seconds indicated by the green bar) (Wilcoxon signed-rank test: $z = -3.73$, $p < .001$). This was in contrast to the duration of time in the standard environment: the participants took approximately as much time to locate help resources (141.71 seconds indicated by the red bar) as the time before

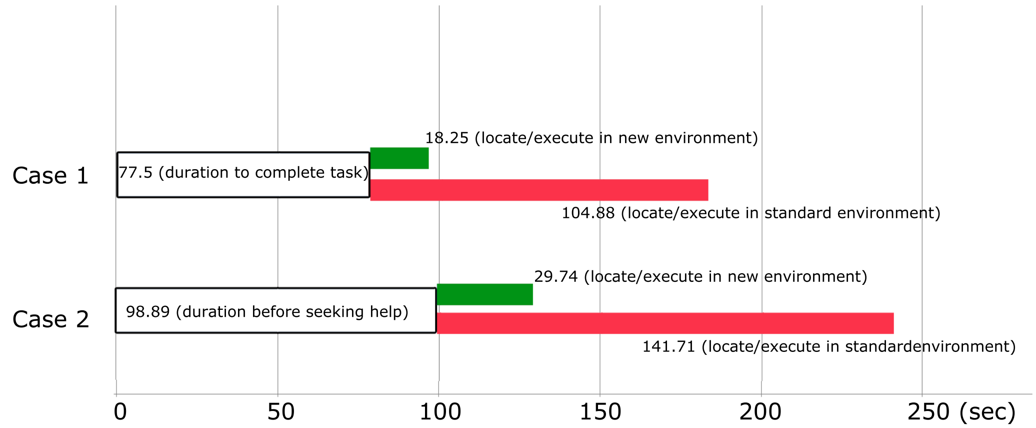


Figure 5.15: Comparisons of mean performance-time (in seconds) for locating help resources and executing operations when tasks were completed successfully (Case 1), and when tasks were not completed without help (Case 2). In the graph, the white bars indicate the duration of time to complete the task (for Case 1) and the duration of time before seeking help (Case 2), and the green (new environment) and red (standard environment) bars indicate the durations of time to locate and execute actions in the respective conditions. Even successful (Case 1), participants spent more time completing tasks on their own than locating and executing actions in the new environment.

they decided to seek assistance (Wilcoxon signed-rank test: $z = -1.52$, $p = .13$), and they spent a longer time locating help resources for incomplete tasks in the standard environment than the new environment (Wilcoxon signed-rank test: $z = -4.37$, $p < .001$).

These results imply that, in both Cases 1 and 2, a user could improve this task performance by consulting the operation previews as the first choice of method to learn the appropriate sequence of actions regardless of the task difficulty: in Case 1, on average, it took 18.25 seconds to locate help with the operation previews and complete the task, but it took 77.5 seconds to complete the task without any assistance or 104.88 seconds with the standard method to locate help and complete the operation. In Case 2, 29.74 seconds to locate help with the operation previews and complete the task in contrast to 141.71 seconds. in the standard environment.

There are a few possible reasons that participants took a longer time for locating help resources in the standard environment: many participants used ineffective search keywords that resulted in lists of unrelated help files or instructional videos. For example, when attempting to record an audio clip, five of sixteen participants did not use the word “record” in their first attempts of the search. The omission of distinct keywords such as “record” but the inclusion of other generic phrases such as “add a new clip” naturally resulted in a list of help resources mostly on how to import media files but not on how to record. As well, even after they located a possible help resource, it was still difficult for them to actually locate the relevant part(s) within the resource: for example, because videos often contained unrelated topics and participants did not want to watch the full length of the video, they needed to fast-forward and rewind it trying to find the segment of the video that was relevant to the task.

Ease of use

The participants found the method with the operation previews easier (the new environment’s mean rating=9.31/10, the standard environment’s mean rating=5.31/10, Wilcoxon signed-rank test: $z = -3.52$, $p < .001$). This is unsurprising as the new environment consistently allowed faster task completion time.

Meta-task support

All of the tested tasks showed the implicit meta-task of “determining a sequence of next actions” (Meta-Task M1) by first “parsing larger goals into viable subtasks (Meta-Task M3) and then “translate a high-level goal into a goal within the MMA system (Meta-Task M2). For example, to record an audio clip, participants first needed to break it down to a few steps of actions (e.g., select a track, arm the track, specify a location, start recording, and stop recording), and then they needed to figure out how to perform each of these steps (e.g., click on a target track to select it). Both the operation previews and instructional videos/website successfully helped users in this regard; however, as the results indicate, the new environment allowed faster task completion time than the standard environment. Locating relevant help resources often requires users to be able to first understand or know specialized terminologies in order to come up with effective keywords. It also requires them to switch between different types of high-level tasks, such as searching and consulting instructions, and actually performing operations in MMA system. The new environment allowed users to first narrow the search space of help resources by using the contextual views, and then initiate operation previews within a relevant view.

Two participants indicated that they knew how to perform certain operations in other programs (e.g., change opacity in Adobe Photoshop and change volume in Windows Movie Maker). Another participant dragged an icon for a transition effect to the time-line, which works on some tools (e.g., Sony Vegas Pro), but not in Premiere Pro. In these cases, their past experience did not help them perform these operations in Premiere Pro and thus they were unable to "apply solutions from a familiar situation to a new situation" (Meta-Task M7). The new environment allowed users to execute actions from the contextual views without knowing methods or techniques that are specific to certain MMA systems.

Further, the ITV operation previews were displayed on the overlay layer superimposed on the actual interface of Premiere Pro itself, thus eliminating the extra step of properly locating corresponding controls in the interface after seeing a video or instructions. For example, the users needed to map between the workspace configuration used in an instructional video and the one used in the actual MMA system, in order to reproduce operations shown in the video. This implies that the new environment reduced the difficulty of the meta-task, "Understand and know common and specific features of MMA systems" (Meta-Task M9) by visualizing the connections between possible actions and corresponding features and UI controls of the MMA system by means of the operation previews.

Finally, although using help files and watching instructional videos were treated as comparative methods to the new environment's operation preview feature in this evaluation, there would be no restriction on combining these two methods. For example, instead of triggering previews as implemented in this prototype, it is possible to directly link corresponding help files and/or instructional videos. In other words, the ITV could be used not only to organize tasks into a more contextually meaningful structure, but it could also be used to reorganize the vast amount of help resources available for these tasks. Organizing help resources based on specific tasks has been found useful [70, 71], and integrating contextual-views with the task-centric help resource search could add further support as it allows users to perceive helps within a larger picture of a task and make connections between helps for related subtasks. One participant remarked that some instructional videos were longer but often more informative than the implemented previews, especially if there was enough time watching its entirety. Allowing users to decide which types of help resources they would like to use may be, therefore, a useful feature for future implementations; i.e., the ITV previews can be a natural "first approach" when seeking help.

Table 5.13 shows the summary of the results from Eval.2 in regards to the key aspects that were evaluated.

5.2.3.3 Evaluation 3: Understanding Task/Data Relationships

In Eval. 3, I investigated the ITV support for understanding the information on the relationships between data clips in the workspace and performed actions. I have observed (in User Study 1) that keeping track of these relationships was difficult: e.g., a user wanted to find out what types of colour correction was applied to which data, but the user could not remember due to a very large number of clips on the

Table 5.13: The result summary of Eval.2 in regards to the key aspects evaluated.

(a) Task performance	(b) Ease of use	(c) Meta-task support
The new environment allowed significantly faster performance time for the tested tasks.	The participants found the new environment easier to perform the tested tasks than the standard environment.	The new environment allowed faster task completion time for tasks involving the meta-tasksM1, M2, and M3 than the standard environment, and reduced the difficulty of the meta-tasks such as M7 and M9.

time-line and there was no easy way to decipher this information in the standard MMA system.

Procedure

I compared task completion time as well as accuracy of participants task performance in the following two conditions for obtaining the information on the relationships between data clips in the workspace and performed actions:

1. using the new environment with the visualization of task/data relationships (described in detail in Section 5.1.3)
2. using the standard method of manually inspecting the data on which actions were performed.

The new environment provided a feature that highlighted the corresponding data clips in the workspace from the ITV. When a user selects an action or a group of actions in the relevant contextual view in the ITV, the relationship was indicated by highlighting corresponding data in the workspace while all other parts of the workspace were greyed out (Figure 5.16). In contrast, in order to obtain the information of these relationships in the standard environment, users inspected properties of each clip in a property window (as shown in Figure 5.17).

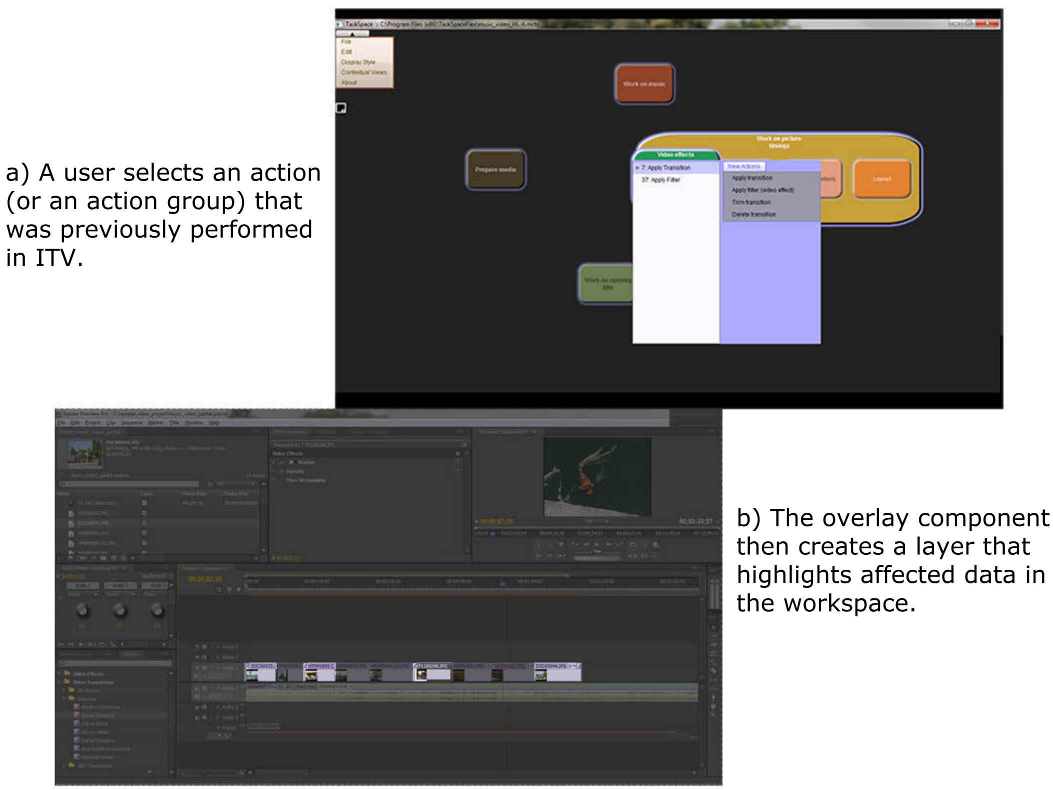


Figure 5.16: Premiere Pro with the highlighted workspace by the ITV, showing the affected data by a group of “Apply Transition” actions. Users interact with the ITV by selecting actions (or action groups) in order to visualize the corresponding data in the workspace.

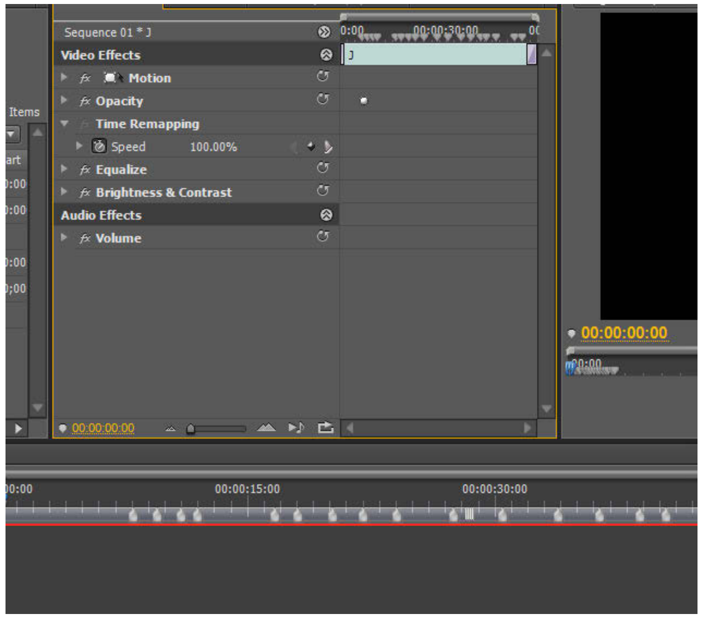


Figure 5.17: A window showing clip properties in the standard environment. Users manually investigate these properties of individual clips to understand what actions were performed on them.

In each condition, the participants answered a set of four questions (shown in Table 5.14) regarding the operations and their corresponding data in the workspace. The chosen questions were based on actual tasks with which users had had issues during the observations in User Study 1 (Section 3.1).

