COAR: An Objective-C Framework for Usability Data Collection

Joji Mori and Daniel Woo
School of Computer Science and Engineering
The University of New South Wales
joji@student.unsw.edu.au
danielw@cse.unsw.edu.au

Abstract The COAR project is developing a framework and associated tools to streamline the process of usability video analysis. The project requires a flexible mechanism for storing and retrieving synchronised media and other arbitrary data from a wide range of sources. The COAR framework leverages Apple's QuickTime technology for this, since it is not limited to storing just video and audio. Rendezvous and Distributed Objects have been incorporated opening up a range of new possibilities for multi-user and wireless usability testing applications.

Introduction

Observation studies of people using technology can provide valuable information about the events of the interaction, revealing design flaws that may surprise the original designers and developers — "Hey, they aren't meant to do that" or "They should have known to press the other button".

Many studies have used video technology to record the events of an experiment, but the review process is time consuming — one hour of video can take at least, if not more than one hour to analyse. Digital video can be used to improve the time to find data suitable for analysis. If the video stream is annotated appropriately, the time to search for relevant events can be reduced. Collecting data from observers and the system under test, in a time synchronised manner will provide a data store that contains a rich description of the events that transpired.

A usability experiment typically consists of collecting data, observing user actions, analysing the data and reporting the results. In fact this is a generalisation of many scientific observation experiments. COAR aims to unify the activities of **C**ollecting data, **O**bserving users, **A**nalysing information and **R**eporting results, hence the acronym. In this paper we discuss two components of the COAR framework: Collection and Observation.

We are focussing on usability experiments primarily for software-based applications but the concepts can be used for applications outside of software evaluation. The typical usability software development approach is iterative where user testing is conducted throughout from paper-based design ([1], p. 93) to the software implementation. We record and measure what the user did when using a software interface identifying when users make errors, how often they make errors, when they do not understand terminology or are confused by any of the details presented by the application. We are also interested in when they need to use additional resources such as on-line help or when specific user interface elements are used. Essentially we are looking for any issues that help explain why the user does not work effectively with the system.

This paper introduces usability experiments and describes the overall goals of COAR. A summary of existing tools is analysed then we show how COAR fits into the usability evaluation environment. COAR system implementation emphasises how Apple technologies are incorporated in developing the framework. Finally, a tool called the *Usability Observer* is introduced which enables rapid expert logging to the COAR framework.

Usability Experiments

In a typical usability experiment, we would assign the user a specific set of tasks to carry out using a software package (or several packages). We then observe (and record) the person attempting to carry out the tasks. A verbal protocol ([1], pp 195–198) may be used to record what the user is thinking throughout. It is important that we do not guide the user too much since we are exploring how they use the package not how they can act under the facilitator's instruction.

Observers take notes on paper whilst video and audio of the experiment is recorded with a video camera. One or more video sources could be used to capture the user's facial expression, the overall desk area including keyboard and mouse and a video of the actual screen the user was looking at.

At the completion of the experiment (or series of experiments) the data analysis phase commences. This consists of converting the paper-based notes into a more suitable format for analysis and reviewing recorded audio and video material. The review process may consist of a fine or coarse annotation of the recorded material. For instance, "user started phase 1" at time 00:05, "error with data entry" at time 00:17, "user did not understand the meaning of the next button" at time 00:23, "user completed phase 1", 00:27. Finally, a report will be created documenting the overall findings.

Goals of COAR

The purpose of COAR is to provide common infrastructure to store and retrieve any data associated with a usability test in a multi user environment. Several channels of various types of usability information must be retrieved and presented in a synchronised manner providing analysts a detailed account of what occurred at any point in the experiment. Migrating from paper notes has the advantage of improving the time to locate specific segments of video.

As an example we could locate all instances of when the observer noted a user error or search for all instances that the user clicked a button named "Cancel" and review 30 seconds of video around that point.

Previous Work

The application of video analysis tools has been investigated in a number of existing tools, especially since affordable video recording technology became readily available. Device control via a serial port expanded the possibilities of linking computers to video recording equipment. The video medium was however still time consuming to analyse. In the early 1990s, usability applications using computers and video were being pursued ([2],[3]). New applications have also been developed in sports video analysis ([4],[5]). The general need to

retrieve video from databases is receiving large attention given the growth of network accessible video material [6].

MacSHAPA [2] was created by the Department of Mechanical and Industrial Engineering at the University of Illinois. It is a package that assists analysts with observational data to extract useful information. Annotating video data for retrieval through VCR control has been applied in a number of human observation experiments for example, usability and psychology experiments.

