This is a pre-print version of the paper published at the: **Journal of Systems and Information Technology, Vol. 12, No. 2, pp. 120-139, 2010**.

Combining Experimental and Inquiry Methods in software usability evaluation: the paradigm of LvS Educational Software

Nektarios Kostaras*, Dimitris Stavrinoudis, Stavroula Sokoli, Michalis Xenos

School of Sciences and Technology, Hellenic Open University, 13-15 Tsamadou str, Patras, 26222, Greece.

* Corresponding author. Tel.: +30 2610 367405; fax: +30 2610 367416.
E-mail address: nkostaras@eap.gr

Abstract

Purpose - The purpose of this paper is to present a methodology combining experimental and inquiry methods used for software usability evaluation. The software product of LeViS project funded by the European Commission (Socrates/Lingua II) is used as an evaluation paradigm. The aim of the paper is twofold: a) to present the results of the usability evaluation using this software as an example and to suggest a number of improvements for the next version of the software tool; and b) to portray the advantages of combining methods from different evaluation approaches and the experiences from their application.
Design/methodology/approach - The evaluation for this experiment combined different usability methods, both experimental and inquiry ones. More specifically the methods employed were the Thinking Aloud Protocol and the User Logging, which were performed in a usability evaluation laboratory, as well as the inquiry methods of Interview and Focus Group.
Findings - In this study, usability problems regarding the LvS educational software were revealed as well as issues regarding the use of Thinking Aloud Protocol method and involving users with a specific profile. The research findings presented in this paper constitute an innovative and effective methodology for software usability evaluation and are useful for laboratories aiming to conduct similar evaluations.
Research limitations/implications - Although this methodology has been successfully applied for over 20 software products, due to practical purposes related to this paper's extent, only one software is used as an example.
Originality/value - Through the evaluation process, apart from discovering certain usability problems related to the software, there are a number of important conclusions drawn, regarding the methods used and the methodology followed in software usability evaluation.
Keywords Software Quality, Usability Evaluation, Usability Evaluation Methods, Quality Assessment Laboratory, Subtitling, Learning Activities
Paper type Research paper

1. Introduction

The development of effective interactive software involves substantial use of evaluation experiments throughout the process (Sharp et al., 2007, Schneiderman, 1998). Usability evaluation is an increasingly important part of the user interface design process. However, usability evaluation can be expensive in terms of time and human resources, and automation is therefore a promising way to augment existing approaches. According to ISO9241-11 standard (ISO9241-11, 1997), usability is the extent to which a computer system enables users, in a given context of use, to achieve specified goals effectively and efficiently while promoting feelings of satisfaction.

Usability evaluation consists of methodologies for measuring the usability aspects of a system's user interface and identifying specific problems. In other words, it is an important part of the overall user interface design process, which consists of iterative cycles of designing, prototyping, and evaluating (Dix et al., 2004, Nielsen, 1993). It is a process that entails many activities depending on the methods employed.

A wide range of usability evaluation methods and techniques have been proposed, and a subset of these is currently in common use. Some of them, such as formal user testing, can only be applied after the interface design or prototype has been implemented. Others, such as heuristic evaluation, can be applied in the early stages of design. Furthermore, usability findings can vary widely when different evaluators study the same user interface, even if they use the same evaluation technique (Jeffries et al. 1991, Molich et al. 1999).

This paper presents the usability evaluation of the Learning via Subtitling (LvS) tool, a subtitling simulator developed within the framework of the LeViS project funded by the European Union (Socrates/Lingua II). More specifically, it presents the experimental plan and the results of the evaluation. The main goals are to compare the results derived from the different evaluation methods, to investigate the possible correlation and overlapping between them and to propose cases where these different methods may be combined. In this experiment both experimental and inquiry evaluation methods were used. The work presented in this paper was conducted by the Software Quality Research Group (SQRG, 2009) of the Hellenic Open University (HOU, 2009).

In the next section the most popular evaluation methods are reported and categorized. Section 3 presents the Learning via Subtitling project and the LvS software. In section 4 the experimental plan of the reported evaluation is analyzed, whereas in section 5 the results of this experiment are demonstrated. Section 6 presents the corrected version of the LvS software after its usability evaluation. Finally, section 7 summarizes the conclusions reached as a result of this research.

2. Evaluation methods

The evaluation methods can be generally divided into analytic and empiric ones (Nielsen, 1993). The analytic methods are theoretical models, rules or standards that simulate user's behavior. They are mainly used during the requirement analysis phase and usually even before the development of the prototypes of a product. As a result, the users' participation is not required in these methods. On the contrary, the empiric methods depend on the implementation, the evaluation and the rating of a software prototype or product. In this rating it is necessary for the participation of a representative sample of the end-users or/and a number of experienced evaluators of the quality of a software product. The empiric methods can be further divided into experimental and inquiry ones.

Experimental methods require the participation of the end-users in a laboratory environment and the most known are the following (Avouris, 2001):

- Performance measurement. It is a classical method of software evaluation that provides quantitative measurements of a software product performance when users execute predefined actions or even complete operations.
- Thinking aloud protocol. This method focuses on the measurement of the effectiveness of a system and the user's satisfaction. According to this method, users interact with the system, while they state aloud their thoughts, opinions, emotions and sentiments regarding the system.
- Co-discovery (Kennedy, 1989). This is a type of usability testing where a group of users attempt to perform tasks together while being observed, simulating typical work process, where most people have someone else available for help.
- User actions logging. There are many techniques to record the actions of users while they interact with a software product. The most common are note taking, voice recording, video recording, computer logging and user logging.

Inquiry methods focus on the examination of the quality characteristics of a software product by measuring users' opinion. The most popular are the following (Stavrinoudis et al., 2005):

- User questionnaires. In this method, users are requested to express their opinion about the quality of a software product by completing a structured questionnaire usually consisting of multiple-choice questions. The use of questionnaires provides valuable feedback and obtains answers to specific questions from a large group of people, especially in the case that the target group is spread across a wide geographical area (Sharp et al., 2007).
- User interviews. This is a structured method of evaluating a software product, where the researcher is in direct contact with the user. The questions of the interview follow a hierarchical structure, through which the general opinion of the product is first formed after which more specific matters of the quality characteristics are considered.
- Focus groups. This method is a variation of the previous one, where a group of about 10 users is formed under the supervision of a coordinator, who is in charge of the topics of conversation. At the end of this conversation, the coordinator gathers their conclusions on the quality of the software product.
- Field observation. With this method, the researcher observes the users at their working place, while they are using and interacting with the software product.