Ordering effects were minimized by randomizing the order of questions with a Latin square and by having half of the participants start the task in the new environment with the highlighting overlay while the other half started in the standard environment. I measured time to answer the questions and the correct answer rates, and I observed general task performance behaviour for qualitative analysis. The participants also rated the ease of each method on 10-point Likert scales (very difficult to very easy).

As in the previous two evaluations (Eval.1 and Eval.2), I analysed the results with the three primary research questions in mind: task performance, ease of use, and meta-task support.

Table 5.14: A list of questions that each participant was asked to answer.

A) Which of the following clips have been modified in brightness levels? (please circle all that apply)
B) Which of the following actions were performed on the clip called "B" ? (please circle all that apply)
C) Which of the following clips have been modified in scale (output video size)? (please circle all that apply)
D) Which of the following actions were performed on the opening title called "Title 01." (circle all that apply)

In addition, the evaluation assumed that the user did not know about the data in advance (i.e., they did not previously perform any actions on those data). This assumption was intended to approximate the condition in which users do not remember the relationships of data and performed actions.

Results and Discussions

Task performance

The results (shown in Table 5.15) show that the participants completed all the given tasks faster in the new environment; mean performance time with visualization=24.79 sec/task, mean performance time with standard method=54.70 sec/task, Wilcoxon signed-rank test: $z = -5.49$, $p < .001$. The correct answer rates were 99.17% (with visualization) and 97.92% (with standard method) and there was no statistically significant difference between the conditions (Wilcoxon signed-rank test: $z = -1.34$, $p = .07$).

Because there was no alternate way to easily provide the same information in

Table 5.15: Mean performance-time (in seconds) for completing operation with the visualization of relationships between task and affected data vs. with the standard methods. p-values were calculated using Wilcoxon signed-rank test. Participants performed all the tasks faster in the new environment.

Questions answered	With visualization (in seconds)	Standard methods (in seconds)	p-value (bold when $p < 0.05$)
A)	26.37	55.11	**<0.001**
B)	26.38	54.50	**<0.001**
C)	23.68	55.58	**<0.001**
D)	23.49	53.10	**<0.001**

the standard environment, the participants were expected to complete tasks faster in the new environment. The new visualization feature was able to indicate data affected not only by a single action, but also by a group of actions. For example, when a user selected the subtask "change opacity" (which grouped several actions to change opacity for individual data clips), the ITV highlighted all the data clips at once that were modified by this subtask, thus indicating a grouping of data based on the corresponding contextual group of actions. This type of aggregated information is not available in the standard environment.

Ease of use

The participants found the new environment easier to use for answering the questions (mean rating for visualization=9.46/10, mean rating for standard method=5.00/10, Wilcoxon signed-rank test: $z = -3.30$, $p < .001$).

Meta-task support

This feature provided support primarily for the meta-task "Maintain contextual meaning of the data clips in relation to performed tasks" (Meta-Task M8), and thus I have observed user behaviours predominantly indicating this meta-task. I observed novice users having difficulty with this meta-task during User Study 1 (e.g., users could not remember which data clips that she needed to work on after a task interruption). No other meta-tasks were indicated by any of the observed user behaviours in this evaluation. This was unsurprising as the participants were shown how to inspect data properties prior to the tested tasks, and this eliminated the necessity of many meta-tasks to determine how to accomplish them.

One issue brought up by a participant was that the visualization can be distracting because this information is not always necessary throughout a project, thus, in future implementations, this issue should be addressed accordingly.

Table 5.16 shows the summary of the results from Eval.3 in regards to the key aspects that were evaluated.

Table 5.16: The result summary of Eval.3 in regards to the key aspects evaluated.

(a) Task performance	(b) Ease of use	(c) Meta-task support
The new environment allowed significantly faster performance time for the tested tasks.	The participants found the new environment easier to use for obtaining the information on the relationships between data clips in the workspace and performed actions than the standard environment.	This feature provided support primarily for the meta-task "Maintain contextual meaning of the data clips in relation to performed tasks" (Meta-Task M8), and thus I have observed user behaviours predominantly indicating this meta-task.

5.2.4 Part II: A Case-Study

In order to evaluate the new environment in approximate real-world situations, I utilized a case-study approach to observe a novice user who completed an MMA project in the proposed environment with the ITV support.

5.2.4.1 Evaluation 4: Using the New MMA Environment in General Authoring Tasks

Participant

One of the participants who took part in the first three evaluations (Eval.1 to Eval.3) was asked to participate in this case-study session. This participant had minimal experience with Windows Movie Maker (less than 10 times) and had never used Adobe Premiere Pro prior to the study sessions. She completed tasks in Eval.1 to Eva.3 with near the mean performance time in the standard environment.[6] She agreed to take part in the additional study session, and was invited back ten days after the first session.

MMA project and ITV construction

The description of an MMA project (Project 2, shown below) was given at the beginning of the study session, and similarly to Project 1, the participant was shown the corresponding raw video and audio clips mentioned in the description as well as the completed sample output.

> **Project 2: Ordering and synchronization of video and audio clips**
>
> *There are 11 video clips that show people saying numbers (0, 1, 2, ..., 10). The corresponding audio was recorded separately on a sound recording*

[6]The participant's performance time for each of the tasks with contextual views: 37.76 sec (mean=46.31 sec), visualization of the relationships: 63.86 sec (mean=54.70), and operation previews: 122.15 sec (mean=119.27).

device, thus there are separate sound clips that need to be synchronized with the videos.

Your task is to edit and combine these clips to create a video so that when it's played back, it shows people saying numbers from the largest to the smallest (i.e., counting down). Each number should last for one second.

A corresponding ITV (shown in Figure 5.18) was constructed for this project, showing possible subtasks and actions associated with the subtasks. This ITV was built based on the processes performed for the same project by the author and another experienced user, and it contained generic subtasks (e.g., "Start new project" and "Collect media") that appear in many other projects.

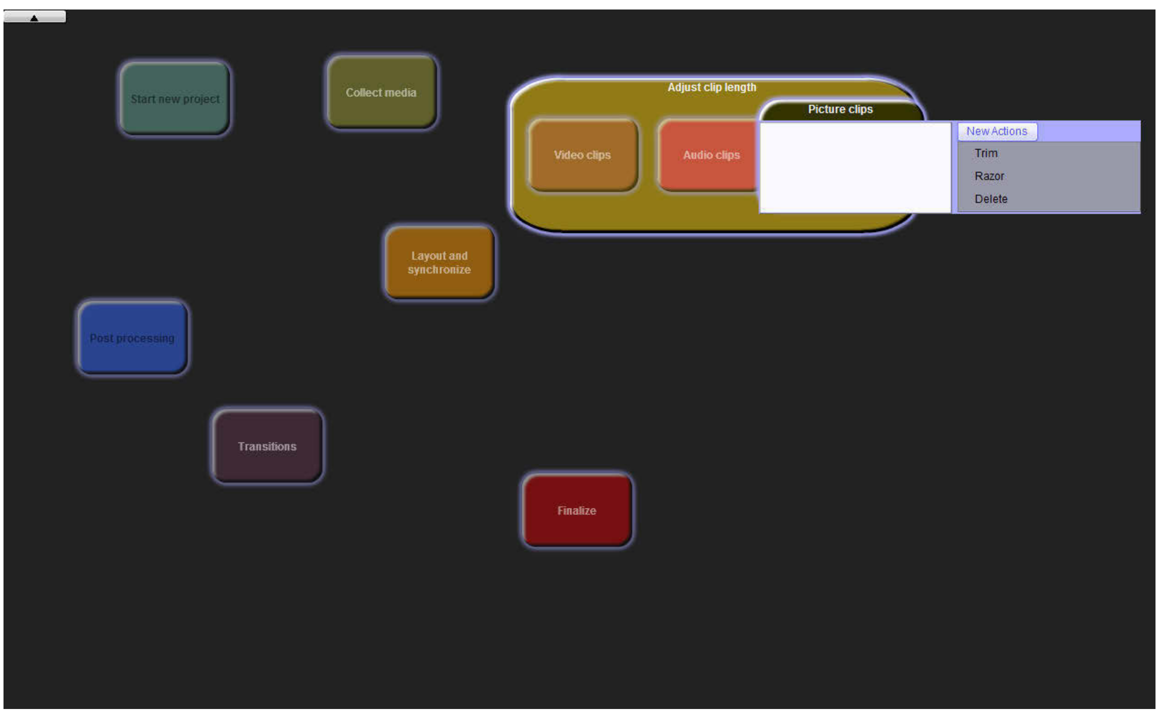


Figure 5.18: The constructed ITV for the project of ordering and synchronization of video and audio clips. This ITV was built based on the processes performed for the same project by the author and another experienced user, and it contained generic subtasks that appear in many other projects.

Procedure

The participant was first asked to read the project description, and then use the provided ITV to examine details of the project plan. After examining the task-view, the participant performed the rest of the tasks to complete the project in the new environment with the ITV features. The observational data as well as conversations between the participant and the experimenter were recorded and analysed with regard to the support provided by the environment.

Figure 5.19 is the schematic diagram describing the set-up of the main authoring tool and the corresponding ITV. The ITV provided an additional input channel through which the user controlled the authoring tool, as described in Section 4.1, and it displayed dynamically updated task-views with those newly performed actions. The ITV also provided the other features as described in Section 5.1. I employed the Wizard-of-Oz method [66] for dynamic update of the task-view for certain operations such as those that required direct manipulation of data (e.g., moving clips).

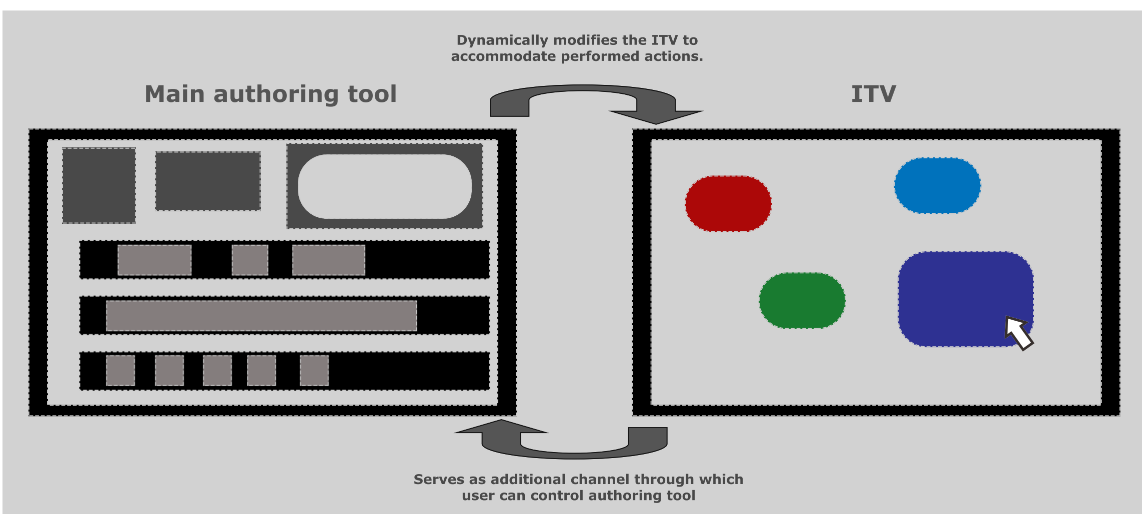


Figure 5.19: A schematic diagram of the MMA environment used in the case-study. The environment consisted of the ITV (shown on the right) and a secondary component which appears as a semi-transparent overlay superimposed on the main MMA system (shown on the left).

Table 5.17 shows the overall flow of the task that the participant followed.

Table 5.17: The overall flow of the task in the case-study. The constructed ITV naturally guided the participant to go through these basic steps. Note that I have grouped subtasks that the participants performed into these basic steps just for the organization purposes, and the user was not provided with this table: i.e., she used only the new MMA environment (i.e, Premiere Pro with the ITV support features) for performing these tasks.

Steps	Subtasks	Time spent (minutes)
1	Read project description & examine the ITV	5
2	Create new workspace, specify project properties, & import media	13
3	Ordering, trimming, & synchronization of clips	25
	(A short break)	10
(3)	(Continuation of ordering, trimming, & synchronization of clips)	14
4	Add transition effects & Render	13
	(Total accumulated time)	80

Results and Discussions

The participant spent about five minutes reading the project description and examining the provided ITV (Step 1 in Table 5.17). The ITV provided a relevant task-view for the project, providing necessary support for certain meta-tasks necessary in this first step of the project, such as “Generate overall task/work-flow” (Meta-Task M4) and “Parse larger goals into viable subtasks” (Meta-Task M3).

The participant then began the next set of subtasks of the project (Step 2 in Table 5.17), creating a new workspace, specifying project properties, and importing the necessary data clips). When the user encountered a problem performing specific tasks (e.g., changing the project video properties), she used the operation preview feature to see how they could be done. This feature allowed her to perform several meta-tasks such as “Determine possible next actions” (Meta-Task M1) (e.g., open project property window) and then “Translate a conceptual goal into a goal within the MMA system" (Meta-Task M2) (e.g., by opening the "Project" menu and then go to the submenu "Project Settings"). This example interaction is shown in Figure 5.20.