The DRUM (Diagnostic Recorder of Usability Measurement) [3] software was developed as part of the project Metrics for Usability Standards in Computing (MUSiC). DRUM is used to obtain metrics for human effectiveness when using a software system. These metrics include task times, help/search times, efficiency, and productivity periods. As with MacSHAPA, DRUM can be integrated with a programmable VCR for reviewing recorded information based on time-based logs.

MacSHAPA and DRUM are useful tools for single user annotation of time-based data. Both use programmable VCR technology to view the video footage. This has obvious drawbacks given access speeds of digital data. Using QuickTime for data storage means that we can bundle the complete usability session data into one movie file. Analysis can then be performed on any platform that supports QuickTime. COAR also leverages recent networking advances to support a distributed multi-user analysis environment.

The COAR Environment

There are many different types of usability evaluation techniques [7]. Heuristic evaluations, observational analysis and event logging [8] are some common examples. The COAR framework is at the centre of these evaluation techniques, assisting usability experts in the experimental process. A familiar scenario is a new application that needs to be usability tested with a novice user, at most two evaluators and a half hour timeframe.

Figure 1 depicts this scenario. The evaluators watch the user and takes notes while the operating system captures events.

The *Usability Observer* (see below) is a tool developed at CSE that allows a usability expert to log comments about the user's interaction. It can also be used to record notes for heuristic evaluations. Other channels of usability data come from numerous video cameras and a desktop boundary microphone. A scan converted video recording of the screen is also captured. Events and internal application state are provided by the application and stored on a COAR server. This summarises possible types of data COAR will manage.

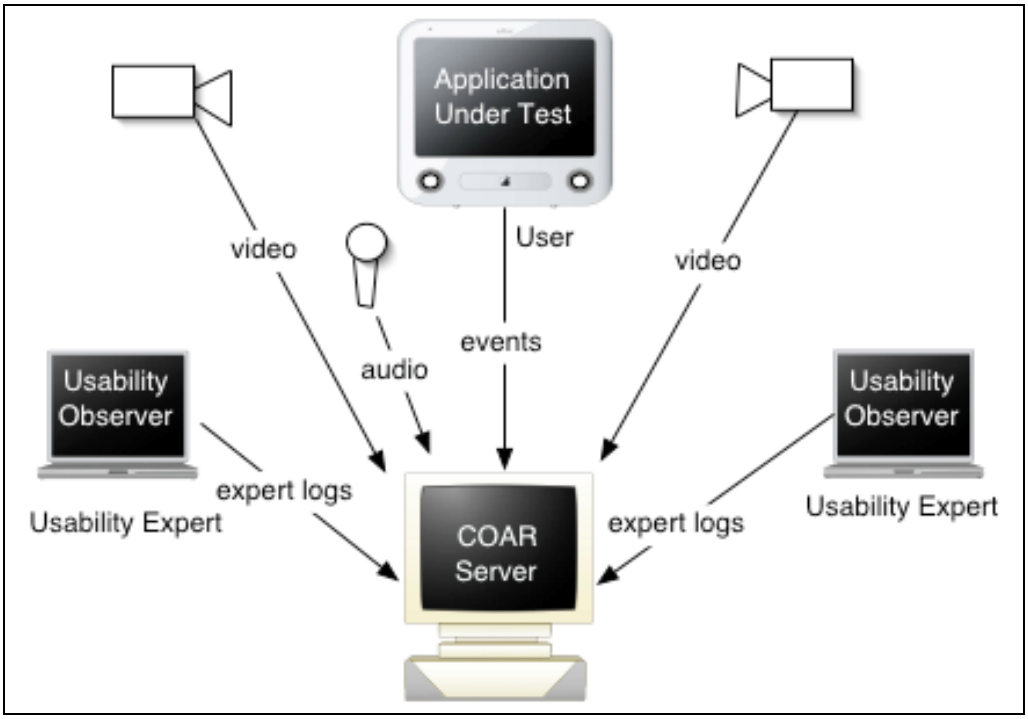

Figure 1 - Scenario

COAR System Overview

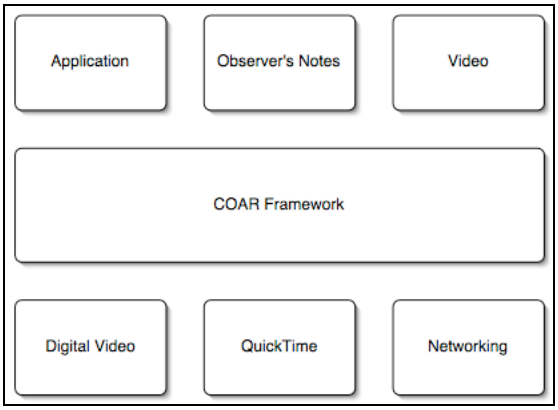

Figure 2 - COAR Framework Context

Figure 2 depicts the COAR framework context. User events from the application under test, observer notes and video and audio streams, are examples of time-based usability data to be collected. Instead of creating our own time-handling architecture, the QuickTime framework [9] is used to manage COAR data. A QuickTime movie is broken up into any number of tracks. This is extremely useful considering there can be any number of cameras and microphones. QuickTime also supports the idea of a text track. Text tracks (originally conceived for movie subtitles) are special tracks used to store and retrieve text. For example, each observer's notes from the usability session can be stored in a separate text track.

