Synchronous Text Reviewing over Internet

MAGNUS MAGNUSSON

TRITA-NA-E04124

Master’s Thesis in Human Computer Interaction (20 credits) at the School of Computer Science and Engineering, Royal Institute of Technology year 2004
Supervisor at Nada was Kerstin Severinson-Eklundh
Examiner was Kerstin Severinson-Eklundh
Synchronous Text Reviewing over Internet

Abstract

This Master’s thesis deals with the development of a prototype application for reviewing texts over the Internet.

The process of developing the prototype included literature studies, research on how ordinary face-to-face text reviewing is conducted, implementation of the prototype in the Java programming language and finally usability tests.

Reading the literature on computer supported collaborative work (CSCW) and the process of writing helped to define what was to be researched. This resulted in two studies on text reviewing; one field study and one case study. The target group for the research was teachers and students in second-language courses. The field study consisted of interviews. They were aimed at identifying the target group’s thoughts about text reviewing.

The case studies took the research a step further with direct observations of text reviewing.

The prototype was programmed and tested iteratively, which means that between each new version a usability test was conducted. This lead to the main goal, which was to develop an application that is easy to use and is supporting the needs of the target group.
Synkron textgranskning över internet

Sammanfattning

Detta examensarbete behandlar utvecklingen av en prototypapplikation för granskning av texter via internet.

Utvecklingsprocessen omfattade litteraturstudier, forskning om textgranskning, implementation av prototypen i programmeringsspråket Java och slutligen användartester.

Litteraturstudierna bidrog till att definiera vad som skulle undersökas inom textgranskning. Detta resulterade i två studier; en fältstudie och en fallstudie. Målgruppen för forskningen utgjordes av lärare och studenter i kurser i andraspråk.

Fältstudien bestod av intervjuer med målgruppen. Dessa intervjuer identifierade vad målgruppen tänkte om text-granskning.

Fallstudien gick ett steg längre genom direkta observationer av text-granskning ansikte-mot-ansikte.

Prototypen programmerades och testades växelvis, så kallad iterativ utveckling, där varje ny version användartestades. Detta bidrog till att huvudmålet för examensarbetet gick att uppfylla, vilket var att utveckla ett program som är lätt att använda och stödjer användarnas behov.
1 Background ................................................................................................................... 1

2 Research Questions ...................................................................................................... 2

   2.1 Goals ....................................................................................................................... 2
   2.2 Target Group .......................................................................................................... 2

3 Overview of the Research Area ..................................................................................... 3

   3.1 Computer Supported Cooperative Work ................................................................. 3
   3.2 CSCW in the Context of Conversations ..................................................................... 3
   3.3 What is Groupware? .................................................................................................. 4
   3.4 Groupware for collaborative writing .......................................................................... 5
   3.5 Face-to-face Communication ..................................................................................... 5
       3.5.1 Body Language ................................................................................................. 5
       3.5.2 Gaze .................................................................................................................. 6
       3.5.3 Tone of Voice ................................................................................................... 6
       3.5.4 Back Channels ................................................................................................. 6
   3.6 Text-based Communication ....................................................................................... 7
       3.6.1 Loss of Back Channels and Body Language ....................................................... 7
       3.6.2 Grounding aspects ........................................................................................... 7
       3.6.3 Loss of Context and Deictic Reference ............................................................... 9
   3.7 Reviewing of texts .................................................................................................... 9
   3.8 Annotating ............................................................................................................... 9

4 Research on Text Reviewing .......................................................................................... 10