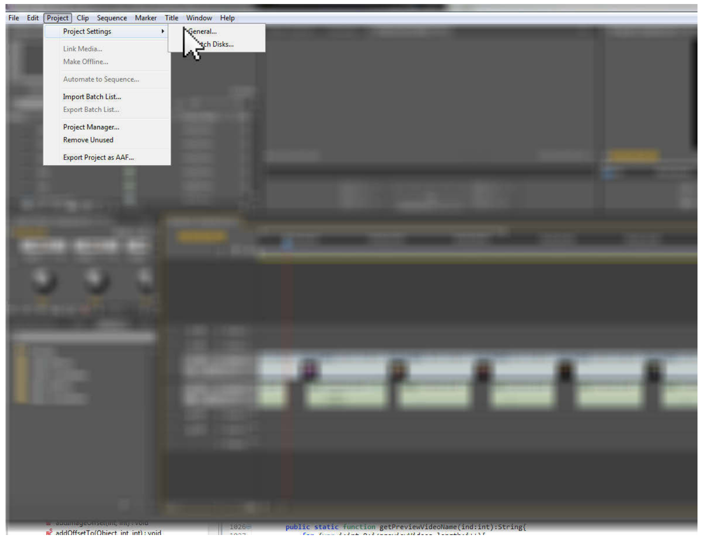


Figure 5.20: The operation preview for changing the project properties is shown on top of the blurred layer. This feature provides support for multiple meta-tasks such as M1, M2, M3, and M9 as shown in Eval.3 (Section 5.2.3.2).

The participant then proceeded to the next set of subtasks of the project (Step 3 in Table 5.17, which were ordering, trimming, and synchronization of the clips). These subtasks indicated some order constraints: for example, video clips needed to be placed in the time-line before the user could synchronize them with audio clips. Thus, if the focus was at a level of subtasks (i.e., placing clips and synchronization), then the order constraint for these subtasks would be (1) place (all) the clips in the time-line and then (2) synchronize. At a lower level of abstraction (i.e., the user deals with individual clips), however, the order constraint only applies to those clips that are involved (e.g., *video-clip1.mp4* and *audio-clip1.mp3* need to be placed in the time-line first before they can be synchronized, but no other clips need to be placed prior to the synchronization of these particular clips) (see Section 3.2.2 for more detailed discussions of constrains). This participant decided to first place all the necessary clips before synchronizing them (i.e., satisfying the higher-level order constraint).

The contextual views of the new environment helped the participant to move on to the next operation(s) (Meta-Task M1): the user was able to execute proper sequences of actions to complete the subtasks by interacting with the ITV to traverse different abstraction levels (Meta-Task M5) and to browse task-decompositions (Meta-Task M3). At the same time, there was no obvious indication of feeling stuck after completing one particular task goal: for example, a wizard-based approach could also guide users through several steps to accomplish a certain task goal, but it usually does not provide follow-up support to view the task process that users went through, or to modify automatically generated media.

The user spent about 25 minutes for Step 3 before the experimenter asked her to have a short break (approximately ten minutes). This break allowed the participant to rest from the unfamiliar work, and simulated a common situation in which the user would need to deal with task interruptions. During the break, the experimenter and the participant were engaged in conversation of random topics in order to limit opportunity for reflection on the previous work.

When resuming the task, the ITV also provided relevant support: instead of trying to recall the task without assistance, the very first thing the user did was to consult the ITV to see the current state of the task in relation to the planned task (saying "Let's see what's left to do"). While she would likely have eventually recalled what needed to be done next without the support of the ITV due to the short duration of the break, the ITV provides support for recalling or reviewing tasks without entirely recalling them from memory. This indicates a type of support to "Maintain overall task/work-flow and be aware of the work-flow location in the task-view" (Meta-Task M6).

After spending another 14 minutes for the rest of Step 3, the user then moved on to the final step (Step 4 in Table 5.17, adding transition effects and rendering the video). The user initially used the operation preview to learn how to apply transition effects for the first set of the clips (i.e., dragging an icon to the target clips), and repeated this process with all the clips in the timeline. She finished the entire process with rendering the video.

This case-study allowed for further insights on how the new environment could be used and improved in a more realistic working condition. For example, the MMA projects used in these evaluations were relatively small and completed within a short period of time. If a project unfolds over a long period of time, the corresponding ITV will also become large and may not scale well. Investigating the usability of the new environment for a larger project is, therefore, one of possible directions for future work.

5.2.4.2 Summary of Meta-Task Support in Eval.1 to Eval.4

Table 5.18 summarises the meta-task performance supported by the new MMA environment in Eval.1 to Eval.4. The new environment provided support for these meta-tasks in various ways: for example, users *viewed* the operation previews that showed users how to "translate a conceptual goal into a goal within the MMA system" (M2), while the initial task-view provided by the ITV allowed the users to *bypass* the process of generating overall task/work-flow (M4) (indirect support). The new environment itself actually maintained "overall task/work-flow" (M6) and "contextual meaning of the data clips in relation to performed tasks" (M8). Thus, the users did not need to directly perform these meta-tasks, but instead, the users utilized the ITV to *retrieve* the maintained information.

I observed a variety of meta-tasks during User Study 3. The tested tasks in the first three evaluations (presented in Sections 5.2.3.1 to 5.2.3.3) were explicitly designed to invoke certain meta-tasks as already discussed in each of the corresponding sections. For this reason, one cannot interpret the number of observations of a particular meta-task in these evaluations as an indicator of its frequency in general MMA tasks.

Table 5.18: The summary of the meta-task performance supported by the new MMA environment in Eval.1 to Eval.4. The new environment provided support for meta-tasks in various ways.

	Meta-task performance
Eval.1	Participants performed Group I actions significantly faster in the new environment by "determining a relevant contextual group" for each action and "traversing abstraction levels" in order to reach the target action command (meta-task M5). The interactive contextual views provided the support to reduce the difficulty of the meta-task that users had trouble with in the standard environment (i.e., M2 and M9).
Eval.2	The new environment allowed faster task completion time for tasks involving the meta-tasks M1, M2, and M3 than the standard environment, and reduced the difficulty of the meta-tasks such as M7 and M9.
Eval.3	This feature provided support primarily for the meta-task "Maintain contextual meaning of the data clips in relation to performed tasks" (Meta-Task M8), and thus I have observed user behaviours predominantly indicating this meta-task.
Eval.4	The initial task-view provided support for meta-tasks that involve overall planning (M4) and task-decompositions (M3) by traversing different abstraction levels of the task-view (M5). During the task, the operation previews provided support for several meta-tasks such as M1, M2, M3, M7, and M9, which were consistent with the result from Eval.2. After a short task interruption, the user utilized the ITV to remind herself what next steps were, indicating the new environment 's support for meta-task M6.

5.2.5 Part III: A Qualitative Video Analysis

5.2.5.1 Summary and Discussions on Participants' Comments and Behaviours

I also observed other behaviours and received feedback relevant to evaluating the new environment. In this section, I will summarise these participants' comments and behaviours into several categories. I collected and annotated a total of 185 comments/behaviours from the sixteen participants who took part in Eval.1 to Eval.4. These collected data highlight the types of support provided in the new MMA environment, and indicate how the environment might be potentially improved by addressing new issues that were found through these studies.

Drawbacks of contextual groupings (9/16 participants)
As discussed in Eval.1 (Section 5.2.3.1), while there was no learning/ordering effect, nine participants scanned through contextual groupings of actions only at the beginning of their sessions, likely trying to understand how the actions were grouped (e.g., “"But I think it [should be] in layout but not there”). This scanning behaviour subsided after two to three evaluation tasks, and the new environment in the end yielded faster task performance overall (average 23.81 seconds/task) than the standard environment (average of 46.3 seconds/task, Wilcoxon signed-rank test: $z = -3.94$, $p < .001$).

The initial grouping was provided in these evaluations, thus the users needed to investigate until they became used to the grouping schemes and possibly to the interface itself. While the prototype of this environment allows manipulation of groupings, an informal user study that I conducted with two novice users prior to User Study 3 showed that they did not possess sufficient knowledge to create a task-view on their own. Future work addressing this issue, therefore, includes different ways to generate and provide task-views. For example, task-view sharing for common MMA projects can be a good starting point as approaches to share similar information has been shown effective (e.g., task-sets used in AdaptableGIMP [70, 71]) and it can be refined as collaborative processes by other users of this tool in the community. Modularising action groups (i.e., sub-trees of a task-view) that can be exported and imported among multiple projects is another approach that is worth investigating.

Comments/behaviours regarding complexity of UI (6/16 participants)

- "That's why I didn't want to learn this [Premiere Pro], it's just too much. Looking at all these small things."
- "It [the action command] should be there, but it's just too much to find it."
- When attempting to change the font of a text clip, five participants looked in the property dialogue of the clip, but they were immutable from this particular dialogue.
- Tried a "find" function but it only searches data and NOT menu items/tools
- A participant found sound effect icons, but could not figure out how to apply them to the clip in the time-line.
- "I can see why people get stressed by looking for things in this [Premiere Pro]."

These comments and behaviours indicate some of the novice issues which are consistent with the findings from User Study 1. The new environment addressed these issues in several ways: e.g., the contextual views provided faster ways to find action commands that were more difficult to do so in the standard environment, and the operation previews allowed users to view how they could use effect icons to apply them to clips.

Showing frustration or wanted to quit a task (6/16 participants)

In the standard environment:

- "I have no idea [. . .] yeah I have no idea."

- “Can I get the answer?”
- “I don’t know [. . .]” (and sighed)
- “Hm, that’s a good one. I have no idea.”
- “I don’t think I’ll find it.”
- Two participants became very quiet and appeared frustrated not being able to locate the help online.

I did not observe these types of comments/behaviours in the new environment even when they made failed attempts. As discussed in Eval.1 (Section 5.2.3.1), the participants made more failed attempts in the standard environment (average of 1.25 per each task) than in the new environment (average of 0.63), and this may have likely contributed to accumulated frustration.

Uncertain/unknown terms (6/16 participants)

- Two participants did not know “gain”: the right-click menu in the standard UI had a “gain” command but did not realize that it would change the volume of the clip
- Two participants asked what “opacity” meant
- One participant thought “clear” referred to delete a transition effect, but it was in fact for deleting the clip itself.
- One participant asked what transition effects were.

In the new environment, the contextual views encapsulated certain terms (e.g., “gain”) with more generic grouping (e.g., Audio actions), allowing users to access these action commands from the contextually meaningful perspective. This encapsulation, however, still did not help when users did not understand the context itself (e.g, when the user did not know what transition effects were).

Using weak/inefficient search keywords (5/16 participants)

Five participants did not include "record" when searching help resources to record an audio clip, but did include "add," resulting in obtaining the search results of topics unrelated to recording. The contextual views allowed a different style of search (e.g., task decompositions), and similarly to the case of encapsulating special terms above, the users did not need to come up with own search keywords.

Comments on the ITV UI/interactions (4/16 participants)

For the new environment, I received the following unsolicited feedback:

- "the best of this is it has so few things [...] it's so fast."
- "Ok, that's cool, that's much easier."
- "Maybe this one [the ITV] should be the one to do first." (after switching from the standard UI)
- "[...] I think I like this one better" (pointing at the ITV, which she used first)
- "[...] yeah, because it [the operation preview] gives me the visual effect on that part there."
- "This tool [the ITV] making it [the task] much easier."
- "I think I'd love to have this one [the ITV], [it]teach me so much, like a lot of things."

Thus, these comments positively support the results from Eval.1 to Eval.3 showing that the participants found the new environment easier to use for performing the given tasks.

5.3 Conclusion

This chapter first described the prototype of the new environment designed to support the guidelines developed in Chapter 3. I selected these features based on the results of the iterative design process described in Chapter 4. Table 5.19 shows the relationships between the implemented features and the corresponding design guidelines.

Table 5.19: The implemented ITV features and the corresponding design guidelines that the features were based on. As shown, the interactive contextual views are the core features of the ITV, addressing the majority of the guidelines. The operation previews and the visualization of task/data relationships complement the interactive contextual views by visualizing connections between the task-view and the corresponding workspace; operation previews mainly focus on the connection between new/potential actions and the MMA system's UI while visualization of task/data relationships focuses on the connection between previously performed actions and data in the workspace. They also together address several frequently observed novice-specific challenges (Challenge-3 to Challenge-6) that are not fully addressed by the contextual views.

ITV features	**Guidelines**						
	G1	G2	G3	G4	G5	G6	G7
Interactive contextual views	✓	✓	✓	✓	✓	✓	
Operation Previews	✓				✓		✓
Visualization of Task/Data Relationships	✓						

The chapter also presented a usability study (User Study 3, summary of the study design is shown in Table 5.20) that I conducted to answer RQ3 by evaluating key support aspects of the new MMA environment:

(a) **Task performance**: to see if and how the new environment would improve novice users' task performance over the standard environment.