The framework is implemented using Objective-C utilising the rich Cocoa API. Multi-document, multimedia integration, Aqua [10] user interface elements, Rendezvous [11] and distributed networking all provide a rapid path to application development. Rendezvous is used to simplify the way that the application under test can locate the COAR server on the local network. By specifying a server name (e.g. "iTunes Usability Test"), Rendezvous handles the publishing and discovery of these COAR services.

Distributed objects simplify the task of exchanging information between a client and server process. Instead of explicitly transferring data with TCP/IP sockets, we can use higher level messaging at the Objective-C level between two processes running on separate machines. In our case, the client (application under test) has a proxy to an object on the COAR server and uses the proxy to call methods from an object on the server.

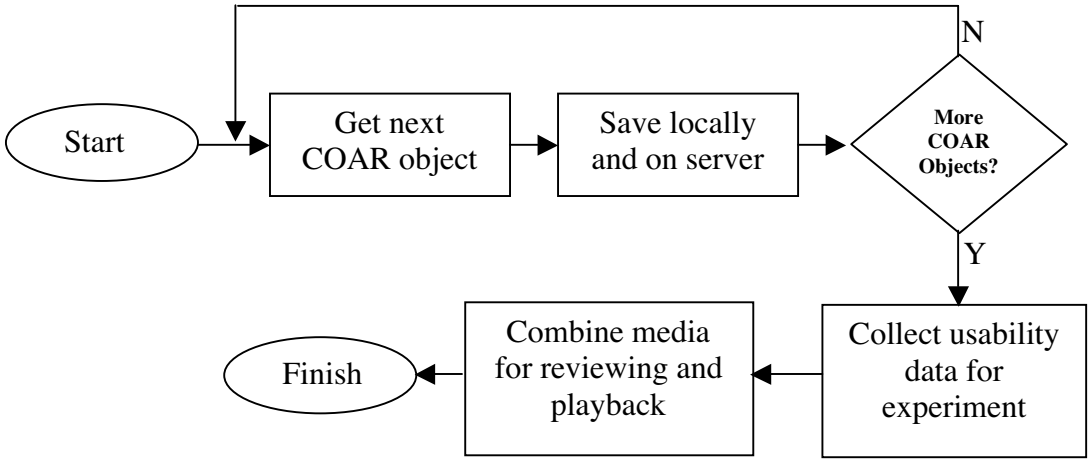

Figure 3 - Flow chart for COAR

The flow diagram (Figure 3) breaks the project into 2 phases

1. Collecting COAR objects during the usability evaluation
2. Combining the media for playback/review on completion of the experiment

A COAR object can be thought of as a discrete usability data item. It may be an event (eg. button "Enter" clicked), a string containing comments from a usability expert, or a segment of video/audio footage. The COAR object may be saved on the local machine and/or on a network server. By saving to a network server, there is a possibility for real-time processing of the data to provide instantaneous feedback to the facilitators. Multiple sources can generate usability data simultaneously. By storing usability data in a QuickTime movie, we have a synchronised representation of the experimental data that can be retrieved using the QuickTime framework.

We are exploring the concept of automating event collection with COAR. An example would be to provide an English like representation of the user interaction with Mac OS X, e.g.

> *Application changed from iTunes to TextEdit,*
> *user typed 14 characters into the Information text field,*
> *selects the menu item "Copy" from the Edit menu in menu bar,*
> *the focus is now the text field*
> *user presses the hot key "control+v" to paste the text.*

To some extent, AppleScript recording could partially provide this information but it assumes that all applications report all Apple Events, which is not the case. A low level approach is required to extract low level events and interaction with user interface controls and physical devices [12]. This form of analysis is closely related to GOMS and its derivatives [13\.

Usability Observer

Usability Observer is a logging tool that collects observers' electronic comments and comment category for an observation experiment, storing the time-stamped information to enable efficient search of a video stream during the analysis phase. It can communicate with the COAR framework to record observer's comments so they can be synchronised with other time-based media collected during an experiment. The COAR framework allows the data to be saved on the local machine and also send it to the COAR server over a network. This allows wireless operation whilst maintaining a local backup of the data.