   4.1 Description of suitable methods .............................................................................. 10
       4.1.1 Field Study ...................................................................................................... 10
       4.1.2 Case study ....................................................................................................... 11
           4.1.2.1 Observation techniques .......................................................................... 11
           4.1.2.2 Reasons for realising a case study .......................................................... 11
   4.2 How the research was conducted .......................................................................... 12
       4.2.1 Realization of Field Study .............................................................................. 12
       4.2.2 Field Study Results ....................................................................................... 12
       4.2.3 Design and Realization of Case Study ............................................................ 13
           4.2.3.1 Practical Preparations ............................................................................ 13
           4.2.3.2 Experiment Goals ............................................................................... 13
           4.2.3.3 Methods of Observation ................................................................. 13
           4.2.3.4 Suggestions for Observation ............................................................ 14
           4.2.3.5 Instruments ...................................................................................... 14
       4.2.4 Case Study Results ......................................................................................... 14
           4.2.4.1 Observation 1 .................................................................................... 15
           4.2.4.2 Observation 2 .................................................................................... 16
           4.2.4.3 Observation 3 .................................................................................... 17

5 Design of Prototype Application .................................................................................... 18

   5.1 Method used in analyses ....................................................................................... 18
       5.1.1 Task Analysis ................................................................................................. 18
   5.2 How the prototype was designed ........................................................................... 19
       5.2.1 Prototype Design ......................................................................................... 19
       5.2.2 Design work flow ......................................................................................... 19

6 Prototype Implementation ............................................................................................. 21

   6.1 GUI specifications ................................................................................................. 21
       6.1.1 Support for different roles of the user ............................................................ 21
       6.1.2 Support for viewing the text ......................................................................... 21
       6.1.3 Support for grounding between users ............................................................. 22
       6.1.4 Support for marking words and sentences .................................................... 22
       6.1.5 Support for “What you see is what I see” ..................................................... 22
   6.2 Implemented specifications ................................................................................... 22
       6.2.1 First specification: Different roles ................................................................. 22
       6.2.2 Second specification: Viewing the text .......................................................... 23
11.3.1 Purpose ........................................................................................................ 49
11.3.2 Problem statements .................................................................................. 49
  11.3.2.1 Questionnaire .................................................................................... 49
  11.3.2.2 Observation part ............................................................................... 49
11.3.3 User profile .............................................................................................. 50
11.3.4 Methodology .......................................................................................... 50
11.3.5 Test environment/equipment ................................................................... 50
11.3.6 Test monitor role ..................................................................................... 50
11.4 Screenshot of GUI ........................................................................................ 51
11.5 Equipment used in the development .......................................................... 52
1 Background

In the light of the last years increase of use of computers and the Internet, in both schools and organisations, the need for effective communication and collaboration tools has grown. Nowadays there are many different courses where you only meet the teacher a couple of times and most of the studies are done at a distance. This is called distance learning or e-learning. In large international organisations the cost of travelling is large and thus they have a need for tools that allow them to do teamwork at a distance. The work is distributed among companies and departments that need to collaborate.

These needs have, in turn lead to current research on web-applications with support for collaborative work. This lead to the idea to develop an application for text reviewing online, within the scope of my master’s project.

The idea for this project was inspired by an application called the Domain Help System (DHS) (Rodriguez, 2002, chapter 3.1). DHS is a system for making online annotations to a text. It is used asynchronously and is accessed through a webpage. It has been researched in several studies with second-language students. The idea is to let a teacher review a text from a student in the system and make annotations that are made available online to the rest of the class. DHS is also used for discussing the contents of a text.

The largest difference between DHS and the system thought of for this project is that it is designed for synchronous use instead of asynchronous as in the DHS.

Synchronous distributed communication is when the participants are located apart from each other and all communication takes place in real time. This is in contrast to asynchronous distributed communication were the communication takes place over time and not at the same time, in for example email systems.

Among the first specifications was that the system should be web based, i.e. reachable through a web page and it should contain some form of chat were the users could communicate directly with text messages. A system of this type would be beneficial in many different settings, from technical writing applications in large organizations to school essay reviewing. It was this latter area that was chosen for my research. The reason was that there already exists a research group on second language learning at the Department of Numerical Analysis and Computer Science at the Royal Institute of Technology. Thus the research done in this project would be relevant to the research group. Another reason was my experience as a teacher in a second language which would be useful when conducting research.