(b) **Ease of use**: to see if novice users would find the methods provided by the new environment easier to perform tasks compared to those in the standard environment.

(c) **Meta-task support**: to explore relationships between the new environment and kinds of meta-tasks to address novice challenges.

In the first three evaluations (Eval.1 to Eval.3), I compared the new environment with the standard UI. The contextual views (Eval.1) improved the novice task completion time for the tasks that were difficult to perform with the standard UI. The results also indicated that the contextual views yielded consistent task performance time regardless of difficulty levels of the tested tasks.

The operation previews (Eval.2) also produced faster task performance than the standard methods of searching help resources, and they enabled smoother transitions from learning the operations to actually executing them within the MMA system. The likely reason was that the ITV displayed the previews on top of the actual UI and eliminated an extra step of mapping between different UI window configurations as often observed in the standard environment. I have also discussed the possible application of the contextual views for organizing the vast amount of help resources available for these tasks, and allowing users to decide which types of help resources to be used may be a useful feature for future implementations.

The visualization of task/data relationships (Eval.3) provided in the new environment was also more effective for discerning information necessary to understand which actions were performed on which data in the workspace. The visualization can indicate data affected not only by a single action, but also by a group of actions. This type of aggregated information is not available in the standard UI, and thus it provided a more efficient method.

I also employed a case-study approach to evaluate the new environment's support in approximate real-world situations (Eval.4). The ITV features were useful in this case study as well: the user utilized the contextual views to narrow the search space of actions, the operation previews to study a set of actions to complete operations, and the overall task-view for initial planning and reviewing the task after a short task interruption. These behaviours indicated the support of the new environment for performing certain meta-tasks.

A qualitative analysis of the participants comments and behaviours (Eval.5) complemented the results from Eval.1 to Eval.4, highlighted types of support provided in the new MMA environment, and indicated how the environment might be potentially improved by addressing new issues (e.g., uncertainty of provided contextual clues and the scalability of the ITV when MMA projects become large) that were found through these studies.

As this was a prototype system and it is still a new approach, there is still room for this environment to mature. Observed issues include inconsistency/invisibility of interaction methods, understanding the new concept and interface of the contextual views, and the unknown scalability of the ITV when a task-view becomes larger and more complex than the ones used in these evaluations. Further studies, analyses, and refinement will be required in order to address these issues.

In this research, I identified both low-level and high-level issues that novice users often experience, and the associated meta-tasks also at both levels (Chapter 3). Each of these issues and their associated meta-tasks are often interdependent, and it is highly impractical to separate individual issues and attempt to solve them separately. I have, therefore, taken a collective approach to address these collective issues.

Furthermore, note that measuring the effect of support tools of this kind is difficult. Generally, measuring performance with a system as complex as an MMA environment can itself be very difficult. Supposing, though, that this performance of support tools could somehow be measured, then within such a complex environment there could conceivably be a variety of difficulties that might mask the positive effects of, for example, meta-task support. Finally, even if the support tools are found to be measurably helpful, then it is still difficult to draw direct conclusions about the exact impact of the tool in supporting specific meta-tasks: the meta-tasks are by definition fairly abstract, and they are often not independent, so it is difficult to isolate their effects. This difficulty is further exacerbated by considering that the guidelines can be implemented in different ways, and any two different implementations may have different interactions with respect to the various meta-tasks that may be underlying any particular task.

These challenges, problematic though they may be, do not imply that one should not explore these questions. They simply mean that we must be cautious in the strength of the conclusions that we draw. Indeed, in the studies I have described in this chapter, it is clear that for the very well-defined and non-exploratory tasks that I included, the ITV clearly led to more effective performance, and novice users clearly preferred this environment over the standard one. These results also support a variety of possible future experiments; indeed, it would be worthwhile to test some of these effects more closely over a longer period of time with a single integrated prototype system used within the context of larger MMA projects.

Table 5.20: Study designs and the ITV support evaluated in User Study 3.

Study Design	**Evaluations (Sections)**	**Support for/in ...**
Part I: Within-subjects comparisons	Eval.1: Locating and Executing Basic Action Commands (Section 5.2.3.1)	tasks with a local scope
	Eval.2: Executing a Sequence of Actions (Section 5.2.3.2)	tasks with a slightly broader scope (e.g., performing a group of actions)
	Eval.3: Understanding Task/Data Relationships (Section 5.2.3.3)	understanding relationships between data clips and performed operations
Part II: A case-study	Eval.4: Using the New MMA Environment in General Authoring Tasks (Section 5.2.4.1)	general MMA, requiring maintenance of a task-view, retaining contextual meanings of tasks, and resuming tasks from an interruption.
Part II: Qualitative video analysis	Eval.5: Summary and Discussions on Participants' Comments and Behaviours (Section 5.2.5.1)	performing meta-tasks

Chapter 6

Conclusions

Users who are new to the domain of multimedia authoring (MMA) have significant difficulties when learning new authoring tasks and tools. These difficulties arise from a number of problems such as the overly complex user interface designs and users' unfamiliarity with the tools themselves. My research sought to address one element of such problems: novice users' inability to establish appropriate mental representation of an unfamiliar task. Generating and maintaining appropriate representation of a task is a necessary step toward successful task performance [63, 130]. My goal in this research was to investigate, propose, and evaluate ways to support these users by providing a relevant MMA environment that helps them to generate and maintain an appropriate representation of MMA tasks.

This concluding chapter first summarises the research and its contributions. It then addresses the limitations of the current research and provides subsequent directions for future work.

6.1 Research Summary and Contributions

In this section, I will describe the contributions that this research makes. Once again I organize them according to the original research questions (RQ):

RQ1: Can we better understand the problems that prevent novice users from performing MMA at expert levels?

(a) Are there common issues experienced by novices?

(b) If so, are there common challenges underlying these issues?

RQ2: How can we address these challenges?

(a) What are possible solutions to provide viable support?

(b) Which solutions are worthwhile implementing initially?

RQ3: Do these solutions help novice users?

(a) Do they help novices to complete MMA tasks faster?

(b) Do novices find the new approach easier to use?

(c) Does the new approach successfully address the identified challenges (in RQ1)?

I first conducted a preliminary user study (User Study 1 in Section 3.1), in which I identified issues experienced by novices that were different from those experienced by experts (addressing RQ1(a)).

Contribution 1 (RQ1(a)): Identified issues that can hinder novice users from performing essential MMA tasks.

The list below summarises the key findings from User Study 1:

1. Novice users often performed editing tasks without planning ahead, while advanced users were very articulate about what to do in relation to the overall goal of a large project, suggesting that novice users have a less concrete mental model of a task.
2. Standard MMA interactions for performing basic actions were not always clear to novice users, who often chose inefficient approaches to accomplish goals. Advanced users followed established procedures for common tasks and could generate new ones when necessary.
3. Novice users did not understand the purpose of many GUI windows and did not customize to improve the editing environment. Advanced users were able to utilize many features across multiple tools.

Based on an analysis of MMA and theory of general problem-solving, I hypothesised that the difficulties experienced by novices are related to *meta-tasks*. Meta-tasks are those tasks that users perform to construct, modify, and utilize an appropriate *task-view*.

Contribution 2 (RQ1(b)): Introduced and defined concepts of task-view and meta-tasks.

A task-view is a (full or partial) representation of the structure of a task (Section 3.2.2). It consists of subtasks and specifies hierarchical and sequential relationships of these subtasks. A task-view is a cognitive snapshot of the user's conceptual model of the task. This snapshot can adapt slowly over time as the user refines her goals, but it can also change quickly as the user switches focus to a different aspect of the overall task. I associate the following three properties (scope, abstraction levels, and constraints) with task-views:

- A *scope* of a task-view defines what portion of the task the user focuses on at a time: e.g., a full-scope may be useful for examining an overview structure of a project, while a partial or local scope is more efficient for focusing on details of the project.
- An *abstraction level* defines the degrees of detailed information that a task-view provides: e.g., a sequence of actions (e.g., copy, paste, and adjust volume) may be grouped into a more abstract but contextually meaningful subtask unit such as "Edit a video clip".
- The *constraints* property of the task-view defines precedences of subtasks within the task-view: e.g., certain low-level subtasks (e.g., import a video clip and place it on the time-line) must be performed before others (e.g., change the duration of the video clip), which in turn may imply that certain high-level tasks must happen before others (e.g., "Import media" and "Layout clips" before "Modify video properties").

Thus, for a task-view, there is an interaction between the abstraction level, the scope and the relevant constraints induced by the partial-order among the subtasks.

MMA users perform meta-tasks to (1) construct, (2) modify and (3) utilize an appropriate task-view (Section 3.2.3). Users have different levels of fluency with these meta-tasks: based on the results from User Study 1, it appears that advanced users seem to have no difficulties executing tasks that involve various meta-tasks, while this is far from the case for novice users. The list below shows the identified meta-tasks.

M1. Determine possible next actions (a sequence of actions/subtasks)

M2. Translate a conceptual goal into a goal within the MMA system

M3. Parse larger goals into viable subtasks (task-decomposition)

M4. Generate overall task/work-flow

M5. Determine/traverse contextual/abstraction levels and scope of tasks

M6. Maintain overall task/work-flow and be aware of where they are in the work-flow

M7. Recognize when and how to transfer and apply solution from a familiar situation to a new situation (as often observed in the bottom-up strategy)

M8. Maintain contextual meaning of the data clips in relation to performed tasks

M9. Understand and know common and specific features of MMA systems.

I then organized the observed issues from User Study 1 into classes of related behaviours. Through analyses of these behaviours and consideration with respect to meta-tasks, I hypothesised that there are several novice-specific challenges underlying the observed novice issues and that these challenges can be explained by considering a set of meta-tasks (addressing RQ1(b)), in Section 3.2.3).

Contribution 3 (RQ1(b)): Identified novice-specific challenges

The list below shows the identified novice-specific challenges.

Challenge-1) Identification of concrete task-flow.

Challenge-2) Maintaining a mental model of a task and retaining contextual meanings of tasks.

Challenge-3) Understanding relationships between data clips and performed operations.

Challenge-4) Recovery from error.

Challenge-5) Recognizing dependencies: Novice users had difficulties understanding what other actions needed to be performed prior to their target operation.

Challenge-6) Recognizing combinations of actions needed to accomplish a certain goal.

Challenge-7) The contextual information of the task is not available in current tools.

Challenge-8) Task interruption could leave users disoriented when they return to the system.

I then developed a set of relevant design guidelines. These guidelines aim at addressing the underlying challenges associated with meta-task difficulty, and thus indirectly address the original novice issues (addressing RQ2(a)), in Section 3.3).

Contribution 4 (RQ2(a)): Developed design guidelines for new MMA environments

The list below shows the design guidelines developed for new MMA environments.

G1. Provide additional contextual information where possible.

G2. Show task-decompositions and structures.

G3. Provide explicit task-views that can dynamically change to show different levels of abstraction, scopes of tasks, and task contexts.

G4. Provide common task-structures that allow integration of newly performed actions.

G5. Indicate possible next subtasks/actions.

G6. Provide a means to carry out actions within contextually meaningful views.

G7. Show sets of common operations.

Table 6.1 summarises the design guidelines that aim at addressing novice-specific challenges and associated meta-tasks. The relationships between the challenges and the meta-tasks are rather complex, and in order for a novice to overcome each challenge, more than one type of meta-tasks are typically involved.

Table 6.1: Design guidelines aimed at addressing novice-specific challenges (and associated meta-tasks).

Design Guidelines	Novice-specific challenges								Meta-tasks								
	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	m1	m2	m3	m4	m5	m6	m7	m8	m9
G1. Provide additional contextual information where possible.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
G2. Show task-decompositions and structures.	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
G3. Provide explicit task-views that can dynamically change to show different levels of abstraction, scopes of tasks, and task contexts. notes, while novices struggle).	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
G4. Provide common task-structures that allow integration of newly performed actions.	✓	✓		✓	✓	✓			✓	✓	✓	✓	✓	✓	✓		✓
G5. Indicate possible next subtasks/actions.				✓	✓	✓			✓	✓	✓		✓		✓		✓
G6. Provide a means to carry out actions within contextually meaningful views.		✓		✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
G7. Show sets of common operations.		✓		✓	✓	✓			✓	✓	✓		✓	✓	✓		✓

With these design guidelines, I followed an iterative approach to establish a framework for an MMA environment (Figure 6.1) that provides support for novice users (addressing RQ2(b), in Section 4.1). This approach involved a dynamic process of developing initial design ideas, evaluating them in low-fidelity implementations and design refinement (Sections 4.2 and 4.3).

> **Contribution 5 (RQ2(b)) Designed and prototyped a new MMA environment.**
> The new MMA environment consists of the original MMA system together with a support tool called the Interactive Task-Viewer (ITV). The ITV augments the standard environment by providing features that provide additional contextual information about subtasks and an additional set of possible interactions. These features allow novice users to view and explore the complex task structure of MMA processes more easily, and allowed effective utilization of an MMA system. Figure 6.1 shows the interaction/feedback cycle between the user, the MMA system, and the ITV.