Usability testing is a method that is conducted in a controlled environment, such as a usability laboratory or similar, where the performance of the participants on predefined scenarios involving the software under assessment is measured (Sharp et al., 2007). In this method, participants evaluate the extent to which a software system meets specific usability criteria. The range of tests that can be conducted by someone is considerable, from true classic experiments with large sample sizes and complex test designs, to informal qualitative studies with a single participant. Each testing approach has different objectives, as well as different time and resource requirements (Rubin, 1994).

Data from Usability testing can be acquired by measuring participants' performance on specific tasks. This is achieved by recording the participants' actions and analyzing the aggregate data to determine whether the product is efficient and effective.

An acceptable number of participants in Usability Testing, according to Dumas and Redish (Dumas and Redish, 1999) is considered to be five to twelve, but sometimes it is possible to use fewer when there are budget and schedule constraints. Nielsen and Lander (Nielsen and Landauer, 1993) suggest that for observational studies, where the aim is simply to uncover usability issues, there is no need to employ more than five participants. If the intention is to discover much more about the extent of usability problems performing statistical analysis of the results, at least twice this number is recommended. Finally, Rubin (Rubin, 1994) proposes at least eight participants. Considering the different suggestions about the number of the users that should participate in usability testing, we conclude that, depending on the case, a number of five to twelve is adequate.

3. Presentation of LvS software

In the so-called "subtitling countries" viewers are exposed to subtitled foreign films or TV programmes from a very young age. Given that this exposure is regarded to promote language learning, teachers have exploited various kinds of audiovisual material in the Foreign Language classroom.

It has also been observed that students of translation attending subtitling courses have improved their linguistic skills. However, only professional subtitling tools have been used and no subtitling software has been designed specifically for language learning - with all the shortcomings this entails.

Advances in Information and Communication Technologies provide new opportunities for the exploitation of subtitling in language teaching and learning, namely the development of a subtitling software for active learning task-based activities. This tool is intended to give learners the chance to use a special version of a professional environment, not for the purposes of training but for its side benefits.

The main focus of Learning via Subtitling project (http://levis.cti.gr) is the development of educational material for active foreign language learning based on film subtitling. It aims to cover the exigency for active learning where cultural elements are involved effectively through real-life (simulated) activities and the need for productive use of multimedia not as a nice add-on but as the core of an activity. This project was funded by the European Union from the Socrates Programme - Lingua 2 Framework (Development of Language Tools and Materials) and was coordinated by the Hellenic Open University. The project's main pedagogical and technical objectives are:

- To develop educational material for foreign language learning based on subtitling.
- To engage European university tutors in developing material for learning Greek, Hungarian, Romanian, Portuguese and Spanish as foreign languages.
- To utilize the learning material in actual university courses.
- To evaluate the material from the point of view of both the tutor and the student.
- To analyze the process of developing learning material and to propose an explicit best-practice implementation roadmap.
- To disseminate the results of the project.

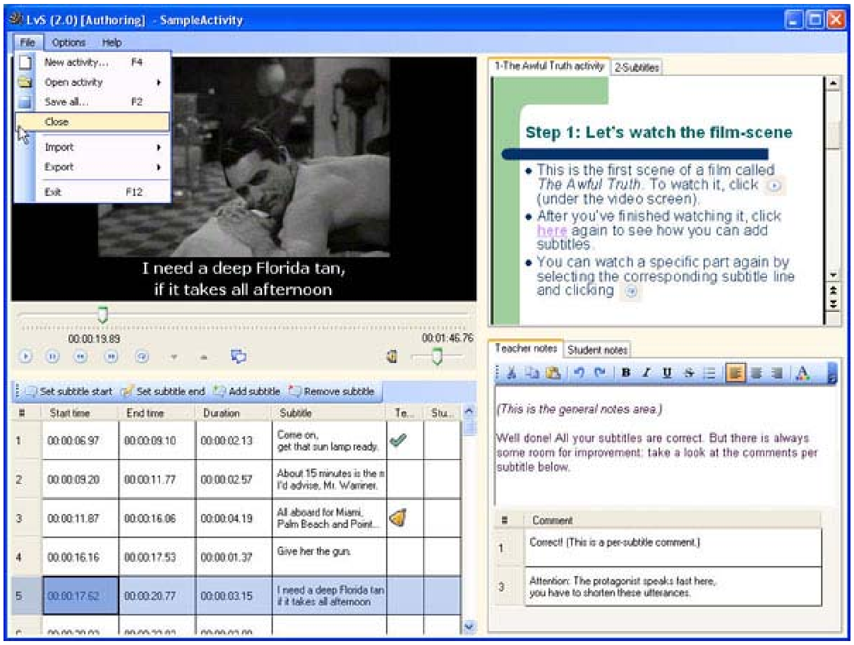
A subtitling simulator called LvS is actually a tool that has been designed for educational activities' purposes in language learning. Through this tool and activities, the learner is asked to add subtitles to a film thus engaging in active listening and writing tasks.

The flexibility in the use of LvS is evidenced in that it can be utilized in any real or virtual classroom and within any curriculum, as it does not imply any change in the methodology used. LvS is also adequate for use in autonomous learning environments: the application's main screen includes a document viewer area, where all the necessary steps for self-study can be provided. Moreover, it may be employed for any number of students, with unlimited choice of video content (film scenes, educational material), for any suitable duration of video segment, student level (beginners, intermediate, advanced), age and interests.

Teachers can use the authoring mode of this software tool to create activities based on subtitling for film-scenes, news, documentaries etc. Learners, on the other hand, can employ it to carry out tasks ranging from filling in the gaps to placing mixed subtitles in the correct order, and from transcribing the original utterances to translating them and creating new subtitles.

Figure 1 presents a typical screenshot of an LvS learning activity. The LvS main screen is divided in four basic areas:

- The video player area allows the learner to view, rewind and forward the film, with or without subtitles.
- The Document viewer area allows the learner to view the instructions and other files necessary for the activity (information about the clip, the script, exercises, etc.)
- The Subtitle editor area allows the learner to edit and manage the subtitles. Each subtitle line is divided in four columns where the subtitle's data is viewed: Start time and End time (the temporal points in the clip when the subtitle text appears on the screen and disappears), Duration, and Subtitle text. The next two columns can be used for teacher and learner comments. The teacher can mark the subtitle line with an icon ("well done", "warning" etc.) which when clicked takes the student to the Notes area.
- The Notes area allows the learner and the teacher to exchange feedback. It is divided in general notes and comments per subtitle.



