

 Université du Québec École de technologie supérieure	PROJECT LESIA	DOCUMENT NO. uObserve_Specs_Eng_A4for mat	DATE 2007-11-03	VERSION 1.0
	TITLE UOBSERVE SOFTWARE SPECIFICATION		PAGE 1	PAGES 16

UOBSERVE SOFTWARE SPECIFICATION

Authors:

Sylvie Trudel
sylvie.trudel@crim.ca

Jean-Marc Lavoie
jeanmarc.lavoie@videotron.ca

 Université du Québec École de technologie supérieure	PROJECT	DOCUMENT NO.	DATE	VERSION
	LESIA	uObserve_Specs_Eng_A4for mat	2007-11-03	1.0
	TITLE	PAGE	PAGES	
	UOBSERVE SOFTWARE SPECIFICATION	2	16	

Table of content

Section	Page
1. Introduction.....	3
1.1 Purpose and scope	3
1.2 Opportunity description: tooling usability testing.....	3
1.3 Project objectives.....	3
1.4 Definitions and acronyms.....	3
1.5 References	4
2. uObserve Description	4
2.1 uObserve perspective	4
2.2 Features and functions.....	5
2.2.1 uObserve functionalities	5
2.2.2 Functionalities considered for future development.....	6
2.3 User characteristics	6
2.4 Assumptions, constraints, and dependancies	6
3. SPECIFIC REQUIREMENTS	7
3.1 External interfaces requirements.....	7
3.1.1 User interface.....	7
3.1.2 Hardware interfaces	7
3.1.3 Software interfaces.....	7
3.1.4 Communication interfaces	8
3.2 Functional requirements.....	8
3.2.1 Start uSleuth	10
3.2.2 Close uSleuth.....	10
3.2.3 Connect uSpy to uSleuth.....	11
3.2.4 Send events	11
3.2.5 Record an experiment	12
3.2.6 Open a recorded experiment.....	13
3.2.7 Playback an experiment	14
3.2.8 Seek from image	14
3.2.9 Scroll and select from the event list	15
3.2.10 Display “About uObserve”	15
3.3 Quality requirements.....	15
3.3.1 Reliability.....	15
3.3.2 Availability	15
3.3.3 Security	16
3.3.4 Maintainability	16
3.3.5 Portability	16

List of figures

Figure	Page
Figure 1 –uObeserve system perspective.....	5
Figure 2 – uSleuth user interface prototype.....	7
Figure 3 –uObserve use case diagram.....	8
Figure 4 –uSleuth finite state machine.....	9
Figure 5 –Button status in “Idle” state.....	9
Figure 6 –Button status in “Playback” state.....	9

 Université du Québec École de technologie supérieure	PROJECT LESIA	DOCUMENT NO. uObserve_Specs_Eng_A4for mat	DATE 2007-11-03	VERSION 1.0
	TITLE UOBSERVE SOFTWARE SPECIFICATION		PAGE 3	PAGES 16

INTRODUCTION

Purpose and scope

The purpose of this document is to define and communicate the software requirements of the first release of a system designated as “uObserve”. uObserve is an application of software usability testing that combines a sniffer library, a video camera and a microphone. The requirements are documented in a way to provide a common understanding to the stakeholders. Those requirements will be verified through reviews. The structure of this software specification is inspired by IEEE standard 830-1998 [1]. Diagrams and detailed use cases it contains apply UML 2 notation and description [2].

The uObserve system is developed for LESIA, an advanced interface and synthetic environment laboratory at École de Technologie Supérieure (ÉTS) in Montreal, Canada. It is meant as a proof of concept for usability testing which could be evolved by LESIA members afterwards.

Opportunity description: tooling usability testing

During a usability testing experiment at LESIA, graduate students must familiarize themselves with a spy program. That spy program is used to gather data that will then be analysed. In order to perform a good data analysis, two (2) persons are required: one user repeatedly entering data with the same interface (10 times) and one experimenter noting the first user's gestures and behaviour. Noted gestures and behavior are then used to try to explain delays and to calculate the user's learning curve. While the first user enters data, the spy program captures every mouse movements and keyboard clicks in an event log.

Experience has shown that the captured notes are often incomplete and difficult to synchronize with entered data when trying to explain delays. Furthermore, the experimenter cannot ask the user to redo the experiment since his learning curve is already over. Analysing experiment data then becomes difficult, it even lead to wrong conclusion in certain cases.

It is possible to film the user during a session. The equipment to do so is expensive, often difficult to move, and may require the skills of a technician to operate properly. Furthermore the resulting video would not be synchronized with the captured data.