To be able to design and develop a new application from start it was important to understand what text reviewing actually is. Therefore research in the area needed to be done. This led to several interviews aimed at defining what was to be researched. Following this, a case study was conducted with a series of observations. They were aimed at detecting the different tasks, actions and roles that exist during the revision process.
2 Research Questions

How do you allow a reviewer to review texts from one or several writers in a setting where the reviewer is not located in the same room as the writer or writers?

Ordinary reviewing of a text is conducted with the writer sitting next to the reviewer or the reviewer is alone with the text. These are the processes that most people are used to from school. With focus on the writer sitting next to the reviewer this implies a need for an application that supports synchronous use. When the reviewer and writer are not co-located a distributed system is necessary.

The research questions below were aimed at clarifying what needed to be researched before a prototype could be developed.

- What tools should be implemented in the prototype to support both synchronous and distributed use?
- How should the communication take place?
- Are there already any applications with support for synchronous distributed reviewing through a web page? If so, are these applications available for research or are they all commercial?

2.1 Goals

The main goal of the project was to develop and evaluate a prototype application. It should support synchronous distributed reviewing as well as the target groups’ particular requirements and needs.

The following sub goals made way for the main goal:

1. Find out if applications that can be used for synchronized distributed reviewing exist and analyses of the possible findings.
2. Learn how ordinary text reviewing is conducted: how a reviewer reads, corrects and communicates with the writer.
3. Learn the writers’ requirements as well as the reviewers.
4. From the above sub goals design a prototype application.
5. Implement the prototype design with enough functionality to make usability tests possible.
6. Finally test if the prototype application supports the requirements and meets the needs of the target group users.

2.2 Target Group

The target group for this project was composed of students in language courses at higher intermediate levels as well as their teachers. Focusing on teachers and students helped narrowing the field of research. A smaller target group allowed for test groups in which the participants had a common background of studies in a second language. The reason for this was that they were believed to have a better understanding of what text reviewing is.

The level of experience of the target group was chosen on the basis that the students should have learnt enough of the language to be able to review other’s texts.
Future versions of the application could support other target groups as well. They could be writers and reviewers in various contexts such as technical writers in organisations and authors of books. A full version of the application could be adapted to any of the mentioned contexts.

3 Overview of the Research Area

3.1 Computer Supported Cooperative Work

The term Computer Supported Cooperative Work (CSCW) was originally coined by Irene Greif and Paul Cashman as a shorthand way of referring to a set of concerns about supporting multiple individuals working together with computer systems (Bannon & Schmidt, 1991, chapter 1).

According to Bannon and Schmidt (1991), the aim of CSCW is a better understanding of cooperative work in the context of developing computer systems with support for this type of work. When talking about support it is important to remember that the system should not make the task at hand more difficult. The first aim is to facilitate for the users to conduct their business as usual and the second aim will be to enhance their collaboration. Questions researched are for example: “How can computer-based technology be applied to enhance cooperative work relations? How can computers be applied to alleviate the logistic problems of cooperative work?”.

One of the core issues mentioned in Bannon and Schmidts article is “sharing an information space”. When conducting research for this project the very same issue was encountered: how to share a text between two or more participants. The chosen solution is described and discussed in chapter 6 “Prototype implementation” in this report.

3.2 CSCW in the Context of Conversations

In Dix et al. (1999, chapter 14) the area of CSCW is approached in a slightly different manner compared to Bannon and Schmidt’s report. Dix et al. start out discussing human conversation and what it consists of. Conversations are not all about what is said between the participants, important information is also embedded in body language, eye-gaze and tone of voice. These issues are important to consider when designing systems that support computer-mediated communication. When the only form of communication is text-based there are even more problems for example how to support grounding between participants (Clark & Brennan, 1991). Clark and Brennan describe grounding as a process during communication in which both participants try to determine if what has been communicated has been understood. That is if they have reached a common ground. An important part of grounding is giving positive and negative evidence, that is letting the other participant know if you understand him or not. This can be done through words such as “ok” or simply by nodding or shaking the head.