#	Start time	End time	Duration	Subtitle	Te...	Stu...
1	00:00:06.97	00:00:09.10	00:00:02.13	Come on, get that sun lamp ready.	✓	
2	00:00:09.20	00:00:11.77	00:00:02.57	About 15 minutes is the n I'd advise, Mr. Warriner.		
3	00:00:11.87	00:00:16.06	00:00:04.19	All aboard for Miami, Palm Beach and Point...		
4	00:00:16.16	00:00:17.53	00:00:01.37	Give her the gun.		
5	00:00:17.62	00:00:20.77	00:00:03.15	I need a deep Florida tan if it takes all afternoon		

Step 1: Let's watch the film-scene

- This is the first scene of a film called *The Awful Truth*. To watch it, click (under the video screen).
- After you've finished watching it, click here again to see how you can add subtitles.
- You can watch a specific part again by selecting the corresponding subtitle line and clicking

(This is the general notes area.)

Well done! All your subtitles are correct. But there is always some room for improvement: take a look at the comments per subtitle below.

#	Comment
1	Correct! (This is a per-subtitle comment.)
3	Attention: The protagonist speaks fast here, you have to shorten these utterances.

Figure 1. Screenshot of an LvS learning activity

It is becoming increasingly evident to foreign language teachers and researchers that there is no one and only guaranteed teaching method and that a variety of technical and methodological resources are needed in the classroom. It has to be clarified that Learning via Subtitling that has not been designed as a learning method to substitute existing teaching methods, but rather as an additional activity to enhance them. Moreover, the LvS tool and its learning activities are intended to be used certain times throughout a learning course and not necessarily on a regular base.

Since the amount of usage of this tool may be limited, it is obvious that students must not spend excessive time in experiencing and understanding its functionality. On the contrary, they must focus on learning the foreign language and complete their activity. Furthermore, the LvS tool may be used by teachers and students with low level of experience in computer applications. As a result, the usability of this software has to be very high. In other words, according to the ISO 9126 standard (ISO9126, 2001), the user's effort for recognizing the underlying concept of this software and for learning how to use the software must be low. Moreover, this software has to be attractive and easily operated.

4. Usability evaluation of LvS

In order to develop a user friendly final version of the subtitling simulator, an appropriate usability evaluation of a prior version was necessary. Within the Learning via Subtitling project's framework, the main purpose of this evaluation was to predict the expected performance of an actual user, using the current software and to locate and fix serious problems prior to release. As a result, we evaluated the usability of the LvS software in order to examine a) how long does it take for the user to get accustomed to the software, b) if the instructions given in the software are clear and understandable and c) how simple it is for the user to navigate through the interface of the software, paying special attention in the extent to which a user can understand the purpose of each button and the use of each field.

4.1 Usability laboratories

Usability testing is conducted in controlled environments. A controlled environment is typically a usability laboratory where the performance of users in preplanned tasks is measured (Sharp et al., 2007). The aim of this process is to assess whether the product under evaluation is usable by the intended user group to achieve the tasks for which it was created (Dumas and Redish, 1999). Quantitative performance measures are obtained during the tests that produce the following types of data (Wixon and Wilson, 1997):

- Time to complete a task.
- Time to complete a task after a specified time away from the product.
- Number and type of errors per task.
- Number of errors per unit time.
- Number of navigations to online help or manuals.
- Number of users making a particular error.
- Number of users completing a task successfully.

There are various settings of usability laboratories. Generally the facilities of a usability evaluation laboratory comprises a main testing room, which is equipped with recording tools and in which the software under assessment is installed, and an observation room where the evaluators analyze the data collected during the usability assessment tests. A usability evaluation laboratory may also include a reception area for the participants, a storage area and a viewing room for observers. The space arrangement of the laboratory may vary according to the product under evaluation, so as to mimic features of the real world.

The different laboratory arrangements may range from extremely simple, low cost to more sophisticated expensive ones (Rubin, 1994). These laboratories can be single-room, double-room or portable.

Single-room setup hosts evaluators and participants in the same room. The evaluator can watch the participant, during the test, directly or indirectly through a workstation which is connected with a camera.

In the double-room setup, the evaluator is physically separated from the participant. In this setup one room serves as an observation area, where the evaluators are located and the other as a testing area where the participants perform the testing activities. The rooms can be in completely different areas of the building or adjacent to each other, in which case the rooms can be separated by a one-way mirror where the evaluator can observe the participant but not vice-versa.

Portable Laboratory does not occupy any room for testing purposes. It consists of suitable testing equipment such as cameras, microphones, laptops and is transported to different sites where the actual product under evaluation is installed.

The Software Quality Assessment (SQA) laboratory, in which the usability evaluation of the LvS software took place, is a suitably equipped laboratory for the performance of usability tests on various software systems. Figure 2 presents the setting of the SQA laboratory.

The SQA laboratory comprises two separate areas. One area is designated as the testing room and a second one is designated as an observation and control room. These two areas are separated by a one-way mirror so that the evaluator leader can observe the participants interact with the software system under evaluation but not vice versa. The maximum number of individuals inside the testing room is two participants and an evaluator leader, depending on the case of study. In the observation area two to three evaluator leaders are situated viewing, commenting and recording the proceedings. Outside the laboratory is a large room that serves as a reception area for the participants.

The equipment that was used in the testing room for the reported experiment is the following:

- One CCTV Roof Camera for the recording of the evaluators' behaviour during the assessment.
- A microphone for the collection of the evaluators' comments.
- A desktop computer where the evaluators interacted with the LvS software.
- Appropriate hardware used for the management and recording of the evaluators' screen throughout the assessment.