Project objectives

The objectives of the project are: to develop a working system that records an audio/video sequence of a user and synchronized with a log of keyboard and mouse events that could be played back on request. This proof of concept system should be developed and tested within 2 months with a team of 2 software engineers. The equipment used should not be too expensive and should be operated from standard PC workstations, much like a Webcam under Windows. Such workstations are already available at LESIA. The record and playback functions shall not affect the performance or the behaviour of the spied user application.

Definitions and acronyms

CFFP	COSMIC-Full Function Points.
COSMIC	Common Software Measurement International Consortium.
CSV	Comma Separated Values.
ÉTS	École de Technologie Supérieure.

 Université du Québec École de technologie supérieure	PROJECT	DOCUMENT NO.	DATE	VERSION
	LESIA	uObserve_Specs_Eng_A4for mat	2007-11-03	1.0
	TITLE	PAGE	PAGES	
	UOBSERVE SOFTWARE SPECIFICATION	4	16	

JDK	Java Development Kit.
JMF	Java Media Framework.
LAN	Local Area Network.
LESIA	Laboratoire d'environnements de synthèse et interfaces avancées.
ms	Milliseconds.
Sniffer	Software program or library that spies upon an application software, thus recording keyboard and mouse events. A sniffer is also called "spy ware" or "spy program".
SRS	Software Requirements Specifications.
TBD	To Be Determined.
UML	Unified Modeling Language.
Usability test	Experiment performed with user participants aiming at discovering any flaws or defects in a user interface. It implies developing an experiment protocol, getting an approval from an ethics committee, recruiting appropriate participants, developing software and/or hardware, performing the experiment, and analysing experiment data.

References

- [1] IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications (SRS), IEEE Computer Society.
- [2] Arlow, J. and Neustadt, I., UML 2 and the unified process: Practical object-oriented analysis and design, 2nd edition, Addison-Wesley, 2005.

UOBSERVE DESCRIPTION

uObserve perspective

uObserve is a tool that supports usability testing of user interfaces. It records audio and video data of a user running a spied application. uObserve synchronizes a log of events sent from the sniffer (records of keyboard and mouse movements). uObserve is composed of two (2) subsystems: the uSleuth server executing on a different workstation than the uSpy client. uSpy is a sniffer library that would be integrated to an existing Java client application to be observed. uSpy sends a log of keyboard and mouse events to uSleuth via a Local Area Network (LAN). uSleuth records and playback audio/video files.

At least one camera shall be used, as shown on Figure 1.

 Université du Québec École de technologie supérieure	PROJECT LESIA	DOCUMENT NO. uObserve_Specs_Eng_A4for mat	DATE 2007-11-03	VERSION 1.0
	TITLE UOBSERVE SOFTWARE SPECIFICATION		PAGE 5	PAGES 16

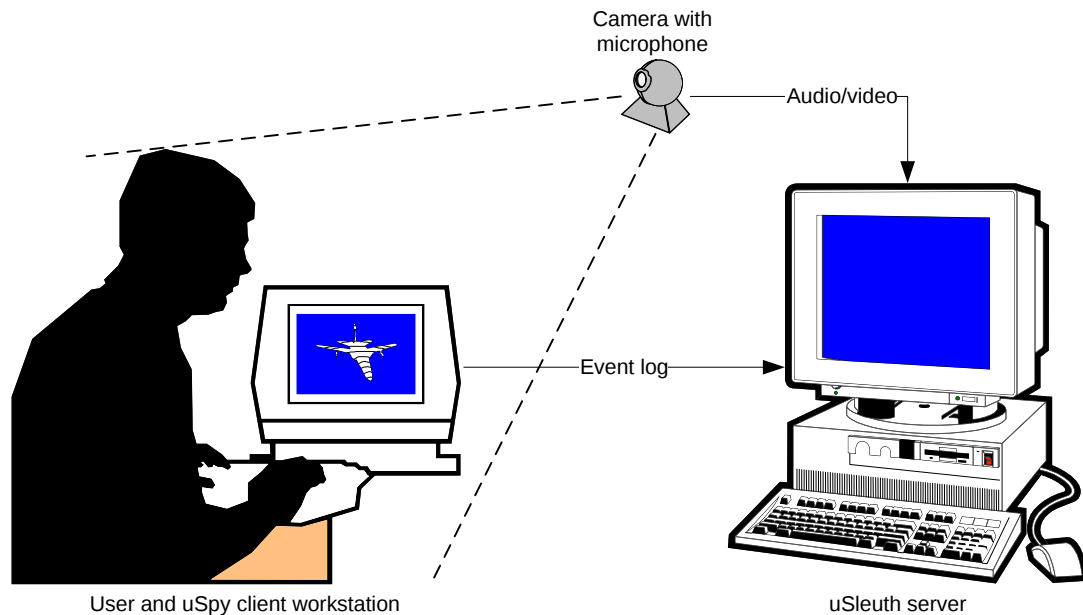


Figure 1 –uObeserve system perspective.