I then implemented a prototype of this new MMA environment and evaluated it in User Study 3 (addressing RQ3(a) to (c), in Chapter 5). Table 6.2 shows the list of implemented ITV features, the corresponding design guidelines that the features were based on, and the novice-specific challenges that they were designed to address.

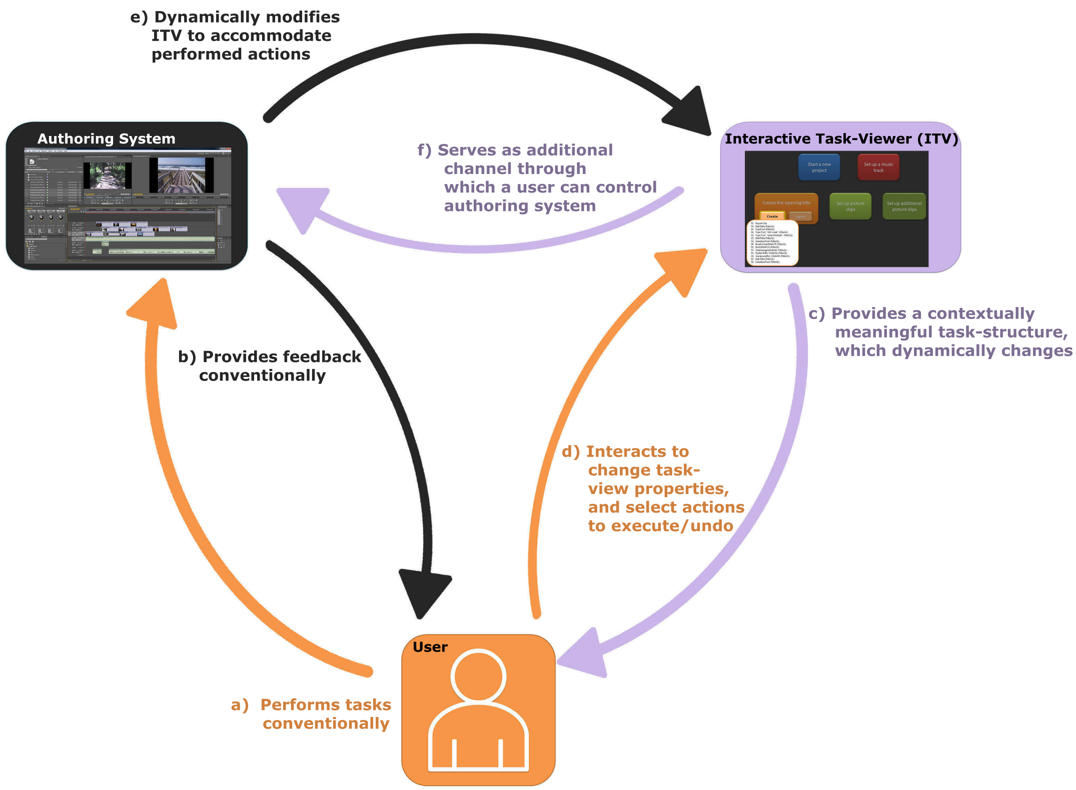


Figure 6.1: A framework for an MMA environment and its interaction/feedback cycle between users, the Interactive Task-Viewer (ITV), and the MMA system. The user executes actions (orange arrows) on both the MMA system and the ITV. The MMA system sends its output (black arrows) both to the user, in the form of visual feedback, and to the ITV, in the form of internal messages allowing the ITV to update its display. The ITV sends its output (magenta arrows) to both the user, in the form of visual feedback, and to the MMA system, in the form of internal messages that the MMA system can then execute as actions (e.g. open a file, trim the endpoint of a video, etc). Thus, each of the three elements—the user, the MMA system, the ITV—interacts in a bi-directional way with the other two.

Table 6.2: The implemented ITV features, the corresponding design guidelines that the features were based on, and the novice-specific challenges that they are designed to address.

ITV features	**Guidelines**							**Novice-specific challenges**							
	G1	G2	G3	G4	G5	G6	G7	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8
Interactive contextual views	✓	✓	✓	✓	✓	✓		✓	✓	✓[1]	✓			✓	✓
Operation Previews	✓				✓		✓				✓	✓	✓	✓	
Visualization of Task/Data Relationships	✓									✓				✓	

[1]Partially supported by utilizing the contextual view based on media-types.

Contribution 6 (RQ3): Presented evidence of relevant support provided by the new MMA environment.

Based on the results of User Study 3 (Chapter 5), I found that, with the support provided by the ITV, users were better able to perform common MMA tasks in this new environment than in the traditional environment. The support from the ITV was most effective for tasks that users had difficulty with in the traditional environment. Participants also found the new environment easier for performing the tasks than the standard environment.

A qualitative analysis of user behaviours showed that the new environment allowed users to perform tasks involving certain meta-tasks faster, implying that the new support positively addressed the meta-task difficulty of novice users for the tested tasks. This support was accomplished by providing functions that integrate task scope and abstraction levels in a manner that would otherwise generally not be feasible for users without the accumulated skills and experiences typical of more advanced users.

6.2 Research Limitations and Future Research Directions

6.2.1 Design of MMA support tools

Limitations

One unknown aspect of my approach is the scalability of the proposed method of support. The usability tests focused on relatively small MMA projects lasting for several minutes to a few hours. The sizes of these projects appeared appropriate for this group of users as these users had no extensive experience in the task domain, and they had not been able to complete longer projects without appropriate support (as observed in the preliminary studies). While tasks selected for these usability evaluations were designed to be as generic as possible, the applicability of the approach to other types of tasks that were not specifically tested in this work is also unknown.

As well, the depth of the prototype implementation used in the user studies was rather shallow: however, this was a necessary repercussion of testing several initial design ideas. The main purpose of this study was to suggest useful initial designs for helping novice MMA users, which can be more rigorously tested and analysed in future research. This is necessarily a multiple stage process, and the focus in the current dissertation was on finding initial working solutions.

Finally, I made a number of assumptions to develop a solution that addresses the issues experienced by novice users during MMA tasks: I assumed that there are common challenges underlying the observed novice issues and that these challenges can be explained in relation to meta-task difficulty. I then hypothesised that by developing design guidelines that aim at addressing these underlying challenges, I can indirectly address the original novice issues. While it is difficult to isolate specific effects of support provided by the ITV as discussed in Section 5.3, the results of the usability study showed that the new MMA environment designed based on these guidelines nonetheless allowed better utilization of the MMA system.

Future directions

Given the above limitations, it would be worthwhile to test the proposed MMA environment in a systematic way involving a larger MMA project with a larger set of various MMA tasks that would unfold over several days or weeks. These tests will naturally require a deeper implementation of ITV features. Potential challenges include handling issues when a task-view becomes large and complex, and maintaining focus and interest of inexperienced users for such a long period of time.

Another research direction is to investigate different approaches for providing constructed task-views. In this research, task-structures were determined and the

ITV was constructed based on actual tasks performed by experienced users for given MMA projects. As many MMA projects share common subtasks (as discussed in Section 2.1.2.2), one extension of the ITV may be to allow import and export of subtasks (or action groups such as "Layout clips" that contain actions only relevant to the particular group) to be shared among other users. Approaches to share similar information has been shown effective (e.g., task-sets used in AdaptableGIMP [70,71]) and it can be refined as collaborative processes by other users of this tool in the community. Users could then use these imported subtasks as building blocks to construct a new task-view for a new project. Adaptable interfaces that can be customisable by users allow better task-performance [42], and thus this extension may likely be an asset for further improving user performance of the proposed MMA environment.

6.2.2 Task domains of the proposed support

Limitations

As this research focused on MMA tasks of video editing and music/sound production, this restricts application of my findings to these tasks, and thus, generalisability of my approach to other task domains is limited.

Future directions

This limitation opens a possible future research direction to explore other task domains for which this type of meta-task support might be useful. For example, other than other creative domains such as photo editing and 3D animation (which also involve very complex tools), computer programming tasks share many characteristics with MMA tasks: both have problem-solving aspects, both often take a long period of time to complete a single project, both tasks are sometimes carried out as collaborative tasks, both use environments that have complex and often overwhelming interface issues, especially for first time users (i.e., many high-end IDEs for programming are complex to set up and utilize), and both involve cyclic refinement strategies and non-linear aspects of the tasks. The basic steps of the approach presented in this dissertation could be followed to address issues experienced by users in these task domains.

6.2.3 Types of MMA systems

Limitations

The primary focus of this research was to provide support for novices to utilize standard MMA systems, typically used by advanced users (e.g., Adobe Premiere Pro [2]). The findings indicate that novice users of these tools benefited from the support provided by the new MMA environment: however, to what extent the current research could improve novice task-performance when they use other types of MMA tools such as mobile apps or those tools with simplified UI designs (e.g., GarageBand [11] or Windows Movie Maker [89]) is still unknown.

Future directions

While the simplified tools hide or eliminate advanced features that are considered difficult for novice users to use, it would be interesting to see if the new environment could still improve the task performance with these simplified tools as well. That is, we could employ the ITV support to compliment the relevant support already in place for these simplified tools rather than as the alternate solution to the existing approaches for addressing the issues of complex interface designs.

6.2.4 Target user group

Limitations

I focused on supporting mainly novice users, and as a result, the developed guidelines address issues experienced by these novice users. While the novice participants who took part in the user studies were from various backgrounds (i.e., both in non-technology disciplines/professions and science/engineering), the application of the findings to general novice users such as young children and older adults is also limited.

Future directions

This limitation leads to a new research prospect: one may apply the approach developed in this study to different user groups performing MMA tasks. For example, various studies that were carried out to create an MMA environment for children [4,41,45] suggest some new and interesting possible research directions. Analysing children's problem-solving behaviours [72, 75] requires a shift in focus from the issues associated with adult novices, thus, one will likely come up with different guidelines than those presented in this dissertation. Furthermore, instead of generic MMA tasks, one may restrict their focus to a more specific type of MMA tasks such as music (only) productions, which are likely to have more specific work-flow patterns than generic MMA tasks. Because of this difference, features and relevant contexts that would likely help users for these tasks may be different as well.

In any case, one should develop new guidelines by following the essential steps introduced in the current research and evaluate them with appropriate UI design methodologies.

6.2.5 Application of the ITV features and techniques

Limitations

Even with the same set of design guidelines, one may come up with a different set of features as there is always a creative leap in going from design guidelines to actual feature designs. While the implemented features provided relevant support to improve novice MMA task performance in this dissertation, there may possibly be a different set of design approaches that could further improve novice task performance.

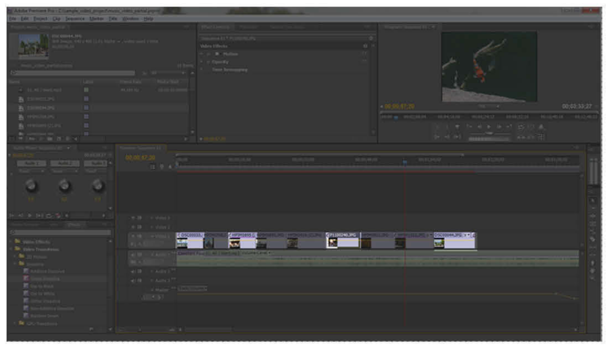
Future directions

Another direction of future work is, therefore, to explore different design features than those that were derived from the guidelines in this dissertation: comparing new design approaches against the current features could potentially lead to a broader set of design features for such support tools. I intend the design features introduced and evaluated in this dissertation to serve as the initial step toward this goal.

Another interesting branch of research to explore is the use of the overlay technique (Figures 6.2 and 6.3) described in Section 5.1. The overlay feature used in the prototype implementations allowed for part of the support tool to appear on top of an existing tool. When needed, the overlay component can also limit interactions with the authoring tool itself: some areas of the overlay are transparent and users can directly access the original MMA tool through these areas, while other areas can be semi-transparent so that the users can only access the prototype tool's UI.

Prototyping with the overlay technique can allow rapid UI evaluation in several situations: when obtaining or modifying the original source code of existing programs is implausible, when some restrictions on tool usage need to be reinforced during evaluation processes, and when a full-scale implementation or modification of original programs is unnecessary (e.g., testing only a handful design features).

Furthermore, as discussed in Section 5.2.3.2, the ITV with this overlay technique



(a) The view with the highlighter overlay as the user sees it

(b) The overlay creates a layer with transparent areas that expose relevant data and a semi-transparent area that greys out the rest of the rather cluttered workspace

(c) The exploded view of the overlay from the interface.