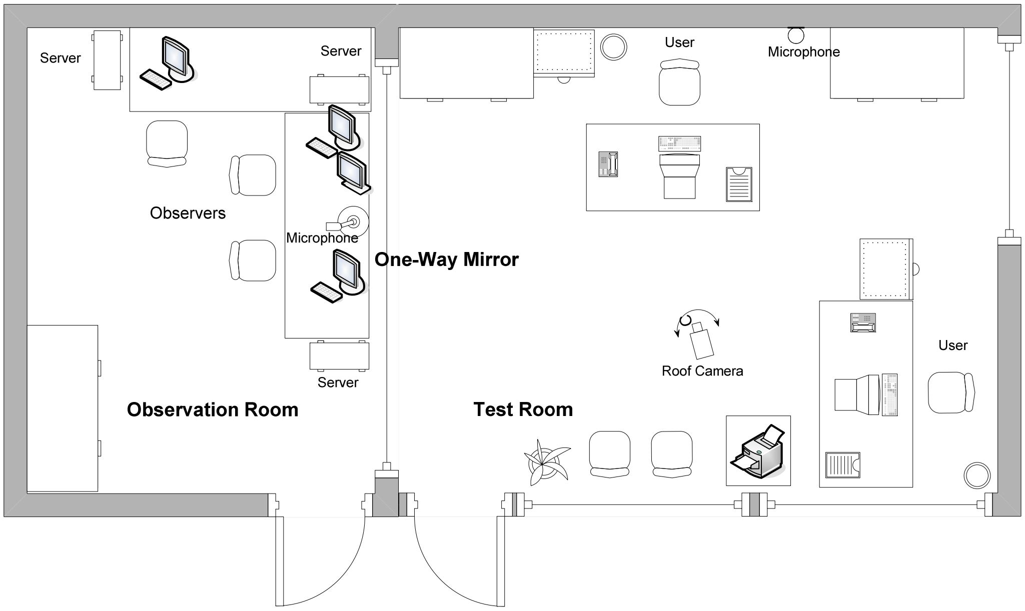


Figure 2. Software Quality Assessment laboratory

Moreover, the equipment in the observation room is:

- A specially set up server with double screen which is the control center of the laboratory and which contains software for the observation, recording and the processing of the assessment data.
- A console that controls the CCTV camera in the Test area.
- A microphone for the collection of evaluation leaders' comments.
- A file server for data storage.

A significant feature of the SQA laboratory is its set up. In the laboratory, the evaluator leaders are not in the same room with the evaluators. This eliminates any possible bias due to inadvertent non-verbal communications or mannerisms almost entirely. The evaluation room is arranged in such a way that gives the participant the feel of an ordinary office working environment and not a laboratory. Furthermore, the camera and the microphone are placed in such a way that are not easily noticeable, despite the fact that the evaluators are informed for their existence before the beginning of the assessment.

4.2 Participants of the experiment

A total of 11 users (5 male and 6 female), that were over the age of 20, participated in the usability evaluation of the LvS, in the SQA laboratory. As far as the participants' background is concerned, regarding computer expertise, 3 of them were experts in computers 5 of them were very experienced whereas 3 of them were relatively less experienced users. It should also be mentioned that, none of the total 11 users had ever participated in a usability evaluation of a software product before. Moreover, none of them had any subtitling experience or ever used a software similar to the LvS tool.

Before evaluating the LvS, the users had already attended a training seminar conducted by the HOU under the framework of the DIEPTELO's network (http://dieptelo.csd.auth.gr/). DIEPTELO was a research-training network which aimed to spread the technical-theoretical knowledge in major Computer Science areas among researchers and professionals.

During this HOU seminar the participants attended a number of training courses about, among others, software development methods, project management methods, software testing techniques and obviously software quality assurance methods focusing mainly on usability evaluation methods. After this course, the users were asked to participate in the usability evaluation of the LvS (version 2.0). The evaluator leaders explained to them that the objective of this experiment was to evaluate the usability and not the educational aspects of LvS. The participants were not asked to assign a simple mark to the usability factor of the software, but to explore and find possible usability faults and to suggest ways to develop a user friendly interface.

4.3 Experimental plan

For the usability evaluation of the tool, methods of two different categories were used; experimental and inquiry methods. In detail, experimental methods on the one hand involved the observation of individual users performing specific tasks with the system under evaluation. On the other hand, in inquiry methods users provide feedback on an interface via interviews, surveys, etc. (Ivory and Hearst, 2001).

Regarding the experimental methods, the users performed representative tasks, under the discrete attendance of usability experts (the personnel supporting the experiment). In the reported evaluation the experimental methods used were a) Thinking Aloud Protocol, in which the users expressed verbally their thoughts, feelings, opinions and specific actions performed while interacting with the system and b) User Logging, where users' activities were recorded with the use of special equipment such as cameras, microphones and specialized software. In addition to the experimental methods, two different inquiry methods were used: a) Interview, with the use of a questionnaire formed by open and closed questions and b) Focus Group, which was a moderated discussion among all the participants.

As far as the experimental plan is concerned, the experiment involved the following four steps. Firstly, the experiment personnel welcomed each participant and made sure that they feel comfortable.

The second step was the orientation phase which lasted approximately 15 minutes. The duration of the orientation phase was limited to 15 minutes, so as to simulate real classroom situations, were the students must not spend excessive time in understanding the functionality of the tool. During this phase the participants received a short demo presentation introducing the software under evaluation. The purpose and objective of the experiment was also presented, as well as some indications about what the participants were expected to do. Specifically, they were assured and reminded that the purpose of the evaluation was the usability of the software and not their personal performance. Therefore, they were encouraged to act in a way that is typical and comfortable to them. Finally, participants were informed that they were being observed and all their actions and comments were recorded with the use of cameras and microphones.

Figure 3. Participant in the test area

Figure 4. Observers in the control room

The third step was the experimental usability evaluation, during which the participants were asked to perform a number of predefined tasks while they were being observed. The scenario stood as follows: The participants were asked to enter the experiment area of the SQA laboratory (one at a time) and sit down at a desk where a PC was placed with the LvS software installed. Then, they were asked to start the software and perform a number of predefined tasks which were written in a piece of paper situated on the desk. It should be noted in this point that the tasks performed were common to all participants. Evaluation had to take not more than 20 minutes for each user. This limitation was set, so as to simulate real classroom situations, where 20 minutes form an average time required for such activities. After all tasks were completed or the predefined time expired the participant left the experiment room and returned to the reception area, whereas a next participant was asked to enter the laboratory. Figure 3 is a photo of a participant interacting with LvS in the test area of the SQA laboratory, whereas figure 4 is a photo of the observers in the control room.

In brief, the predefined tasks that the participants were asked to complete were the following:

- Firstly, they had to double-click on the LvS Activity icon to start the interaction with the software interface.
- Then, they had to read and follow the instructions positioned on the top right corner of the software interface.
- After they had read the instructions, participants were expected to start the video and either complete as many subtitles as possible in the given time or at least 6 subtitles.
- Finally, they had to save their activity to the hard disk of the PC.

All the above actions were recorded for each user. During the interaction with the LvS, participants were asked to verbalize their thoughts and emotions, as well as to explain their actions when problems were occurring.

The main objective of the aforementioned tasks was, to observe how easy or not is for the users to interact with the software and complete a common task, having attended an approximately 15 minutes tutorial and having read the instructions, regarding software’s function. More specifically, the time it takes for the user to get accustomed to the software, the easiness of a user to navigate through software’s interface and the extent to which provided instructions are clear and understandable.