Features and functions

uObeserve functionalities

In agreement with the project sponsor, LESIA's director, an expert in user interface ergonomics, it has been decided to implement uObeserve using the video tape player/recorder paradigm. More specifically, within this paradigm, the cassette on which the recording is made will be materialized by an electronic file. A consequence is that there is only one active file open at anyone time; in order to open a new file, the active one must be ejected (or closed).

Log application events (mouse and keyboard events) (essential):

Filter out selected event types;

Record audio/video frames of a user running an application using the uSpy library on the uSleuth station (essential):

At least one camera, with a (minimal) image resolution of 320 x 240:

One microphone only.

On request, playback of the event log synchronized with the audio/video files (essential):

Open [existing] file;

Play [current file]:

Synchronize audio/video frames with the event log;

Highlight the current event;

Video scroll bar (one frame at a time);

Event scroll bar (one event at a time);

Pause;

Stop;

Record;

Eject [current] file;

Seek position within the video progress bar (important);

Scroll the event list and select an event within the event log (important);

 Université du Québec École de technologie supérieure	PROJECT	DOCUMENT NO.	DATE	VERSION
	LESIA	uObserve_Specs_Eng_A4for mat	2007-11-03	1.0
	TITLE	PAGE	PAGES	
	UOBSERVE SOFTWARE SPECIFICATION	6	16	

Display event log (essential):

Time in ms since the beginning of record;

Event type;

Event logo for a subset of event types (nice to have);

Time difference between the current and the preceding event in ms;

Duration (or delta time) since the last event of the same type in ms;

Event parameters (text string);

Import log files (essential):

Load the event log directly in Excel;

Load the event log directly in a text editor running under Linux;

Functionalities considered for future development

The following functionalities were considered but a joint decision has been made by the project team and the sponsor to implement them in a subsequent release.

Support more than one camera:

- Two cameras (important);
- Three cameras or more (nice-to-have).

Slow motion playback. This could be simulated by seeking events or video frames;

Reverse Play. This could be simulated by stepping through events or video frames;

Electronic note taking while recording with a time stamp link;

Reproducing the user application on the server station (uSleuth). This could be done by aiming one of the video cameras directly at the user screen;

Pattern identification within the event log. This would require a significant amount of effort and time;

Record a Web application. Significant effort and time would be required in order to adapt the sniffer from a Java client-server application to a Web application.

User characteristics

Potential uObserve users are LESIA experimenters who are skilled software engineers or persons with Java programming and/or usability testing knowledge and skills.

Assumptions, constraints, and dependancies

The amount of time required to queue an event shall not exceed 20 ms;

Java Media Framework (JMF) shall be used to manage audio and video aspects, which implies that the system will be developed and executed under Windows operating system;

uObserve will be developed in English, as well as its artefacts (code and comments, Maven scripts, and unit tests);

The user Java application shall be run locally on the workstation;

JDK version 1.4 + shall be used;

uObserve reliability is dependant on the reliability of the LAN on which it is operated;

The user application to spy upon must be modifiable by this project team members or any other LESIA member in order to include the uSpy library;

 Université du Québec École de technologie supérieure	PROJECT LESIA	DOCUMENT NO. uObserve_Specs_Eng_A4for mat	DATE 2007-11-03	VERSION 1.0
	TITLE UOBSERVE SOFTWARE SPECIFICATION		PAGE 7	PAGES 16