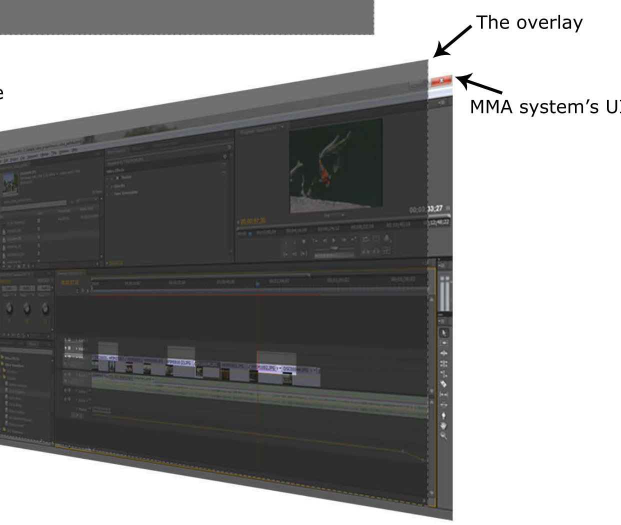


Figure 6.2: The highlighter overlay on an existing MMA tool. (a) The transparent areas highlight relevant data while the grey area masks the rest of the cluttered workspace. (b) The overlay creates a layer with transparent areas that expose relevant data and a semi-transparent area that greys out the rest of the rather cluttered workspace. (c) The image at the bottom shows the exploded view of the overlay from the interface. Note that the printed version of the image appears much darker than on screen; in practice, the greyed-out area is still clearly legible, but simply lightly greyed out.

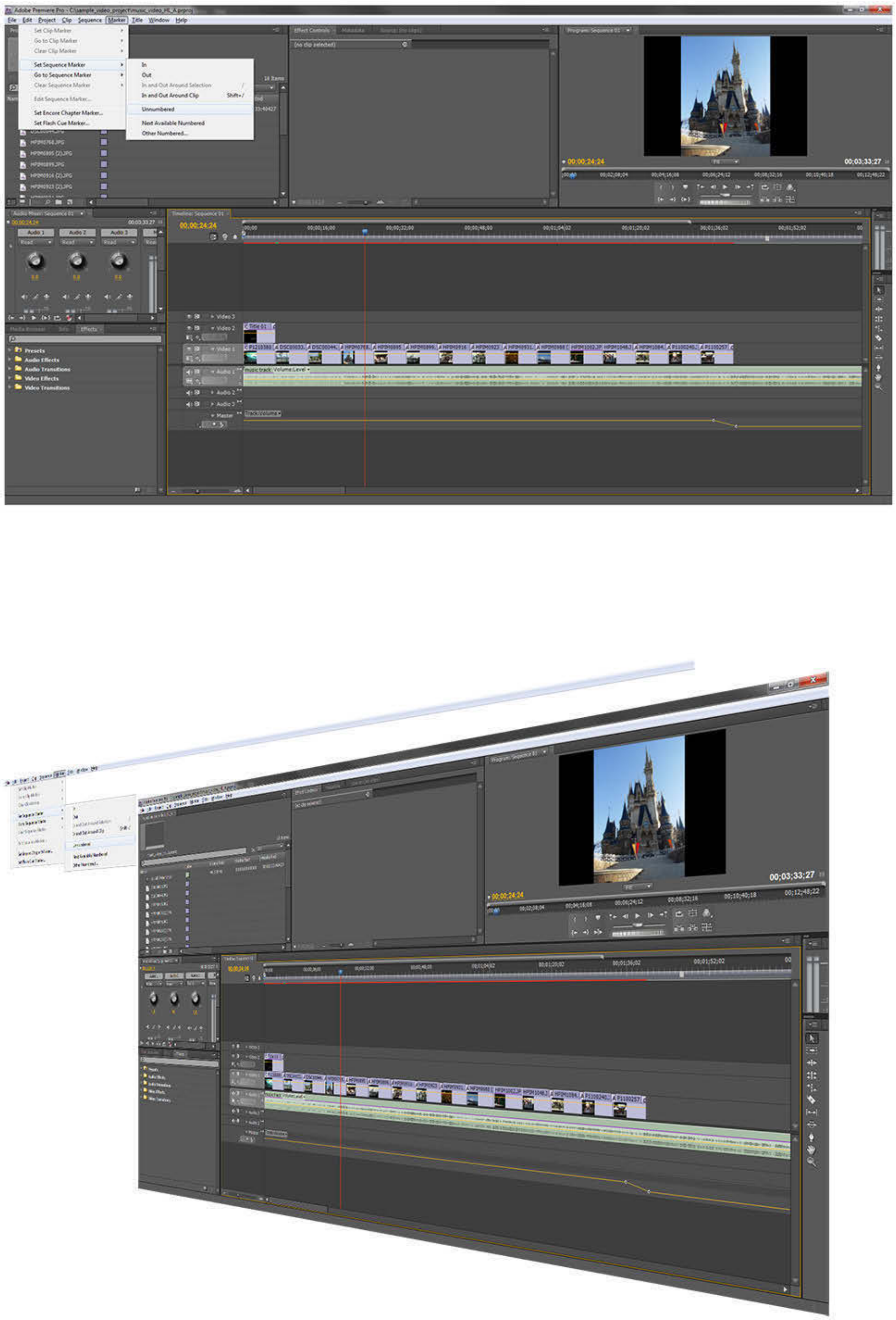


Figure 6.3: Menu overlay superimposed on the actual menus of the MMA system. The figure shows a static image of the interface as seen by the user (shown at the top), and the exploded view shows distinct layers (shown at the bottom). The menu overlay only simulates their behaviours (e.g., open/close menu) thus showing previews of action execution.

can provide a new way to organize vast amounts of available help resources and could allow users to locate relevant ones faster through the contextual views. The operation previews utilizing the overlay technique could also be used as a type of tutorial tool that displays relevant information on top of the MMA tool's UI without switching between tutorial videos and the actual UI (which was found difficult in User Study 3).

6.3 Conclusion

This dissertation presented guidelines for constructing new multimedia authoring environments. The new environment augments existing tools with the Interactive Task-Viewer in order to provide support for meta-tasks of novice users by incorporating additional contextual information of a task as well as new interactions. The results of the usability studies showed that the users found the new environment easier to use, and they performed common MMA tasks faster in the new environment than in the traditional environment.

The main purpose of this study was to suggest useful initial designs for helping novice MMA users. This opens possibilities for many new research directions and applications, including longitudinal testing with different user groups and refinement of such a tool for use in existing common applications.

Bibliography

[1] Ableton. Live. https://www.ableton.com/en/live/new-in-9/, April 2015.

[2] Adobe. Premiere. http://www.adobe.com/products/premiere.html, April 2015.

[3] Akiyama, Y., and Oore, S. A multimodal input device for music authoring for children. In *Advances in Multimedia Modeling, 14th International Multimedia Modeling Conference, MMM 2008* (Kyoto, Japan, January 2008), pp. 165–174.

[4] Akiyama, Y., and Oore, S. PlaceAndPlay: a digital tool for children to create and record music. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (Florence, Italy, 2008), ACM, pp. 735–738.

[5] Akiyama, Y., and Oore, S. Supporting problem-solving approaches in multimedia authoring task. In *Proceeding of the 2013 International Conference on Multimedia and Human-Computer Interaction, MHCI'13* (Toronto, Canada, July 2013), pp. 60.1–60.8.

[6] Akiyama, Y., and Oore, S. Designing tools to support multimedia authoring by incorporating problem-solving strategies. *Journal of Multimedia Theory and Application 2*, 1 (2014), 11–19.

[7] Akiyama, Y., Oore, S., and Watters, C. Framework for constructing task-space to support novice multimedia authoring. *Multimedia Tools and Applications 74*, 15 (July 2015), 6111–6147.

[8] Andrade, O. D., Bean, N., and Novick, D. G. The macro-structure of use of help. In *Proceedings of the 27th ACM International Conference on Design of Communication* (New York, NY, USA, 2009), SIGDOC '09, ACM, pp. 143–150.

[9] Anzai, Y. Cognitive control of real-time event-driven systems. *Cognitive Science 8* (1984), 221–254.

[10] Apple. Final Cut Pro. http://www.apple.com/final-cut-pro/, April 2015.

[11] Apple. GarageBand. http://www.apple.com/mac/garageband/, April 2015.

[12] Arndt, T., and Katz, E. Visual software tools for multimedia authoring. *J. Vis. Lang. Comput. 21*, 3 (June 2010), 184–191.

[13] AutoDesk. Maya. http://www.autodesk.com/products/autodesk-maya/overview, April 2014.

[14] Balbo, S., Ozkan, N., and Paris, C. Choosing the right task modelling notation: A taxonomy. In *The Handbook of Task Analysis for Human-Computer Interaction*. Lawrence Erlbaum Associates, 2004, pp. 445–465.

[15] Barrett, A., and Weld, D. S. Task-decomposition via plan parsing. In *National Conference on Artificial Intelligence* (1994), pp. 1117–1122.

[16] Berlage, T. A selective undo mechanism for graphical user interfaces based on command objects. *ACM Transactions on Computer-Human Interacttion 1*, 3 (Sept. 1994), 269–294.

[17] Blender.org. Blender. http://www.blender.org/, April 2015.

[18] Braun, T., Siegal, H., Beck, N., Boloni, L., Maheswaran, M., Reuther, A., Robertson, J., Theys, M., Yao, B., Hensgen, D., and Freund, R. A comparison study of static mapping heuristics for a class of meta-tasks on heterogeneous computing systems. In *Heterogeneous Computing Workshop, 1999. (HCW '99) Proceedings. Eighth* (1999), pp. 15–29.

[19] Bruno, A., Paternò, F., and Santoro, C. Supporting interactive workflow systems through graphical web interfaces and interactive simulators. In *Proceedings of the 4th International Workshop on Task Models and Diagrams* (New York, NY, USA, 2005), TAMODIA '05, ACM, pp. 63–70.

[20] Bull, R., Espy, K. A., and Senn, T. E. A comparison of performance on the Towers of London and Hanoi in young children. *Journal of Child Psychology and Psychiatry 45*, 4 (2004), 743–754.

[21] Bulterman, D. C. A., and Hardman, L. Structured multimedia authoring. *ACM Transactions on Multimedia Computing, Communications, and Applications 1* (February 2005), 89–109.

[22] Bunt, A., Conati, C., and Mcgrenere, J. What role can adaptive support play in an adaptable system. In *In Proc. of IUI, 2004* (2004), ACM Press, pp. 117–124.

[23] Bunt, A., Conati, C., and McGrenere, J. Supporting interface customization using a mixed-initiative approach. In *Proceedings of the 12th International Conference on Intelligent User Interfaces* (New York, NY, USA, 2007), IUI '07, ACM, pp. 92–101.

[24] Bunt, A., Dubois, P., Lafreniere, B., Terry, M. A., and Cormack, D. T. Taggedcomments: Promoting and integrating user comments in online application tutorials. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2014), CHI '14, ACM, pp. 4037–4046.

[25] Burne, M. D. Cognitive architecture. In *The Human-computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, J. A. Jacko and A. Sears, Eds. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 2003, ch. 5.

[26] Buxton, W. A. S. Chunking and phrasing and the design of human-computer dialogues. In *Human-computer Interaction*, R. M. Baecker, J. Grudin, W. A. S. Buxton, and S. Greenberg, Eds. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 494–499.

[27] Candy, L., and Hori, K. The digital muse: HCI in support of creativity: "Creativity and Cognition" comes of age: towards a new discipline. *Interactions 10*, 4 (July 2003), 44–54.

[28] Carroll, J. M. *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill.* MIT Press, Cambridge, MA, USA, 1990.

[29] Carroll, J. M., and Mazur, S. A. Lisalearning. *Computer 19*, 11 (Nov. 1986), 35–49.

[30] Carroll, J. M., and Rosson, M. B. Paradox of the active user. In *Interfacing Thought: Cognitive Aspects of Human-computer Interaction*, J. M. Carroll, Ed. MIT Press, Cambridge, MA, USA, 1987, pp. 80–111.

[31] Dancyger, K. *The Technique of Film and Video Editing: History, Theory, and Practice.* Taylor & Francis, 2011.

[32] Davies, S. P. Planning and problem solving in well-defined domains. In *The cognitive psychology of planning*, R. Morris and G. Ward, Eds., Current issues in thinking & reasoning. Psychology Press, 2005, ch. 2, pp. 35–52.

[33] Davis, M. Editing out video editing. *IEEE MultiMedia 10*, 2 (Mar. 2003), 54–64.

[34] Dmytryk, E. *On Film Editing: An Introduction to the Art of Film Construction.* On filmmaking. Focal Press, 1984.

[35] Duignan, M., Noble, J., Barr, P., and Biddle, R. Metaphors for electronic music production in *Reason* and *Live*. In *Asia-Pacific Conference on Computer-Human Interaction 2004 (APCHI 2004)* (2004), pp. 111–120.

[36] Ecker, D. W. The Artistic Process as Qualitative Problem Solving. *The Journal of Aesthetics and Art Criticism 21*, 3 (1963), 283–290.

[37] Edmonds, E., Turner, G., and Candy, L. Approaches to interactive art systems. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia* (New York, NY, USA, 2004), GRAPHITE '04, ACM, pp. 113–117.

[38] Edwards, W. K., Igarashi, T., LaMarca, A., and Mynatt, E. D. A temporal model for multi-level undo and redo. In *IN PROCEEDINGS OF UIST 2000* (2000), ACM Press, pp. 31–40.