Dix et al. point out that text-based meetings are less effective when it comes to conflict solving compared with face-to-face meetings. This is just one of many problems related to lack of grounding.

Dix et al. also discuss how to design good systems that support cooperative work. One solution suggested and adopted for this project was the ethnographic
approach. The basic idea is that the designer tries to become one of the users. This is accomplished by working together with the people who are going to use the system and watching them as they collaborate. The approach is part of a user-centred design in which systems are designed for the users not the designers themselves (Rubin, 1994).

### 3.3 What is Groupware?

Groupware is defined as “a computer based system that supports groups of people engaged in a common task (or goal) and that provides an interface to a shared environment” (Ellis, Gibbs, and Rein, 1991).

Groupware can be classified according to three different criteria: a) where the participants are located, b) if they are working at the same time and c) what functions are supported by the groupware. The first two classification criteria can be put into a time/space matrix (Dix et al., 1999, chapter 13). This type of matrix is used as a rough guide to decide what type of system to develop, or what type of system is being analyzed.

The time/space matrix consists of two axes: the geographical dimension and the time dimension. The former is divided into co-located and remote participants, i.e. when the participants are working in the same room they are co-located and when they are not, they are working remotely. The latter dimension is divided into synchronous and asynchronous. The former is when the participants are working on something at the same time and the latter when they are not. See figure 1 for an example of a time/space matrix where the prototype described in this report has been entered into the matrix together with a couple of more familiar systems.

The classification of groupware by what functions it supports could for example be done through the distinction between a shared editor and group authoring system. It is also possible to make the distinction broader. Dix et al. (1999) suggest there are three main types of a groupware system: computer-mediated communication, meeting and decision support systems and shared applications and artefacts. These are not always used exclusively rather they are used together.
in varying degrees. The prototype for this project uses both “computer-mediated communication” and “shared applications and artefacts”, the former through the use of a text chat and the latter through the use of a shared editor.

3.4 Groupware for collaborative writing

Groupware tools for collaborative writing are becoming more common and word processors such as Microsoft Word supports different forms of collaborative writing. Other systems support collaborative writing over the web, for example BSCW (Basic Support for Collaborative Work) (Appelt & Hinrichs, 1996) which is a system that sets up the document as a shared document to which the co-authors have access. Lotus Notes which is a commercial application used for accessing and maintaining databases also includes a system for collaborative work over the web called Lotus Sametime. It allows users to enter a website and start sharing of a common document. It is not solely intended for collaborative writing rather it can be used to display other documents as for example images or graphics.

The Anchored Conversations tool is an application that allows the participants to set up small chat windows in a Word document. These windows are then “anchored” to the position in the word document. By using Anchored Conversations participants can communicate and discuss their work through a linear chat without losing context and without losing the possibility to refer to what they are working on through location.

3.5 Face-to-face Communication

When thinking of face-to-face communication it is easy to just consider the act of speech, but there is more to it than just speech and hearing. Included in face-to-face communication is also body language, gaze and tone of voice. The importance of these forms of communication depends on which context they are in.

When giving instructions on how to install an application on a computer the participants can do quite well with only speech. On the other hand when discussing a business deal in a meeting it is very important to see what the other participants are doing, as well as being able to make eye contact. The above example is one of many possible situations where body language and gaze are important.

3.5.1 Body Language

Body language is used in face-to-face conversations as a way to show what you are talking about. For example when reviewing a text together with the writer, deictic reference is important (Dix et al., 1999, chapter 14.2.3). Deictic reference is when you point at something, for example a word or sentence and say “let us delete this”. Body language also shows your feelings and attitudes to what is expressed.

Gesture is the part of body language which encompasses the arms and hands. It is used for pointing at objects, for emphasizing speech and assisting in turn-taking (Whittaker & O’Conaill, 1997). Whittaker & O’Conaill also refer to a study by Casell et al. which shows that gestures can be used to communicate more abstract
ideas. This is accomplished by showing relative positions of objects and orientations of the same.