SPECIFIC REQUIREMENTS

External interfaces requirements

User interface

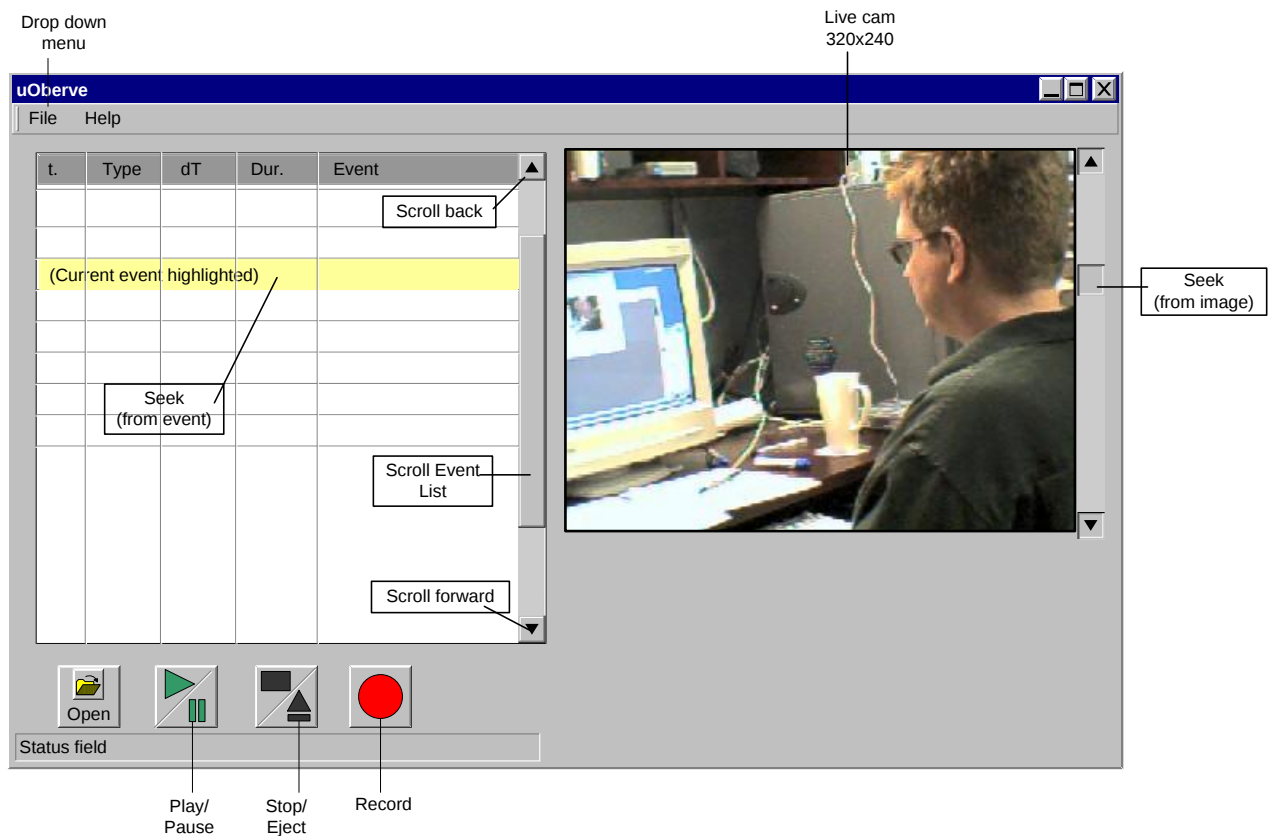


Figure 2 – uSleuth user interface prototype.

Hardware interfaces

Cameras

The cameras should support color (RGB format) and provide (at least) 320 x 240 pixels resolution.

Microphone

Only one microphone shall be supported. Audio recording shall be sampled according to the following parameters: 16 bits, 44100 Hz, stereo.

Software interfaces

A Java application shall integrate the uSpy library in order to send an event log to uSleuth.

 Université du Québec École de technologie supérieure	PROJECT LESIA	DOCUMENT NO. uObserve_Specs_Eng_A4for mat	DATE 2007-11-03	VERSION 1.0
	TITLE UOBSERVE SOFTWARE SPECIFICATION		PAGE 8	PAGES 16

Communication interfaces

A TCP/IP link shall be established between the client workstation (using the uSpy library) and the server workstation (uSleuth). The IP address configuration shall be set manually in a configuration file of the application using the uSpy library.

Functional requirements

Figure 3 provides the use case model of the uObserve system. The following sections provide detailed use cases as applied during a usability test. Unless otherwise specified, most use case primary actor is the Usability Tester (UT).