[39] Eisenstein, S., Taylor, R., and Glenny, M. *Towards a Theory of Montage: Sergei Eisenstein Selected Works*. I. B. Tauris, 2010.

[40] Elio, R., and Scharf, P. B. Modeling novice-to-expert shifts in problem-solving strategy and knowledge organization. *Cognitive Science 14*, 4 (1990), 579–639.

[41] Farbood, M. M., Pasztor, E., and Jennings, K. Hyperscore: A graphical sketchpad for novice composers. In *IEEE Emerging Technologies* (2004), pp. 50–54.

[42] Findlater, L., and McGrenere, J. A comparison of static, adaptive, and adaptable menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2004), CHI '04, ACM, pp. 89–96.

[43] Findlater, L., Moffatt, K., McGrenere, J., and Dawson, J. Ephemeral adaptation: The use of gradual onset to improve menu selection performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2009), CHI '09, ACM, pp. 1655–1664.

[44] Fireman, G. Developing a plan for solving a problem: A representational shift. *Cognitive Development 11*, 1 (January–March 1996), 107–122.

[45] Fischer, T., and Lau, W. Marble track music sequencers for children. In *IDC: International Conference for Interaction Design and Children '06* (Tampere, Finland, June 2006), pp. 141–144.

[46] FlexiMusic. Kids composer. http://fleximusic.com/product/fleximusic-kids-composer, April 2015.

[47] Frøkjær, E., Hertzum, M., and Hornbæk, K. Measuring usability: Are effectiveness, efficiency, and satisfaction really correlated? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2000), CHI '00, ACM, pp. 345–352.

[48] Gick, M. L. Problem-solving strategies. *Educational Psychologist 21* (1986), 99–120.

[49] Gilhooly, K. J. Working memory and planning. In *The cognitive psychology of planning*, R. Morris and G. Ward, Eds., Current issues in thinking & reasoning. Psychology Press, 2005, ch. 4, pp. 71–88.

[50] Guindon, R. Designing the design process: exploiting opportunistic thoughts. *Human-Computer Interaction 5*, 2 (June 1990), 305–344.

[51] Hayes-Roth, B., and Hayes-Roth, F. A cognitive model of planning. *Cognitive Science A Multidisciplinary Journal 3*, 3 (1979), 275–310.

[52] Heer, J., Mackinlay, J., Stolte, C., and Agrawala, M. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (Nov. 2008), 1189–1196.

[53] Helmert, M. *Understanding Planning Tasks: Domain Complexity and Heuristic Decomposition*. LNCS sublibrary: Artificial intelligence. Springer, 2008.

[54] Hoc, J.-M. *Cognitive psychology of planning*. Academic Press Professional, Inc., San Diego, CA, USA, 1988.

[55] Hornbæk, K. Current practice in measuring usability: Challenges to usability studies and research. *International Journal of Human-Computer Studies 64*, 2 (February 2006), 79–102.

[56] IAB Canada. 2014: Internet in the media garden. White Paper, Nov. 2014.

[57] Iqbal, S. T., and Horvitz, E. Disruption and recovery of computing tasks: field study, analysis, and directions. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2007), CHI '07, ACM, pp. 677–686.

[58] Jeffries, R., Turner, A. A., Poison, P. G., and Atwood, M. E. The processes involved in designing software. In *Cognitive Skills and Their Acquisition* (1981), J. R. Anderson, Ed., Lawrence Erlbaum, pp. 255–283.

[59] Jeffries, R., Turner, A. A., Polson, P. G., and Atwood, M. E. *The Processes Involved in Designing Software*. Defense Technical Information Center, 1981.

[60] Johnson, B., and Shneiderman, B. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2Nd Conference on Visualization '91* (Los Alamitos, CA, USA, 1991), VIS '91, IEEE Computer Society Press, pp. 284–291.

[61] Johnson, H., and Carruthers, L. Supporting creative and reflective processes. *International Journal of Human-Computer Studies 64*, 10 (Oct. 2006), 998–1030.

[62] Johnson, T., and Kellstedt, P. M. Media consumption and the dynamics of policy mood. *Political Behavior 36*, 2 (June 2014), 377–399.

[63] Jonassen, D. Toward a design theory of problem solving. *Educational Technology Research and Development 48*, 4 (2000), 63–85.

[64] Jourdan, M., Layaïda, N., and Roisin, C. A survey on authoring techniques for temporal scenarios of multimedia documents. In *Handbook of Multimedia Computing*, B. Furht, Ed., vol. 5 of *Internet and Communications*. CRC Press, Boca Raton, Florida, 1998, ch. 22.

[65] Jourdan, M., Roisin, C., and Tardif, L. Multiviews interfaces for multimedia authoring environments. In *Proceedings of Multimedia Modeling 98 (MMM98)* (1998), pp. 72–79.

[66] Kelley, J. F. An empirical methodology for writing user-friendly natural language computer applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1983), CHI '83, ACM, pp. 193–196.

[67] Kreuseler, M., and Nocke, T. A history mechanism for visual data mining. In *IEEE Symposium on Information Visualization* (2004), pp. 49–56.

[68] Kurlander, D., and Feiner, S. Editable graphical histories. In *Proc. 1988 IEEE Workshop on Visual Languages* (1988), IEEE Press, pp. 416–423.

[69] Lafreniere, B., Bunt, A., Lount, M., Krynicki, F., and Terry, M. A. AdaptableGIMP: Designing a Socially-adaptable Interface. In *Proceedings of the 24th Annual ACM Symposium Adjunct on User Interface Software and Technology* (New York, NY, USA, 2011), UIST '11 Adjunct, ACM, pp. 89–90.

[70] Lafreniere, B., Bunt, A., and Terry, M. Task-centric interfaces for feature-rich software. In *OZCHI '14* (Sydney, Australia, December 2014).

[71] Lafreniere, B. J. *Task-Centric User Interfaces*. PhD thesis, University of Waterloo, 2014.

[72] Lawton, J. *Introduction to child development*. W.C. Brown Co., 1982.

[73] Layaïda, N., and yves Vion-dury, J. Multimedia authoring: A 3D interactive visualization interface based on a structured document model. In *Sixth International Conference on Human-Computer Interaction* (1995).

[74] Leung, R., Findlater, L., McGrenere, J., Graf, P., and Yang, J. Multi-layered interfaces to improve older adults' initial learnability of mobile applications. *ACM Trans. Access. Comput. 3*, 1 (Sept. 2010), 1:1–1:30.

[75] Levin, I., and Gilat, I. A developmental analysis of early time concepts: The equivalence and additivity of the effect of interfering cues on duration comparisons of young children. *Child Development 54*, 1 (1983), 78–83.

[76] Lewis, C. H. Using the "thinking aloud" method in cognitive interface design. Tech. rep., IBM, 1982.

[77] Losh, S. Gundo plugin for vim. Software: http://sjl.bitbucket.org/gundo.vim/, 2010.

[78] Lu, S., Paris, C., and Linden, K. V. Tamot: Towards a Flexible Task Modeling Tool. In *Human Factors* (2002), pp. 878–886.

[79] Lu, S., Paris, C., Linden, K. V., and Colineau, N. Generating UML diagrams from task models. In *Proceedings of the 4th Annual Conference of the ACM Special Interest Group on Computer-Human Interaction* (New York, NY, USA, 2003), CHINZ '03, ACM, pp. 9–14.

[80] Luitjens, S., and Rijckaert, A. The history of consumer magnetic video tape recording, from a rarity to a mass product. *Journal of Magnetism and Magnetic Materials 19*, 1–3 (March 1999), 17–23.

[81] Machover, T. Shaping minds musically. *BT Technology Journal 22*, 4 (2004), 171–179.

[82] Mackay, W. E. Triggers and barriers to customizing software. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1991), CHI '91, ACM, pp. 153–160.

[83] Magisto. http://www.magisto.com/, April 2015.

[84] Makedon, F., Rebelsky, S. A., Cheyney, M., Owen, C., and Gloor, P. Issues and obstacles with multimedia authoring. In *Educational Multimedia and HyperMedia, AACE* (1994), pp. 38–45.

[85] McFarland, M. C. Using bottom-up design techniques in the synthesis of digital hardware from abstract behavioral descriptions. In *Papers on Twenty-five years of electronic design automation* (New York, NY, USA, 1986), 25 years of DAC, ACM, pp. 602–608.

[86] McGrenere, J., Baecker, R. M., and Booth, K. S. An evaluation of a multiple interface design solution for bloated software. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2002), CHI '02, ACM, pp. 164–170.

[87] McGrenere, J., and Moore, G. Are we all in the same "bloat"? In *Proceedings of the Graphics Interface 2000 Conference, May 15-17, 2000, Montréal, Québec, Canada* (May 2000), pp. 187–196.

[88] Meng, C., Yasue, M., Imamiya, A., and Mao, X. Visualizing histories for selective undo and redo. In *APCHI* (1998), pp. 459–464.

[89] Microsoft. Movie maker. http://windows.microsoft.com/en-ca/windows-live/movie-maker, April 2015.

[90] Müller, P., Schubiger-Banz, S., Arisona, S. M., and Specht, M. Soundium's design tree: Supporting multimedia composition and performance. In *Digital Art Weeks 06* (ETH Zurich, Zurich, Switzerland, July 2006).

[91] Muvee. http://www.muvee.com/home/, April 2015.

[92] Nardi, B. A. Context and consciousness. Massachusetts Institute of Technology, Cambridge, MA, USA, 1995, ch. Studying Context: A Comparison of Activity Theory, Situated Action Models, and Distributed Cognition, pp. 69–102.

[93] Newell, A., Shaw, J. C., and Simon, H. A. Elements of a theory of human problem solving. *Psychological Review 65*, 3 (1958), 151–166.

[94] Newell, A., and Simon, H. *Human Problem Solving*. Prentice Hall, Englewood Cliffs, 1972.

[95] Norman, D. A. *The Design of Everyday Things*. Basic Books, September 2002.

[96] Novick, D. G., Andrade, O. D., and Bean, N. The micro-structure of use of help. In *Proceedings of the 27th ACM International Conference on Design of Communication* (New York, NY, USA, 2009), SIGDOC '09, ACM, pp. 97–104.

[97] Ormerod, T. C. Planning and ill-defined problems. In *The cognitive psychology of planning*, R. Morris and G. Ward, Eds., Current issues in thinking & reasoning. Psychology Press, 2005, ch. 3, pp. 53–70.

[98] Osburn, H. K., and Mumford, M. D. Creativity and planning: Training interventions to develop creative problem-solving skills. *Creativity Research Journal 18*, 2 (2006), 173–190.

[99] Patalano, A. L., and Seifert, C. M. Opportunistic planning: Being reminded of pending goals. *Cognitive Psychology 34*, 1 (1997), 1–36.

[100] Paternò, F. *Model-Based Design and Evaluation of Interactive Applications*, 1st ed. Springer-Verlag, London, UK, UK, 1999.

[101] Paternò, F. Model-based design of interactive applications. *Intelligence 11*, 4 (Dec. 2000), 26–38.

[102] Paternò, F., Mancini, C., and Meniconi, S. Concurtasktrees: A diagrammatic notation for specifying task models. In *Proceedings of the IFIP TC13 Interantional Conference on HumanComputer Interaction*, vol. 96. Citeseer, 1997, pp. 362–369.

[103] Paternò, F., and Zini, E. Applying information visualization techniques to visual representations of task models. In *Proceedings of the 3rd Annual Conference on Task Models and Diagrams* (New York, NY, USA, 2004), TAMODIA '04, ACM, pp. 105–111.

[104] Preece, J., Rogers, Y., and Sharp, H. *Interaction Design*, 1st ed. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[105] Rieman, J. A field study of exploratory learning strategies. *ACM Trans. Comput.-Hum. Interact. 3*, 3 (Sept. 1996), 189–218.

[106] Rubin, M., and Diamond, R. *Nonlinear: A Field Guide to Digital Film and Video Editing*. Triad Publishing, 2000.

[107] Sacerdoti, E. D. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence 5* (1974), 115–135.

[108] Salvucci, D. D. On reconstruction of task context after interruption. In *Proceedings of the 28th international conference on Human factors in computing systems* (New York, NY, USA, 2010), CHI '10, ACM, pp. 89–92.

[109] Scarr, J., Cockburn, A., Gutwin, C., and Bunt, A. Improving command selection with CommandMaps. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2012), CHI '12, ACM, pp. 257–266.

[110] Scarr, J., Cockburn, A., Gutwin, C., and Bunt, A. StencilMaps and EphemeralMaps: spatially stable interfaces that highlight command subsets. *Behaviour & Information Technology* (2015).

[111] Scott, S. D., Mercier, S., Cummings, M. L., and Wang, E. Assisting interruption recovery in supervisory control of multiple uavs. In *HFES 2006: 50th Annual Meeting of the Human Factors and Ergonomic Society* (October 16-20 2006).

[112] Seago, A., Holland, S., and Mulholland, P. Synthesizer user interface design–lessons learned from a heuristic review. Tech. Rep. 2004/20, The Open University, Milton Keynes, UK, June 2004.