A facial expression is the part of body language where information from the eyes, eyebrows, nose, mouth and forehead give the listener feedback and shows the emotional state. Information from the lips can also help in understanding what is being said. From the view of the speaker, facial expressions give information about if the listener understands, is interested, puzzled or not believing what is being said (Whittaker & O’Conaill, 1997).

Posture is how you move your upper body, thus it is literally a body language. Interest and boredom are two expressions easily conveyed through posture. When listeners lean forward and when speakers lean forward and draw back their legs then they are more likely to be interested. Boredom on the other hand is shown by lowering the head, turning the head to one side, supporting the head on one hand and leaning back and stretching out one’s legs (Whittaker & O’Conaill, 1997).

3.5.2 Gaze
Gaze is when you look at something or someone; it gives us visible information about our environment. If listeners didn’t gaze at the speaker occasionally they wouldn’t be able to see facial expressions, gestures and postures. As for the speaker it is helpful to watch the attentional behaviour of the listener since it gives the speaker a cue on whether the listener is actually listening or not. Combined gaze and body language help the participants reach a common ground. Joint attention is when both participants look at the same object, be it a pen or a word pointed to and they both know that the other person is looking at the same thing (Whittaker & O’Conaill, 1997).

The pattern and frequency of gaze is also an indicator of attitude, avoiding eye contact might indicate that the person is “defensive” or “evasive”. On the other hand when a person is searching eye contact it could indicate that the person is “friendly”, “mature” and “sincere” (Whittaker & O’Conaill, 1997).

3.5.3 Tone of Voice
Tone of voice carries information about the affective state of the person speaking, if he/she is happy, angry or sad. Tone of voice also expresses confusion or hesitation all of which are important cues for the listener (Dix et al., 1999, chapter 14.4.1).

3.5.4 Back Channels
Turn-taking is a basic activity in conversation. When two people are having a conversation it is common for each participant to wait on his turn before speaking. This behaviour is easy to recognise in contrast to what is called back channel responses (Dix et al., 1999, chapter 14.2.4) which are created during a conversation. Back channels are all the small nods, grimaces, shrugs, and noises for example “er” or “mm” that give feedback to the speaker on whether the listener understands or not. When back channels are lost, as they are in text-based communication, every acknowledgement or little noise has to be written explicitly.
In face-to-face communication back channels let the participants be more vague in their statements while at the same time allowing for the listener to give hints if he understands or not. It is these back channels that speed up the conversation and also help the speaker when he/she is unsure of what to say.

3.6 Text-based Communication

In most asynchronous and some synchronous groupware systems communication through text is the most common form. Even though people are used to writing emails and letters, using text as a substitute for speech is quite different. Dix et al. (1999, chapter 14.4) discuss what problems there are when adapting from speech to text-based communication.

3.6.1 Loss of Back Channels and Body Language

When communicating with only text there is an important difference compared with face-to-face communication and that is the lack of back channels. Small nods, “mm” and other similar feedback are absent which in turn affects the contents of the written word. This means that the messages that are written will not look the same as if they had been spoken, they will be more explicit (Dix et al., 1999, chapter 14.4.1). In text-based communication not only back channels are lost but also tone of voice and body language. This cuts off our natural way of conveying our feelings and interpreting whether the other person is listening or not.

As mentioned earlier, in some situations body language, gaze and tone of voice are very important. This is one reason for the invention of so called “smileys” used by people all over the world to convey feelings through text-based media. A smiley is when you create a face through the use of a couple of characters for example: ;-) (Dix et al., 1999, chapter 14.4.1).

It has been noted that textual messages use stronger language than face-to-face speech. The writer of the message might feel more inclined to criticize than if he was standing right in front of the receiver. Fortunately the way people respond when reading a message is also different than the way they respond to a spoken message. For example when a participant in a conversation reads a message that criticizes him/her or his/her work it does not have the same emotional effect as a spoken word could have had (Dix et al., 1999, chapter 14.4.1, p.530).