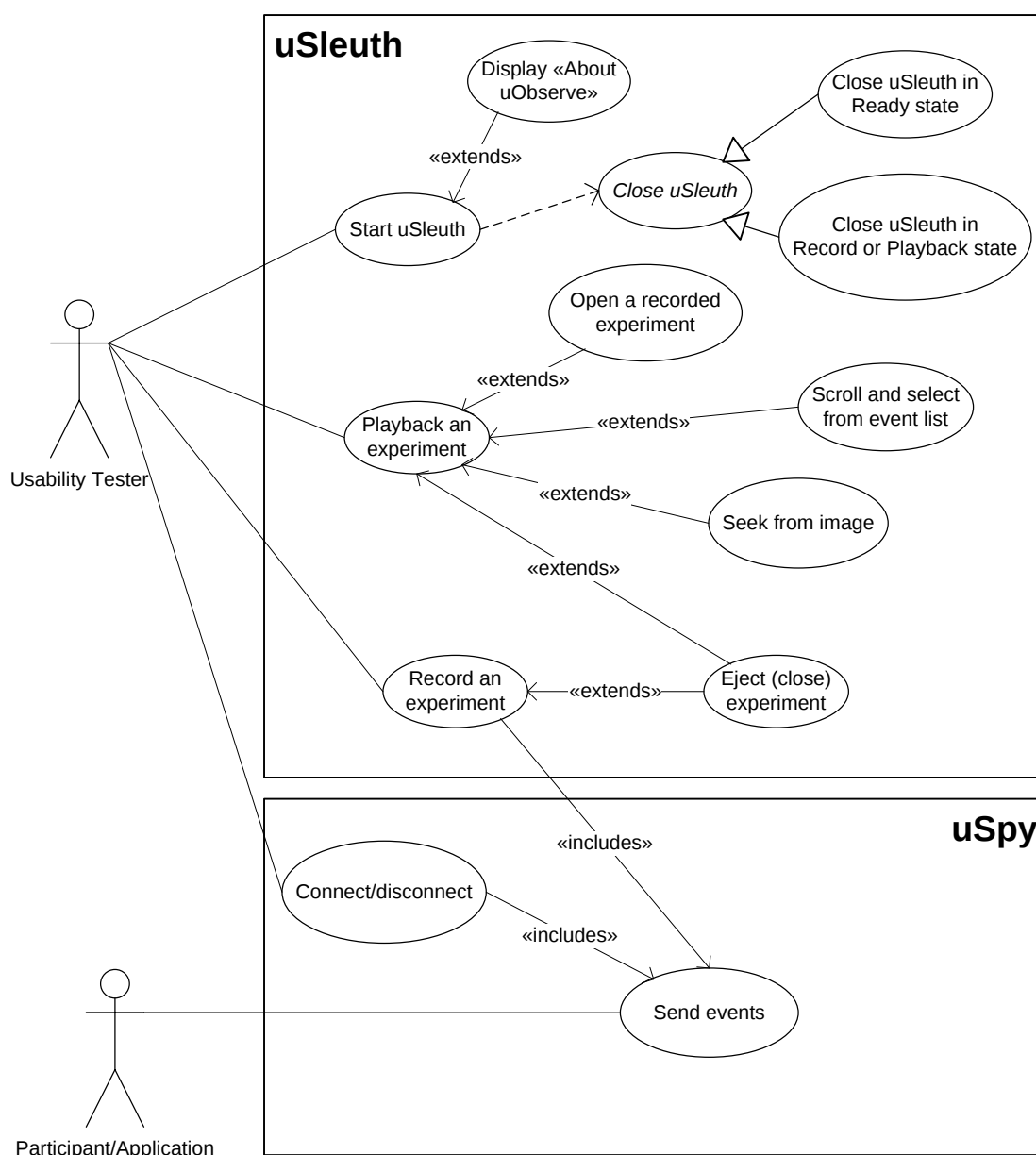


Figure 3 –uObserve use case diagram.

 Université du Québec École de technologie supérieure	PROJECT LESIA	DOCUMENT NO. uObserve_Specs_Eng_A4for mat	DATE 2007-11-03	VERSION 1.0
	TITLE UOBSERVE SOFTWARE SPECIFICATION		PAGE 9	PAGES 16

uSleuth behaves as illustrated in the state machine at Figure 4. Activation of the different fonctions in the user interface shall respect states as shown below.