The fourth step comprised the interview and the focus group sessions, which were the inquiry evaluation methods that were chosen for this experiment. The interview session, took place in the reception area and each of the participants was interviewed for approximately 20 minutes. The interview session was based on a sort questionnaire (Appendix) regarding demographic data and the subjective perception of usability and aesthetics of the software. More specifically, the questionnaire was separated into four sections, including open and closed questions. The aim of the first section was the collection of basic demographic information regarding the status of the participants. Section B included, one open and three closed questions, which were used to determine participants’ foreign languages knowledge and computer skills. Section C included questions which were measured on a five-point Likert scale whereby 1 represented “strongly disagree” and 5 “strongly agree”. These questions were used to determine the usability of the interface. Finally, section D included open questions aiming to gather information regarding participants’ perception on technical issues and usability faults, as well as their opinion on features that should be added or removed from the software’s interface.

During the focus group session, which lasted approximately 40 minutes, the participants under the supervision of a coordinator discussed all together about the usability of the LvS software and were free to ask any questions relevant to the experiment. Figure 5 is a photo of this session. Finally, after finishing, they were thanked for their time, effort and contribution to this experiment.

Figure 5. Focus group session

5. Findings of the experiment

In this experiment 11 users evaluated the usability of LvS version 2.0, which was a working release of this software already used by the Learning via Subtitling project members to create the first learning activities. Thus, it was not expected to find many usability faults by the reported evaluation. It has to be noted though that, since the 11 users had attended the course in software quality assurance methods, they faced their participation to this experiment as a way to test and prove their capabilities in evaluating the usability of a software product. As a result, they appeared overeager to find usability faults in LvS.

The evaluation measures that were collected and calculated in the laboratory are the following:

- The time it took participants to perform the tasks or the number of subtitle lines completed in the predefined time.
- The percentage of participants who completed subtitle lines successfully.
- Count of all incorrect selections for each user.
- The time users spent to read the instructions.
- The time each participant needed to start writing the first subtitle.
- Count of the number of times a user accessed the instructions after they started writing the first subtitle.
- Count of negative comments or facial expressions and body language.

5.1 Outline of the evaluation results

The evaluation results revealed that 10 participants managed to finish the predefined tasks within the given time, whereas one failed and was stopped when the 20 minutes of the experimental evaluation completed. During the interaction with the software only 4 out of 11 participants actually read the instructions provided in the LvS, although they faced some problems while completing their tasks. When facing a technical difficulty most of the users tried firstly to solve it on their own and then maybe refer to the provided instructions. Two of them preferred to give up and move on to the next task. Four of them didn’t need to read these instructions, because they didn’t face any problem or, as they said, didn’t reach such a level of desperation so as to look at them. Finally, two of them couldn’t overcome the difficulties they faced despite reading the instructions, because they didn’t understand them clearly. The usability faults reported by the participants during the inquiry methods or observed by the evaluator leaders when examining the results from the experimental methods are described in the following paragraphs.

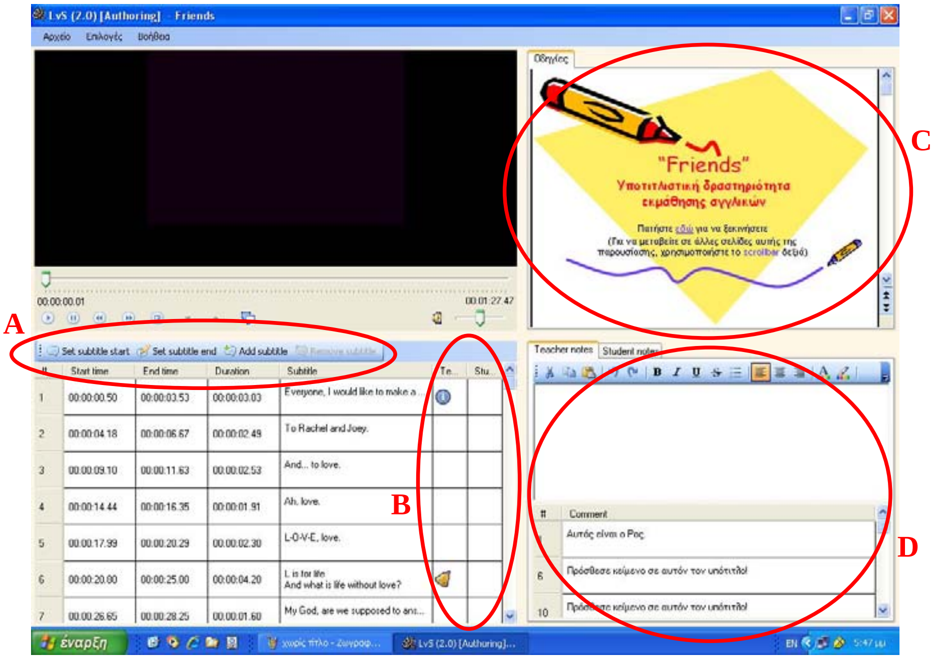
First of all, at the initial screen of the tool all the users delayed to start interacting with it. The splash screen of the application, covered the first dialog box. Most of the users were searching for several seconds how to start their activity or even thought that it would start automatically.

It was also reported that the "add subtitle" toolbar (figure 6, area A) was not immediately visible. Many users spent a lot of time searching for it, since they thought that this line was just part of the background. They would have preferred a bigger toolbar with more visible choices like buttons on it, clearly separated from each other. Moreover, it was difficult for some of them to find the “remove subtitle” choice. When no subtitle was selected, this choice was inactive, but because of the color chosen for the toolbar, these users couldn’t notice that this choice actually exists. In other words, this choice appeared to be rather hidden than inactive. As far as the grid below that toolbar is concerned (figure 6, area B), the headers of the last two columns “Teacher” and “Student” are not legible. Users didn’t manage to resize them, so they requested that these column headers must be entirely visible.

The users expected that the text boxes of the subtitles would support keyboard actions in a certain way. For example, by pressing the “tab” button while editing a subtitle, the focus would move to the next line. Or by pressing the “enter” button the focus would move to the next subtitle.

While writing in the text box of a subtitle, if its text size exceeds the 39 characters, the color of these characters turns to red, revealing that this subtitle is too long for the specific time allocated to it. Many participants couldn’t understand what this actually means, since no one of them had any experience in subtitling. They proposed that the LvS tool should better inform them in that case with an appropriate message, or even that the edited text should automatically continue in the next line of the subtitle.