3.6.2 Grounding aspects

Different types of media need different techniques for the grounding process to function properly. Thus changing between media implies changing how grounding is performed. Clark and Brennan (1991) explain the eight most important characteristics of a medium that affects communication and grounding. The following list and descriptions are a summary of Clark and Brennan’s explanations.

1. Co presence
2. Visibility
3. Audibility
4. Co temporality
5. Simultaneity
6. Sequentiality
7. Reviewability
8. Revisability

Co presence is when participants are in the same room or area. Co presence is only available in face-to-face conversation. Since in other media it is not possible to see, hear and touch what the other participants is looking at, listening to or touching.

Visibility is when participants are visible to each other. It is available in face-to-face conversation, video conferences, certain mobile phone systems and web cameras, although the last three media cannot adequately show what the participants are looking at or doing.

Audibility is when both participants can hear each other. This preserves important parts of the grounding process such as tone of voice.

Co temporality is when an utterance is heard as soon as it is said. It is not available in text-based media or very long distance satellite phones, thus these media impose a constraint on the communication.

Simultaneity is when the participants can send and receive at the same time. It is available in face-to-face conversation as well as some older chat systems where each character was sent as it was typed.

Sequentiality is how the utterances are ordered. In face-to-face conversation interruptions are often disturbing since the sequence of turns is interrupted. This is not the case in for example email conversations where a participant might receive several different emails in between conversation.

Reviewability is the possibility for the participants in a conversation to review what has been said, available in text-based communication and recorded messages.

Revisability allows a participant to read through his message before sending it.

One thing that the characteristics have in common is that they are all weaker in text-based communication than in face-to-face communication (Dix et al., 1999, chapter 14.4.2). Most of the above characteristics impose more constraints on text-based communication than face-to-face communication. Co temporality can never be achieved in the same way with text-based communication as in face-to-face communication since messages written at the same time will arrive one after the other. This is inherent in a linear transcript where text messages are entered one after one and cannot be avoided. In face-to-face communication simultaneity allows for back channel feedback which is lost in text-based communication. As Dix et al. put it: “…the nature of typing makes it all but impossible to type your message whilst looking for your colleague’s “back channel” response”.

Sequentiality is present in linear transcripts but due to the lack of co temporality and simultaneity the sequence of messages can easily get muddled up.

When communicating through text the results of the limitations put on these constraints all add up, resulting in problems with achieving a common ground. Although there are grounding problems it has not stopped chat applications like ICQ or MSN messenger from becoming very popular and widely spread.
3.6.3 Loss of Context and Deictic Reference

Text-based communication also suffers from lack of external and internal context (Dix et al., 1999, chapter 14.4.4). External context is the surroundings in which the participants are located, for example if they are looking at the same screen then this is their common external context. One solution to the problem with lack of external context adopted in groupware has been to try and give the participants the same view, WYSIWIS (what you see is what I see). The same view means that each participant sees the same application on their screen. Thus the application itself or some special software keeps track of the view.

Internal context is the conversation itself. In speech the context relies heavily on sequence and adjacency. Dix et al. (1999, chapter 14.4.4) give an example that shows lack of internal context, three people are having a conversation using a linear text transcript chat:

1. Alison: Brian’s got some lovely roses.
2. Brian: I’m afraid they’re covered in greenfly.
3. Clarise: I’ve seen them, they’re beautiful.

Clearly the above conversation suffers a break down when Clarise’s message gets displayed after Brian’s. In cases like this one the breakdown is easily repaired but it shows the inherent problems with linear chats. A solution to this specific problem is to use a hypertext-based system that keeps track of which answers belong to which messages.

Participants also face another problem and that is how to show what they are talking about since in most cases they have no way of using deictic reference (Dix et al., 1999, chapter 14.4.4) to an object. There is no easy fix to this problem and solutions have to be made up according to the context of the system. For example: a text chat can be embedded within the object referred to, as in the Anchored Conversations tool (Churchill et al., 2000).