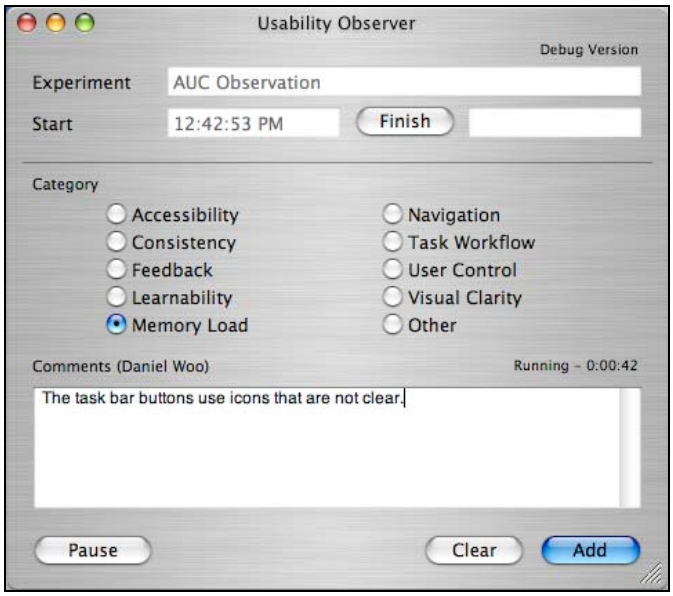

Figure 4 - Usability Observer

The interface has been designed for data collection, not analysis or review. The primary goal is to capture the comments and category as quickly as possible. The interface can be used solely with the keyboard. All radio buttons have short cut key equivalents corresponding to the first letter of the category.

QuickTime Player integration has been developed so that studies of pre-recorded material can be carried out. This will be used in forthcoming Human Computer Interaction teaching laboratories [14].

The general principles of the Usability Observer can be applied to many other experimental applications that require time stamping of observations such as experimental psychology and sports events video analysis.

The Future Directions for COAR

This paper has described how we have developed a working framework to support usability testing and video analysis. We can log information from several sources including the application and Usability Observer.

QuickTime integration meant that we did not have to reinvent a storage architecture for time-coded information. Specifically, text tracks are used for storing textual metadata meaning all annotations are stored within the movie.

Rendezvous and distributed objects assisted programming network service discovery issues enabling an environment to explore multi-user and wireless applications.

A long-term goal for COAR is that it becomes an integral part of our Human Computer Interaction teaching syllabus. By introducing the principles of COAR to student developers, they are exposed to usability-aware software development.

References

[1] NIELSEN J. (1993) *Usability Engineering* Morgan Kaufmann.

[2] SANDERSON P., SCOTT J. JOHNSTON T., MAINZER J., WATANABE L., JAMES J. (1994) *MacSHAPA and the enterprise of exploratory sequential data analysis (ESDA)* International journal of Human-Computer Studies **41** (5) 633 - 681.

[3] MACLEOD, M., RENGGER, R. (1993) *The Development of DRUM: A Software Tool for Video-assisted Usability Evaluation* Proceedings of the HCI'93 Conference on People and Computers VIII, Loughborough UK, 293-309.

[4] SPORTSCAD MOTION ANALYSIS (2002) http://www.sportscad.com/

[5] LIVE CHANNEL PRO 2.0 (2003) http://www.channelstorm.com/

[6] WACTLAR H., CHRISTEL M., GONG Y., HAUPTMANN A. (Feb 1999) *Lessons Learned from Building a Tetrabyte Digital Video Library* IEEE Computer **32** (2) 66-73.

[7] HOM J. (1996) *The Usability Method Toolbox*, http://www.best.com/~jthom/usability/ .

[8] NIELSEN J. (1993) *Usability Engineering* Kaufman 195-198.

[9] APPLE COMPUTER INC. (2003) http://developer.apple.com/quicktime/

[10] APPLE COMPUTER INC. (June 2002) http://developer.apple.com/techpubs/macosx/Essentials/AquaHIGuidelines/

[11] APPLE COMPUTER INC. (December 2002) http://developer.apple.com/techpubs/macosx/Networking/Rendezvous/dns_discovery_api/index.html

[12] HILBERT D. M., REDMILES D. F. (2000) *Extracting Usability Information from User Interface Events* ACM Computing Surveys **32** (4) 384-421.

[13] BAUMEISTER L. (2000) *A Comparison of Tools for Building GOMS models* CHI Proceedings 1-6 April, Netherlands.

[14] WOO D.T. (2003) *Evolving the Learning and Research Experience in Human Computer Interaction* (These Proceedings) Apple University Conference 2003.