The instructions for carrying out the learning activity in the LvS interface (figure 6, area C) were in a PowerPoint format and could be accessed only sequentially. Thus, a user could not refer automatically and rapidly to the topic that he was interested in. The set of these instructions should be provided in a different way, beginning with a brief index as a table of contents and supporting hyperlinks inside them to make the navigation easier when a user searches for a specific topic.



#	Start time	End time	Duration	Subtitle	Te...	Stu...
1	00:00:00.50	00:00:03.53	00:00:03.03	Everyone, I would like to make a ...	●	
2	00:00:04.18	00:00:06.67	00:00:02.49	To Rachel and Joey.		
3	00:00:09.10	00:00:11.63	00:00:02.53	And... to love.		
4	00:00:14.44	00:00:16.35	00:00:01.91	Ah, love.		
5	00:00:17.99	00:00:20.29	00:00:02.30	L-O-V-E, love.		
6	00:00:20.00	00:00:25.00	00:00:04.20	L is for life And what is life without love?	◀	
7	00:00:26.65	00:00:28.25	00:00:01.60	My God, are we supposed to ans...		

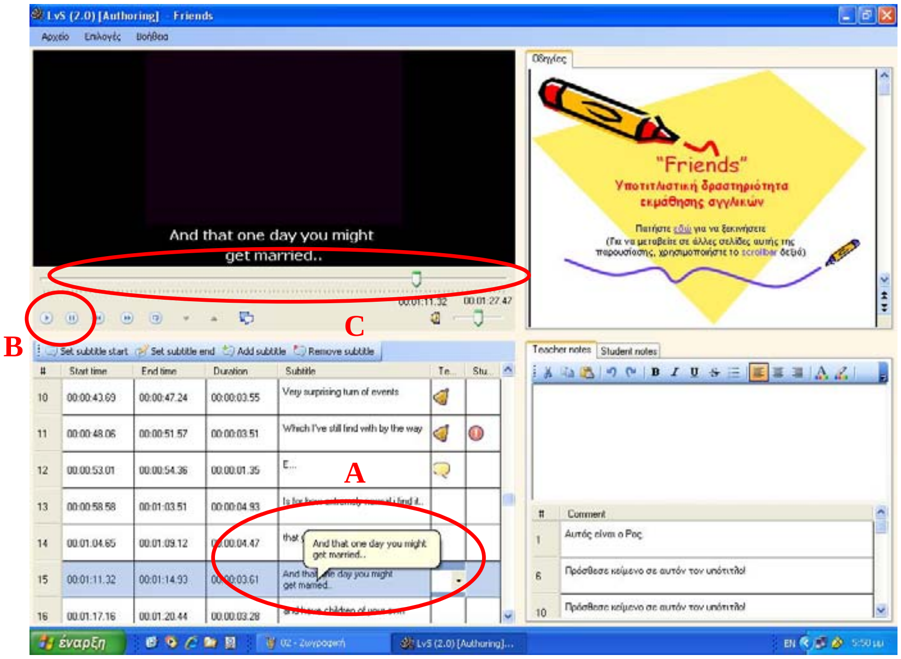
#	Comment
1	Αυτός είναι ο Ρος
6	Πρόσθεσε κείμενο σε αυτόν τον υπότιτλο!
10	Πρόσθεσε κείμενο σε αυτόν τον υπότιτλο!

Fig.6 Screenshot 1 of the LvS tool

The comments and the notes area (figure 6, area D) were not clear for two users. In order to insert a comment they used the notes area instead. According to their suggestions, it is preferable to have the comments area on top and the notes on bottom. However, all the rest participants found no difficulty according to this matter.

While interacting with the software, the usage and the appearance of the tooltips (bubbles) was not considered appropriate. For example, a bubble appears when a user enters the text box of a subtitle and while he edits the subtitle, this text appears into the bubble as well (figure 7, area A). All the participants reported this occurrence as irritating, especially when this bubble covered the subtitle text. Moreover, no tooltip appears when a user moves the mouse over a word which is not entirely visible, such as the column headers "Teacher" and "Student" of the grid (figure 6, area B).

Many users proposed that the video toolbar should not have two different buttons "play" and "pause" (figure 7, area B). Instead, these two buttons should be joined in one, according to the philosophy of commonly used video or audio players, such as Windows Media Player, Real Player etc. According to the users, this joining will reduce the necessary mouse movements, allowing the user to pay more attention to the timing of the subtitles. Moreover, one user expected that he could use the spacebar from the keyboard as a shortcut to these buttons, in order to start and pause watching the video.


LvS (2.0) [Authoring] - Friends
Αρχείο
Επιλογές
Βοήθεια
And that one day you might get married..
00:01:11.32
00:01:27.47
B
C
A
Set subtitle start
Set subtitle end
Add subtitle
Remove subtitle
Οδηγίες
"Friends"
Υποτιτλιστική δραστηριότητα εκμάθησης αγγλικών
Πατήστε εδώ για να ξεκινήσετε
(Για να μεταβείτε σε άλλες σελίδες αυτής της παρουσίασης, χρησιμοποιήστε το scrollbar δεξιά)
Teacher notes
Student notes
And that one day you might get married..
έναρξη
02 - Ζωγραφική
LvS (2.0) [Authoring]...
EN
5:50 μμ

#	Start time	End time	Duration	Subtitle	Te...	Stu...
10	00:00:43.69	00:00:47.24	00:00:03.55	Very surprising turn of events		
11	00:00:48.06	00:00:51.57	00:00:03.51	Which I've still find with by the way		
12	00:00:53.01	00:00:54.36	00:00:01.35	E...		
13	00:00:58.58	00:01:03.51	00:00:04.93	Is for how extremely normal i find it..		
14	00:01:04.65	00:01:09.12	00:00:04.47	that		
15	00:01:11.32	00:01:14.93	00:00:03.61	And that one day you might get married..		
16	00:01:17.16	00:01:20.44	00:00:03.28	and have children of your own		

#	Comment
1	Αυτός είναι ο Ρος
6	Πρόσθεσε κείμενο σε αυτόν τον υπότιτλο!
10	Πρόσθεσε κείμενο σε αυτόν τον υπότιτλο!

Fig.7 Screenshot 2 of the LvS tool

The participants noticed that the time bar of the video (figure 7, area C) has a short lack of accuracy. In particular, when moving the time bar it was very difficult to find a desired second or a specific frame. As a result, they proposed that the next version should offer a more accurate navigation bar for the video. Moreover, many users tried to double-click on a subtitle expecting that the part of the video referring to the timing of this subtitle would be shown. However, this functionality was not available in the LvS tool.