3.7 Reviewing of texts

The revision of a text is part of the process of writing and includes: reading and editing (Rodriguez, 2003). Depending on the medium used in the writing process different problems with revision appear. A writer who only revises his text on the computer will often concentrate on local problems and low-level errors such as misspelling. In comparison a writer who writes with pen and paper and reviews manually will have a better understanding of the overall text (Rodriguez, 2003).

3.8 Annotations

Rodriguez (2003) discusses the importance of annotations. Annotations are among the most common techniques used in the reviewing process. They range from notes on new ideas to simple spelling corrections. Recent research on how to support digital annotations has resulted in several systems, for example the DHS (Rodriguez, 2003) a system for annotating a document on the web.
Important aspects to consider when implementing a system for digital annotations are: allowing the reviewer freedom to choose his/her own annotation style and how to connect the annotations to the text. The latter is important since different ways of locating the annotations in the text will lead to different types of problems. For example positioning the annotations directly in the text will make it easier to understand their context although it is more difficult to read them in sequence.

4 Research on Text Reviewing

As stated earlier in the chapter on project goals the main goal of the project was to develop a program that is usable. Usable for whom and what you might ask?. To be able to answer these questions research on what text reviewing is needed to be conducted. The following two methods were regarded as suitable for researching the user’s requirements and needs:

- Field study
- Case study

4.1 Description of suitable methods

4.1.1 Field Study

A field study is a non-experimental scientific inquiry (Kerlinger & Lee, 2000, p.585). The goal of a field study is to discover relations and interactions in real social structures. Thus the study takes place outside a laboratory environment and the subjects are studied in their own location. For example: when we are studying office employees they are studied in their own offices. The focus of such a study could for example be to discover relations and interactions from a sociological or psychological perspective, how group work is conducted or what distinguishes a leader from others.

The type of field study chosen for this project was the exploratory and qualitative approach. Exploratory in the sense that relations and interactions between subjects are explored. This type of field study is also used in the preparation of more systematic and rigorous studies. In this project the field study paved the way for case studies with filmed observations.

The qualitative part of the study meant that subjects with known backgrounds were chosen rather than studying random subjects.

The reasons for choosing to conduct a field study were the following:

- Field studies are realistic since they are conducted with participants in their own surroundings, for example their office work place.
- My lack of resources and time to conduct a quantitative study with randomly selected subjects.
- Interviewing experts in the area rather than asking ordinary users was judged by me to give more insight into the area of second language learning.
- I believe interviews face-to-face gave me a better understanding of the user requirements compared with studies conducted solely through surveys and queries.
4.1.2 Case study
The case study is a research form suitable for researching a small, defined target group. The “case” can be an individual, a group of individuals, an organisation or a situation. When conducting case studies the aim is to try and get the “big picture”, to gather as much information as possible about the case. The representativeness of the results acquired depends on how the target group was chosen. Thus it is important to try and chose test subjects carefully, considering their background and suitability for the case study (Patel & Davidson, 1994).

4.1.2.1 Observation techniques
Two different observation techniques were used in my case study: Behavioural observation (Kerlinger & Lee, 2000, chapter 31) and Remembered observation (Kerlinger & Lee, 2000, chapter 31). Behavioural observation is a broad definition of different techniques for observing behaviour. Remembered observation is a simple technique that people use in everyday life. People observe what others do and say in one moment and remember this for later reference. Although this type of observation is unsatisfactory for science it is useful as a complement to other types of observation (Kerlinger & Lee, 2000, chapter 31, p. 728).

Kerlinger & Lee (2000, chapter 31) divide behavioural observation into two modes:

1. Watching other people do and say things.
2. Asking people to tell us about their actions and their behaviours.

Out of the above two modes of observation the first was chosen. Using the first mode meant being part of the experiment, observing but not participating. Kerlinger & Lee (2000, chapter 31) mention this as a possible problem since it is the observer who has to draw inferences and make conclusions from what he/she is seeing or hearing. The conclusions will always be based on the observer’s previous knowledge. Thus the result is subjective, although he/she can try to be objective. On the other hand if the observer had no previous knowledge then his/her conclusions would be inadequate (Kerlinger & Lee, 2000, chapter 31, p. 729). The dilemma is not easily solved, but with careful preparation of the experiment it can be minimized.