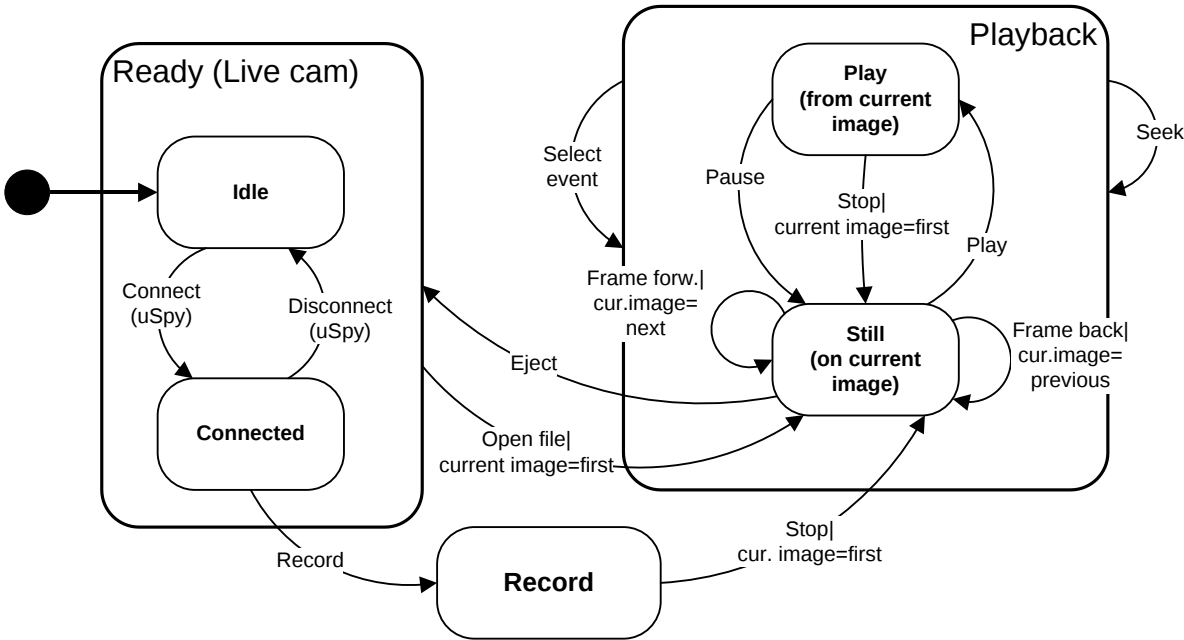


Figure 4 –uSleuth finite state machine.

In the “Idle” state, only the “Open” button shall be active, forcing the Usability Tester to open an existing experiment file to get into the playback mode, as shown in Figure 5.

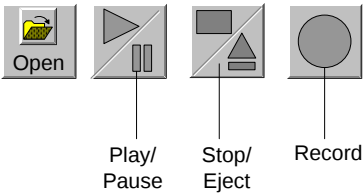


Figure 5 –Button status in “Idle” state.

In the “Playback” state, an experiment file is opened and the Usability Tester can play/pause or stop/eject the current experiment file, as shown in Figure 6.

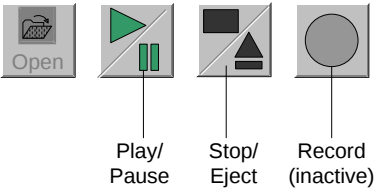


Figure 6 –Button status in “Playback” state.

 Université du Québec École de technologie supérieure	PROJECT	DOCUMENT NO.	DATE	VERSION
	LESIA	uObserve_Specs_Eng_A4for mat	2007-11-03	1.0
	TITLE	PAGE	PAGES	
	UOBSERVE SOFTWARE SPECIFICATION	10	16	

Start uSleuth

Use case title	Start uSleuth
Brief description	Starting uSleuth allows the UT to ensure that a camera is found and that the system may be ready to operate.
Preconditions	uSleuth is installed on the workstation.
Trigerring event	The UT starts the uSleuth application on the server workstation.
Main flow	1. The uSleuth window displays. 2. The log section displays empty. 3. uSleuth search for an installed camera. 4. uSleuth displays the live view of the camera found. 5. The status field displays "Waiting for uSpy to connect...". 6. The Open [file] button is activated.
Postconditions	The "Idle" state is displayed.

Close uSleuth

Use case title	Close uSleuth
Brief description	Closing uSleuth aims at closing open files properly, then closing the system.
Preconditions	uSleuth application is running.
Trigerring event	The UT closes uSleuth.
Main flow	When in Ready state: The status field displays "Closing..."; uSleuth closes any open file; The uSleuth window closes and uSleuth terminates.
Alternate flow	When in Record or Playback state: 1. The status field displays "Closing..."; 2. uSleuth stops recording audio, video, and log files; 3. uSleuth closes any open file; 4. The uSleuth window closes and uSleuth terminates.
Postconditions	The uSleuth application is closed.