Finally, the great majority of the participants discovered a fault that was rather a software error (a bug) than a usability fault. This error had to do with the problem of overlapping subtitles, when the start time of a subtitle was equal or less than the end time of a previous subtitle. If this problem occurred, the user couldn't edit the overlapping subtitle. The only way to confront this problem was to delete the subtitle and then inserting it again correctly. Although this is not actually a usability fault, it affected the participants' opinion in the usability factor of the LvS tool.

Besides the usability problems mentioned above, the participants also stated a number of expectations and suggestions for the new version of the LvS. For example, they preferred to have the video stopped when they clicked on a subtitle. They proposed the "move subtitle" and "copy subtitle" functions to be added. Finally, they suggested that a speech recognition utility would improve the teacher's preparation of the learning activity, whereas an available spellchecker inside the LvS tool would be remarkably useful for the students.

5.2 Comparison and combination of the results

As previously mentioned, both experimental and inquiry methods were used for the usability evaluation of the LvS tool. The purpose of this combined approach was not only to discover more usability faults than if only one method was followed, but also to compare the results derived from the different evaluation methods and to investigate the possible correlation and overlapping between them.

First of all, it must be mentioned that most of the usability faults were found and reported by all the evaluation methods that were used. However, specific faults were discovered only by one method, either experimental or inquiry one. In detail, the problem with the long duration of the splash screen was revealed only by the User Logging method, since it was not mentioned during the inquiry methods. Another usability fault discovered only by experimental methods was when users tried to double-click on a subtitle expecting that the part of the video referring to the timing of this subtitle would be shown.

On the contrary, the limited operability and attractiveness of the instructions for carrying out the learning activity in the LvS interface were revealed only by inquiry methods. While examining only the experimental methods' results, we couldn't notice the actual reason that participants didn't use the provided instructions. However, during the interview and focus group sessions they mentioned that this function was not very usable, because instructions were in a PowerPoint format and could be accessed only sequentially. Moreover, the problem with the limited accuracy of the time bar was also revealed only by inquiry methods. During the analysis of the experimental methods' recordings we initially assumed that participants were moving the time bar so often in order to ensure that the inserted subtitles were displayed correctly. We couldn't notice that it was difficult for them to find a desired second or a specific frame.

Regarding the problems that we faced, we must mention that the recordings of the Thinking Aloud Protocol method didn't provide satisfactory data to analyze in combination with users' actions. Participants were asked to express verbally their thoughts feelings and opinions while interacting with the system. Unfortunately, only 3 of them and only in particular points of the evaluation actually followed these guidelines, whereas the rest of the participants preferred to remain silent. As a result, during the analysis of these recordings separately from the inquiry methods' results, it was difficult and in some cases impossible to interpret their reactions during the experimental usability evaluation. However, after examining the participants' responses in the interview session, the comprehension and the analysis of the experimental methods' results were feasible and efficient.

All the users that participated in this experiment had no previous experience in any experimental usability evaluation of a software product. Furthermore, almost all of them were ignorant about the Thinking Aloud Protocol method and its requirements from the participants. Hence, in spite of the training course about usability evaluation methods they had already attended and the specific guidelines that were given to them, they found it difficult or even odd to express their feelings loud during the evaluation, simply because they were on their own. If the participants were more experienced in this method, the data derived from this experimental evaluation would be more useful and could be directly analyzed, without the assistance of inquiry methods. Another solution to this problem was to conduct further evaluation seminars to the participants, prepare extra evaluation tests and give them the opportunity to observe an evaluation experiment by expert evaluators. In that case, their experience and their capability to fulfill the requirements of this method would increase. Finally, we could also have the software product evaluated simultaneously by two participants, in order to collaborate and talk to each other. Thus, as far as this method is concerned, we would be able to record sufficient data to analyze.

Regarding the inquiry methods used in this experiment, we found that some of the responses or the statements given by the participants were much easier to analyze when we referred to the data collected by the experimental methods, especially the User Actions Logging. By the means of the recordings saved in this method we were able to understand and clarify more efficiently what participants stated during the interview and the focus group sessions. Furthermore, having these recordings available we could also confirm that the participants' responses were valid and that the usability faults discovered by them actually existed.

Finally, we noticed that some participants were not very talkative during the interview session. Their responses were usually brief and sometimes of one word. However, during the focus group session, which was a moderated discussion among all the participants, they were able to collaborate with each other. In this way they had the opportunity to collect others opinions, agree or disagree with whatever was stated and clarify what they actually mean. In other words, they were able to determine their own opinion and express themselves clearly and with confidence. Moreover, during this session there was a brainstorming, where participants suggested various ideas for improving the LvS software and proposals to develop a user-friendly interface.

6. Revised version of LvS software

Being open-source, the LvS software is under continuous development in order to enhance its usability and extend its functionality and available features. The current version incorporates bug fixes and performance enhancements, as well as suggestions by teachers/project partners who used it in their foreign language classes.

The results of the experiment described in this paper were also taken into consideration, although not all the points made by the evaluators were considered problems needing correction. The fact that the evaluators were not education specialists had as natural corollary that the suggestions they made were not always in line with the purpose of the software. In some cases LvS was erroneously perceived as a professional subtitling software with the expectations that this entails, whereas in reality the software was designed with the help of teachers for the specific needs of foreign language learners.

The usability faults reported by the evaluators and corrected in the current version are the following: The duration of the splash screen appearing at the start was fixed to last only a second and then disappear so as to uncover the first dialog box and let the user continue interacting with the software. The other feature that was changed was the use of the tooltips which now also appear when the user passes the pointer over the column headers “Teacher” and “Student” of the grid, as these headers were not visible otherwise. Moreover, the software error concerning the inability to edit overlapping subtitles was corrected and a message was added to inform the user when the text size exceeds 39 characters per line. Finally, the usability fault concerning the activity instructions was taken into account at the stage of developing instructions for new activities, even though this feature is not related to the software but rather to the design of the activity itself.

7. Conclusions

In this experiment four complementary usability evaluation methods were employed for the evaluation of the Learning via Subtitling tool. The experiment took place in a Software Quality Assurance Laboratory and each evaluation method revealed a number of usability problems to the evaluators. Although most of these faults were common with all methods, some of them were unique to only one of them. Besides, in order to determine and understand clearly the results arisen from the evaluation, the outcomes of all the employed methods had to be investigated and compared to one another. This fact indicates that, even though all methods are useful, none is sufficient to reveal or clarify all usability problems of a software application, thus justifying the need for combining both experimental and inquiry methods such as presented in this paper.