[113] Seffah, A., Donyaee, M., Kline, R., and Padda, H. Usability measurement and metrics: A consolidated model. *Software Quality Journal 14*, 2 (2006), 159–178.

[114] Shneiderman, B. Promoting universal usability with multi-layer interface design. In *Proceedings of the 2003 Conference on Universal Usability* (New York, NY, USA, 2003), CUU '03, ACM, pp. 1–8.

[115] Shneiderman, B. Creativity Support Tools: A Grand Challenge for HCI Researchers. In *Engineering the User Interface*. Springer London, 2009, pp. 1–9.

[116] Sibelius. Groovy music. http://www.sibelius.com/products/index.html, April 2014.

[117] Simon, H. A. Information-Processing Theory of Human Problem Solving. In *Handbook of Learning and Cognitive Processes: Volume 5 Human Information Processing*, W. K. Estes, Ed. Erlbaum, Hillsdale, NJ, 1978, pp. 271–295.

[118] Sony. Super Duper Music Looper. http://www.sonycreativesoftware.com/products/sdml/sdml.asp, April 2015.

[119] Sony. Vegas pro. http://www.sonycreativesoftware.com/vegassoftware, April 2015.

[120] Stavness, N., and Schneider, K. A. Supporting workflow in user interface description languages. In *Workshop on Developing User Interfaces with XML: Advances on User Interface Description Langauges, AVI2004* (2004).

[121] Suchman, L. *Plans and Situated Actions: The Problem of Human-Machine Communication*. Learning in Doing: Social, Cognitive and Computational Perspectives. Cambridge University Press, 1987.

[122] Taneja, H., Webster, J. G., Malthouse, E. C., and Ksiazek, T. B. Media consumption across platforms: Identifying user-defined repertoires. *New Media & Society 14*, 6 (September 2012), 951–968.

[123] The Apache Software Foundation. Apache flex framework. http://flex.apache.org/, April 2015.

[124] Tomlinson, H. *Educational Management*. No. v. 2 in Educational Management: Major Themes in Education. RoutledgeFalmer, 2004.

[125] Turner, G., and Edmonds, E. Towards a supportive technological environment for digital art. In *OzCHI Asia-Pacific Computer Human Interaction Conference* (Downer, Australia, 2003), S. Viller and P. Wyeth, Eds., pp. 44–51.

[126] Turner, P., and McEwan, T. Activity theory: another perspective on task analysis. In *The Handbook of Task Analysis for Human-Computer Interaction*, D. Diaper and N. Stanton, Eds. CRC, New Jersey, USA, September 2003, pp. 423–444.

[127] van Wijk, J. J., van Ham, F., and van de Wetering, H. Rendering hierarchical data. *Communications of the ACM 46*, 9 (Sept. 2003), 257–263.

[128] Vermette, L., Chilana, P., Terry, M., Fourney, A., Lafreniere, B., and Kerr, T. CheatSheet: A contextual interactive memory aid for web applications. In *Proceedings of the 41st Graphics Interface Conference* (Toronto, Ont., Canada, Canada, 2015), GI '15, Canadian Information Processing Society, pp. 241–248.

[129] vidify. http://www.vidifyapp.com/, April 2014.

[130] Ward, G., and Morris, R. Introduction to the psychology of planning. In *The cognitive psychology of planning*, R. Morris and G. Ward, Eds., Current issues in thinking & reasoning. Psychology Press, New York, NY, USA, 2005, ch. 1, pp. 1–34.

[131] Yamamoto, Y., and Nakakoji, K. 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), 513–535.

[132] YouTube. Statistics. https://www.youtube.com/yt/press/statistics.html, April 2015.

Appendices

Appendix A

User Study 1 Questions and Additional Information

A.1 Interviews

A.1.1 Interviewee Information

The background information of the participants is as follows:

Professionals

(A) A sound engineer whose typical tasks include live recording and band tracking.

(B) A sound engineer who works at a studio/hall.

(C) A TV/media sound engineer/producer.

(D) A composer/sound designer for TV/games and sound engineer for voice-overs and live instruments.

(E) A composer/sound designer for intelligent robots.

Non-professionals with some MMA experience

(F) A pianist/band leader, who records gigs once in a while using the digital 4-track recorder. Uses only a single-track for the entire band (3-7 piece band).

(G) A home-recording musician (drummer). Despite her extensive knowledge about computers (CG programmer), she uses only a digital 4-track recording device.

A.1.2 Tools Used by the Interviewees

The following is the list of the software programs used by the interviewees as indicated in their answers. Some programs come with specially designed hardware devices such as control surfaces and audio interfaces.

- Digital Performer (MOTU) (Professional (A))
- MIO Console (Metric Halo) (Professional (A))
- Logic Pro (Apple) (Professionals (B), (C), and (D))
- Pro Tools (Digidesign) (Professional (C))
- Vegas (Sony) (Professional (D))
- Acid (Sony) (Professional (D))
- Sound Forge (Sony) (Professional (D))
- SoundScape (hardware DAW) (Sydec Audio Engineering) (Professional (B))

A.1.3 Interview Questions

The following is the list of questions, followed by the answers from the participants, summarised into several categories.

Question 1: What digital audio editing software(s) do you use most commonly?

Question 2: For what kind of projects do you use this software? (e.g. film/TV composition, film/TV sound effects, song-writing, recording bands, recording classical, etc)

Question 3: What are the typical "time lines" on your projects?

e.g.) 30 seconds of music which you start and finish over two 12-hour days of working? Or 5 minutes of music which you put together in about 48 hours spread over a period of 4 weeks? etc.

Question 4: What is the work-flow? Are there any particular "bottlenecks"?

Question 5: Many software programs seem to closely emulate existing hardware devices (fader, pan, and other little knobs and buttons here and there). Does this work for you?

Question 6: What are the most common problems you run into that don't have a good solution? Are there any common problems that arise frequently but are easy to solve?

Question 7: If you're familiar with the concept of "user interfaces", do you have any comments about the interfaces of existing audio editing programs?

Question 8: Have you ever used tools other than the keyboard-and-mouse, (e.g. pen-and- tablet) for music editing? If so, did you like it, and if not, would that be of interest in the future?

Question 9: Would you like to see more features other than the standard "waves" or spectral visualizations of music tracks? (e.g. would it help if you could see some more musical features such as a melodic contour or rhythmic content) (transcription)?

Question 10: Do any editing programs allow you to "annotate" tracks with comments? (e.g. circle certain sections of a take, mark one part as "good", another part as "messy", etc.) Would this be of interest?

Question 11: How do you organize your music clip files? Group the clips from the same session into the same folder? Is there any way you know the contents of the files without listening to them?

A.2 Surveys

A.2.1 Survey Questions

General Experience in MMA tasks (Pre-survey questions) Do you have experience in video editing or sound editing/music production?

Video editing:

- None
- Less than 1 year
- 1 to less than 3 years
- 3 to 5 years
- More than 5 years

Sound editing:

- None
- Less than 1 year
- 1 to less than 3 years
- 3 to 5 years
- More than 5 years

If you have experience in video/sound editing, in what context did you perform editing? (Select all that apply)

1. Very short videos to only trim the beginning and ending
2. Short videos captured with your phone to share with your friends/family
3. Longer (more than 30 minutes) home videos captured with camcorders
4. Multitrack recording for yourself/your bands
5. Others (Specify)

Question 1: Preparation and initial steps of MMA tasks You are planning to edit home footage of your week-long vacation to a resort. You have about 2 hours worth of videos and 300 pictures from your own camera as well as your friends' pictures. List the first 3 steps that you need to do in order to prepare/start this editing task.

Question 2: Most likely scenarios Which of the following scenarios best represents your actions if you were to create a short movie clip as a "birthday present" for a friend. Rate each of the scenarios:

1. Scenario # 1 You first collect a bunch of pictures and videos of your friend with you and other friends and find some music track that he or she likes. You will then choose which pictures and videos to use. Based on the chosen media, you will come up with a story or structure of the video that can best utilize the media that you already have. You will then put those media together in a rough order along with the music you have selected to see how the resulting video looks. Starting from this rough edit, you will modify some details until you are satisfied with the video.

   (a) Would almost never do that

   (b) Unlikely to do that

   (c) Would probably do that

   (d) Would almost certainly do it this way

2. Scenario # 2 You first think about what kind of story that you want to tell with this video, such as the history of your friendship with this person or the accomplishments of her/his life etc. You will then start collecting pictures and videos that you already have. If you cannot find suitable media, you will go

out and ask mutual friends if they have something you're looking for, and/or take pictures or shoot videos of places or things that are important to the story of the video. You will then carefully decide how these media should be put together to best match the storyline that you have in mind before you actually start working on these clips.

   (a) Would almost never do that
   (b) Unlikely to do that
   (c) Would probably do that
   (d) Would almost certainly do it this way

3. Scenario # 3 You first think of rough sketches of an ideal video structure, such as the opening credit, memory scenes, messages, the ending, and so on, and then work on one section at a time until that section is more or less complete. You continue this process for all the rest of the sections. You will then put together all the completed sections to review the entire video. Some details are fixed after viewing this draft. .

   (a) Would almost never do that
   (b) Unlikely to do that
   (c) Would probably do that
   (d) Would almost certainly do it this way

Question 3: Information to be available You are in the process of editing a video of your friend's wedding reception. There were 3 video cameras used; one from the back of the room to capture the entire room but mainly focused on the groom and the bride, one from the front of the room to capture most of the tables, and one camcorder that moved around the tables and captured close-ups of the groom and bride as well as the guests. The edit is almost half-way completed, but there is much more work to be done to finish this project.

What types of information will be helpful for you to be able to view while performing this editing project. List all that you'd like to be able to see.

In addition to the list you have created above, which of the following information would you want to see?(The list is shown in Table 3.1.)

Information	definitely	maybe	not sure
List of Goals/To-do list (e.g., colour correction for the opening, re-mix audio, etc)	☐	☐	☐
List of actions performed (as in undo list)	☐	☐	☐
Project media and their original locations as well as where they are used in the workspace	☐	☐	☐
Short notes for each editing action (e.g., readjusting clip 1, etc)	☐	☐	☐
Diagram indicating project stages and current stage	☐	☐	☐
Preview of actions before actually executing them (i.e., play short animation/video of how to perform certain things)	☐	☐	☐
Review of actions to remind yourself what you did (play animation/video of what you did, similar to preview)	☐	☐	☐
Description of each data/cut/clip (e.g., clip 1 used for the explosion scene, etc)	☐	☐	☐

Table A.1: List of additional information.

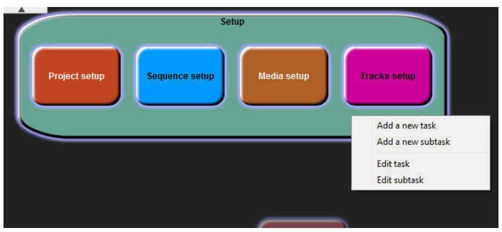
Question 4: Difficulties that existing tools exhibit If you have any experience using the multimedia authoring tools such as video editors (e.g., iMovie and/or Windows Movie Maker), and/or digital music production tools (e.g., GarageBand and/or Logic Pro), what were the biggest obstacles you had while learning them?

OR

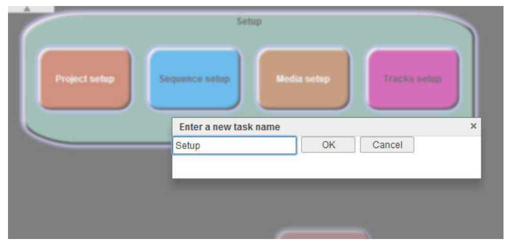
If you have no experience using multimedia authoring tools, but you are/were interested in creating and editing your own videos or recording and producing your own music, what do you think is/was preventing you from learning these tools and starting to use them?

Appendix B

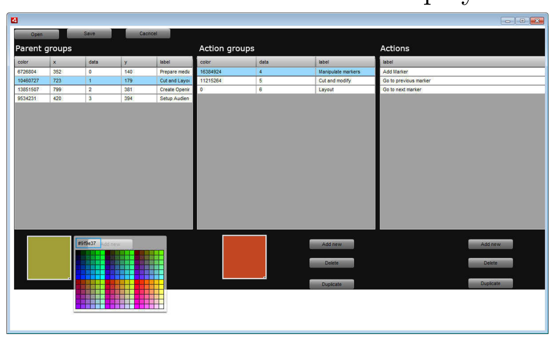
Task-View editor of the ITV



(a) A task-view can be edited directly by using a pop-up menu.



(b) Once a menu item is selected (e.g., "Edit task"), a corresponding dialogue will be displayed.



(c) The ITV also provides a more advanced editing environment of a task-view.

Figure B.1: The ITV provides a few ways to edit task-views. Figures (a) and (b) show the editing mode within the task-view itself, in which the user can use a pop-up menu to perform basic editing of the task-view. Figure (c) shows an advanced task-view editor in which the user can modify details of the task-view. Note that, in the current implementation, this editor is meant to be used by users who provide novice users with initial task-views for certain projects.