 Université du Québec École de technologie supérieure	PROJECT LESIA	DOCUMENT NO. uObserve_Specs_Eng_A4for mat	DATE 2007-11-03	VERSION 1.0
	TITLE UOBSERVE SOFTWARE SPECIFICATION		PAGE 11	PAGES 16

Connect uSpy to uSleuth

Use case title	Connect uSpy to uSleuth
Brief description	By establishing the connexion between uSpy and uSleuth, the UT ensures that uSpy start sending all keyboard and mouse events.
Preconditions	• <u>Configuration of parameters</u>: the IP address of the uSleuth server will be kept in a configuration file • <u>LAN connecting</u>: The uSpy workstation and the uSleuth server shall be connected to the same LAN. • <u>Security access</u>: The uSpy workstation shall benefit from the necessary security access on the uSleuth server.
Trigerring event	The UT starts the application using uSpy on the client workstation.
Main flow	uSpy gets the server IP address and connects to it; uSleuth accepts the uSpy connection and displays "uSpy connected" in the status field; The Record button is activated.
Postconditions	The "Connected" state is displayed.
Extension point	<u>Disconnect uSpy from uSleuth</u> : This point start when someone closes or disconnects uSpy; • Current state is "Connected". uSpy sends a disconnect event to uSleuth; uSleuth stops its current activities and displays "Waiting for uSpy to connect..." in the status field; The following buttons are deactivated: Play, Stop, Record; • The "Idle" state is displayed.

Send events

Use case title	Send events
Brief description	uSpy send all keyboards and mouse events to uSleuth.
Actor	The user application, which includes the uSpy library.
Preconditions	Current state is "Connected". The participant [of usability testing] starts using the application.
Trigerring event	The current state has been changed from "Idle" to "Connected".
Main flow	While connected: 1. uSpy captures any [keyboard and mouse] event; 2. uSpy sends any captured event to uSleuth.
Postconditions	Events are sent to uSleuth.

 Université du Québec École de technologie supérieure	PROJECT LESIA	DOCUMENT NO. uObserve_Specs_Eng_A4for mat	DATE 2007-11-03	VERSION 1.0
	TITLE UOBSERVE SOFTWARE SPECIFICATION		PAGE 12	PAGES 16

Record an experiment

Use case title	Record an experiment
Brief description	Recording an experiment is mandatory if the UT desires to playback that experiment upon request to perform data analysis. An experiment is composed of the audio-video file and the log of events. The UT needs to keep and record a subset of all recieved events. To define that subset of relevant events, the UT had previously setup which event types are to be filtered out from recording them in the configuration file.
Preconditions	• <u>Configuration of parameters</u>: the list of event type to filter out of the event log is kept in a configuration file. • The system is in “Connected” state. • uSleuth recieves events from uSleuth.
Trigerring event	The UT presses Record
Main flow	While in “Record” state: uSleuth displays “Recording...” in the status field; The Stop button is activated; uSleuth filters out unwanted events from list of events to filter out; uSleuth calculates the delta-time in ms with the previous event; uSleuth calculates the duration in ms with the previous event of the same type; uSleuth displays each relevant event; uSleuth stores events in the log file; uSleuth records the active video file; The UT presses Stop; uSleuth stops recording audio, video, and log files; uSleuth displays the video file on the first image uSleuth displays the log file on the first event; uSleuth displays “Playback – Still” in the status field.
Alternate flow	When in “Ready” or “Playback” state: 1. uSleuth discards all received events.
Postconditions	The event log file contains all relevant events.

Experiment directory name structure

uObserve shall create one directory per experiment, in the directory from which it is operated.

All recorded experiment files (audio, video, log) shall be grouped in an experiment directory uniquely identified with the date and time when the recording was initiated. The directory name format could be yyyy_mm_dd@hh_mm_ss. Example: 2004_03_19@21_02_23.

Event log format

The event log shall be stored in Comma Separated Values (CSV) format to be able to load it in Excel.

 Université du Québec École de technologie supérieure	PROJECT LESIA	DOCUMENT NO. uObserve_Specs_Eng_A4for mat	DATE 2007-11-03	VERSION 1.0
	TITLE UOBSERVE SOFTWARE SPECIFICATION		PAGE 13	PAGES 16

Open a recorded experiment

Use case title	Open a recorded experiment
Brief description	Prior to playback a previously recorded experiment, the UT needs to open files from a specific experiment. Data is organized chronologically to ease browsing through experiment directories.
Preconditions	The system is in "Ready" state.
Triggerring event	The UT selects to Open a recorded file: • By pressing the Open icon; or • By selecting Open from the drop down File menu.
Main flow	uObserve opens a window to allow the user to select a recording directory: • The starting point is the root directory; The UT browses through directories and confirms recorded experiment selection; uObserve closes the file selection window; uObserve opens: • the selected audio-video file and • the corresponding event log file; uObserve displays events: • the current event is the first event; uObserve displays the first video frame as the current image.
Postconditions	The status is "Still".