However, it is obvious that there is a significant tradeoff between the number of the different methods that will be employed in order to evaluate a software application and the time and the resources demanded to achieve this usability evaluation. The evaluation leaders must firstly decide how much effort is to be exerted in similar experiments. But they must also bear in mind that combining different methods can help identify the majority of the usability problems and provide a spherical point of view in terms of usability.

The participants in the usability evaluation of LvS ranged from relatively experienced users to experts in computer skills and had attended a training seminar including courses relevant to software development methods, software testing techniques and software quality assurance methods focusing mainly on usability evaluation methods. Involving participants with this particular profile had a weakness and an advantage as shown in the evaluation process. The weakness was that, due to their participation in the seminar, they faced the evaluation process as a way to test and prove their capabilities in evaluating the usability of a software product. As a result of this, they appeared overeager to find usability faults in LvS and in some cases they exaggerated in judging the usability of its interface. The main advantage was that the particular participants, due to the training they received and their experience with software, managed to uncover a number of serious usability faults that were not discovered in previous evaluation sessions.

A further conclusion drawn from this study relates to the employment of the Thinking Aloud Protocol method in the usability evaluation of the software. The use of this method did not provide us with the data that we were expected since the majority of the participants did not verbally express their actions, thoughts, feelings and opinions during the evaluation session. Despite the training course and the specific guidelines provided in the orientation phase, they found it difficult to express themselves loudly during the whole evaluation session and concentrate in the interaction with the software at the same time. Furthermore, they were not accustomed to hearing their own voices throughout the whole process. An additional preventing factor was the nature of the software which includes listening sessions. A solution to these problems would be the further training of the participants, so as to become more experienced, in the Thinking Aloud Protocol method and have better performance. Concluding, we can say that despite the fact that Thinking Aloud Protocol is a very useful usability evaluation method, it is not applicable in all cases of usability evaluation and requires experienced users for its performance.

References

Avouris N.M. (2001). An Introduction to Software Usability. Workshop on Software Usability. Proc. 8th Panhellenic Conference on Informatics, vol 2, pp. 514-522, Nicosia, November 2001, Livanis Publ., Athens.

Dix A., Finlay J., Abowd G., Beale R. (2004). Human-Computer Interaction (3rd edition). Prentice Hall, NJ.

Dumas, J.S. and Redish, J.C. (1999). A Practical Guide to Usability Testing (Revised Edition). Intellect, Exeter.

Hellenic Open University (HOU), (2009). Web site: http://www.eap.gr/

ISO/IEC 9126 (2001). Software Product Evaluation – Quality Characteristics and Guidelines for the User, International Organization for Standardization, Geneva.

ISO 9241-11 (1997). Draft International Standard on Ergonomic Requirements for office work with visual display terminals (VDT), Part 11: Guidance on Usability, ISO.

Ivory M., Hearst M. (2001). The State of the Art in Automating Usability Evaluation of User Interfaces. ACM Computing Surveys, 33(4), 470–516.

Jeffries R., Miller J. R., Wharton C., Uueda K. M. (1991). User interface evaluation in the real world: A comparison of four techniques. In Proceedings of the Conference on Human Factors in Computing Systems (New Orleans, LA, April), pp. 119–124. New York, NY: ACM Press.

Kennedy, S. (1989). Using video in the BNR usability lab. SIGCHI Bulletin, 21(2), pp. 92–95.

Molich R., Thomsen A. D., Karyukina B., Schmidt L., Ede M., Van Oel W., Arcuri M. (1999). Comparative evaluation of usability tests. In Proceedings of the Conference on Human Factors in Computing Systems (Pittsburgh, PA, May), pp. 83–86. New York, NY: ACM Press.

Nielsen J. (1993). Usability Engineering. Boston, MA: Academic Press.

Nielsen, J., and Landauer, T.K. (1993). A mathematical model of the finding of usability problems. Proccedings of ACM INTERCHI'93 Conference, pages 206-13, Amsterdam, The Netherlands,24-29 April 1993.

Rubin, J. (1994). Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. John Wiley & Sons, Inc.

Schneiderman B. (1998). Designing the User Interface, Addison Wesley.

Sharp, H., Rogers, Y., and Preece, J. (2007). Interaction Design: beyond human-computer interaction (2nd ed.). Wiley.

Software Quality Research Group (SQRG), (2009). Web site: http://quality.eap.gr/

Stavrinoudis D., Xenos M., Peppas P., Christodoulakis D. (2005). Early estimation of users' perception of software quality. Software Quality Journal, 13(2), 155-175.

Wixon, D., Wilson, C. (1997). The usability engineering framework for product design and evaluation. In M.G. Helander, T.K. Landauer and P.V. Prabju (eds), Handbook of Human-Computer Interaction. Elsevier, Amsterdam, 653-688.

Appendix. Interview Questionnaire

Section A	
General information:	
1. Age	
2. Gender	
3. Profession	
Section B	
1. Knowledge of foreign languages and Level (Excellent, Very good, Good, Basic)	
2. Frequency of computer use	a. Everyday b. Three to four times a week c. Once or two times a week d. Less than once a week
3. Computer skills	a. Very experienced b. Relatively experienced

	c. Inexperienced
4. Use of computer for learning purposes	a. Very often b. Often c. Rarely d. Never

Section C (1=strongly disagree, 2=disagree, 3= uncertain, 4=agree, 5= strongly agree)					
1. The LvS software is easy to use	**1**	**2**	**3**	**4**	**5**
2. The LvS software is attractive	**1**	**2**	**3**	**4**	**5**
3. The time spent for learning to use the LvS Software was appropriate in comparison with the learning result.	**1**	**2**	**3**	**4**	**5**
4. The LvS Software offers all appropriate features to complete the LvS activity	**1**	**2**	**3**	**4**	**5**
5. The structure of the LvS interface is helpful for completing the tasks of an activity	**1**	**2**	**3**	**4**	**5**
6. The language used in the LvS interface is clear and understandable	**1**	**2**	**3**	**4**	**5**
7. I have read the instructions	**1**	**2**	**3**	**4**	**5**
8. The instructions were explicit	**1**	**2**	**3**	**4**	**5**
9. The instructions were easy to use	**1**	**2**	**3**	**4**	**5**
10. There is consistency in the terms and symbols used in the LvS software	**1**	**2**	**3**	**4**	**5**
11. The texts and the icons used in the LvS software are readable	**1**	**2**	**3**	**4**	**5**

Section D (open questions)	
1. Did you encounter any technical difficulties?	
2. As far as usability is concerned, which do you think is the software's greatest problem?	
3. Would you change or add any features into the LvS Software?	