 Université du Québec École de technologie supérieure	PROJECT LESIA	DOCUMENT NO. uObserve_Specs_Eng_A4for mat	DATE 2007-11-03	VERSION 1.0
	TITLE UOBSERVE SOFTWARE SPECIFICATION		PAGE 14	PAGES 16

Playback an experiment

Use case title	Playback an experiment
Brief description	
Preconditions	A recorded file has been opened [state is "Playback"/"Still"].
Triggerring event	The UT presses Play.
Main flow	uSleuth replaces the "Play" button by the Pause button; uSleuth activates the Stop button; uSleuth starts playing the current audio video file while displaying the synchronized log of events; - The status bar displays "Playing..."; - uSleuth highlights the current event; uSleuth stops playing when the current file reaches its end; uSleuth replaces the Stop button by the Eject button.
Alternate flow – Pause	During step 3 of the main flow, the UT presses Pause while playing: - uSleuth stops playing; - The status bar displays "Pause..."; The current image is the image that was displaying when Pause was pressed; - The current event is the last event encountered before Pause was pressed; - uSleuth replaces the Pause button by the Play button; - If the UT presses Play, continue from step 3 of the main flow; - If the UT presses Stop, continue from step 1 of alternate flow – Stop.
Alternate flow – Stop	During step 3 of the main flow or during alternate flow Pause, the UT presses Stop: - uSleuth stops playing; - The status bar displays "Still"; The current image is the first frame of the video file; - The current event is the first event of the log; - Continue from step 5 of the main flow.
Postconditions	- The status is "Still". - The current image is the last image of the video. - The current event is the last event of the log.

Seek from image

Use case title	Seek from image
Brief description	
Preconditions	A recorded experiment has been opened but it is not playing; State is Playback/Still.
Triggerring event	The UT clicks and drags the image cursor
Main flow	- uSleuth displays the image corresponding to the cursor position; - When UT releases the cursor, uSleuth synchronizes the current event in the event log corresponding to the last completed event at the current image time; - uSleuth refreshes the event log display in order to show the current event in the list window;
Postconditions	The status is "Still".

 Université du Québec École de technologie supérieure	PROJECT	DOCUMENT NO.	DATE	VERSION
	LESIA	uObserve_Specs_Eng_A4for mat	2007-11-03	1.0
	TITLE	PAGE	PAGES	
	UOBSERVE SOFTWARE SPECIFICATION	15	16	

Scroll and select from the event list

Use case title	Scroll and select from the event list
Brief description	
Preconditions	A recorded experiment has been opened but it is not playing; State is Playback/Still.
Trigerring event	The UT clicks and drags the event list cursor
Main flow	- uSleuth displays the portion of the event list corresponding to the cursor position; - The UT selects (clicks on) an event in the list; - uSleuth highlights the selected event as the current event; - uSleuth synchronizes the current event with the last corresponding video frame at the current event time; - uSleuth displays the current video frame.
Postconditions	The status is "Still".

Display "About uObserve"

Use case title	Display "About uObserve"
Brief description	Be able to confirm that the appropriate version of uObserve is running, as well as general information about the application.
Preconditions	uObserve is running.
Trigerring event	The UT selects "About uObserve" from the Help menu.
Main flow	- uObserve opens a window and displays the following: - The uObserve logo; "uObserve v X.Y", where X.Y represents the current software version; - The copyrights notice; - The following brief description: "uObserve is a usability testing tool providing record and playback of audio and video of a user in correlation with an event log." - An "OK" button. - The UT presses "OK". - The "About uObserve" window closes; - The uObserve main window refreshes.
Postconditions	uObserve is back where is was before that use case started.

Quality requirements

Reliability

No specific reliability feature shall be implemented in the event of a uObserve failure.

Availability

No specific availability feature shall be implemented within uObserve in the event of a LAN failure, i.e. if uSpy is unplugged during a recording, the event log transfer stops, and the current recording will have to be redone.

 Université du Québec École de technologie supérieure	PROJECT	DOCUMENT NO.	DATE	VERSION
	LESIA	uObserve_Specs_Eng_A4for mat	2007-11-03	1.0
	TITLE	PAGE	PAGES	
	UOBSERVE SOFTWARE SPECIFICATION	16	16	

Security

No specific security feature shall be implemented in uObserve. Data files security (audio/video files and event logs) shall be performed by operating system functions under which they are stored.

Maintainability

The Sun Microsystems Java coding standard shall be applied and systematically verified with the « CheckStyle » plug-in integrated to the development environment.

Automated unit testing (JUnit) shall be provided along with the Java code and ran as development progresses.

Portability

The uSleuth application shall only be used under the Windows operating system. The uSpy library shall be used under both the Windows and the Linux operating system.