4.1.2.2 Reasons for realising a case study
After having conducted field studies with the aim of finding out what I needed to know about text reviewing, a case study was chosen as the next step.

The choice of researching text reviewing through a case study was based on a need to understand face-to-face text reviewing. Without knowledge of the processes that take place and the interactions between participants in the study it is difficult to build a computer system with support for these processes and interactions. It is my belief that direct observation is the best way of learning and understanding text reviewing when the number of test subjects is relatively small (6 subjects were used for this experiment). The study was filmed since later reviewing would be easier and more reliable when caught on tape compared with only having notes on paper.
4.2 How the research was conducted

4.2.1 Realization of Field Study

As mentioned earlier the field study was aimed at discovering the user’s needs. This meant that the questions were aimed at: understanding how the target group performed text reviewing, asynchronous or synchronous, what problems they had when reviewing and how they preferred to conduct revision of texts.

The target group consisted of three students and one teacher in a distance learning course in which one of the students participated. The target group was interviewed in their own environment; in this case it was at school. The following questions were posed to the target group:

Questions for the teacher

1. How long have you worked as a teacher in language courses?
2. Do you usually use the Internet in your work?
3. If the answer to the last question is yes, how often and for what purpose?
4. When commenting on a text face-to-face which problems arise, if any?
5. When commenting on a text through email which difficulties do you face?
6. How would you prefer to correct a text, face-to-face, in a class-room or at home?

Questions for the student

1. Sex:
2. Age:
3. Language level?
4. For how many years have you studied this language?
5. Do you use the Internet when studying?
6. If yes, how often and for what?
7. When receiving comments on a text face-to-face are there any problems?
8. When receiving comments on a text through email are there any difficulties?
9. How would you prefer to receive comments on a text, face-to-face or through email?

The questions were written as to gather information about the participant’s background; the reason was to better understand the future demands on a text reviewing application. It was also important to know what to observe in the next step when the laboratory experiments were to be conducted. The questions where deliberately chosen to be open, this in turn allowed the subjects to respond freely. The answers are summarized in the next chapter featuring the results.

4.2.2 Field Study Results

Three students from second language courses were interviewed, one female and two males. They were all studying higher intermediate levels in Swedish and English.

A common problem perceived by the students was that the teachers often had too little time available to help them with their questions. This was the case both when asking for help face-to-face and through email. This problem led to another
one, that the teachers often gave vague and unclear answers and sometimes misunderstood the questions posed.

The teacher chosen for the interview worked at the time on a daily basis with e-learning and was thought to be able to answer questions on how e-learning works. He used email as a means of receiving and sending texts and when asked “When commenting on a text through email what difficulties do you face?” he answered that it is sometimes difficult to give a good explanation for a certain correction or comment. He preferred to give explanations over the phone or in person rather then in writing.

After the last interview was finished it became clear that the interviews with the presumptive users did not give enough information about the reviewing process. Thus the results from the interviews helped define the text reviewing observations by giving a hint on what to observe and not observe.

4.2.3 Design and Realization of Case Study
Revision of texts can be done either by the student himself or by the teacher; it is the latter process that was investigated. The focus was on revision of texts in a language course, especially in a second language.

The case study was conducted in order to clarify the user’s requirements and needs. The study gave an understanding of the requirements through observing the process of revision done face-to-face. Face-to-face revision was chosen since I believed it would be the most relevant type of reviewing for this project.

4.2.3.1 Practical Preparations
Three teacher-student pairs were observed while reviewing a text. All three students were attending different second language courses, one took a Swedish course, one an English course and one studied Spanish. The teacher had taught or was teaching second languages, one was a retired Swedish teacher, one taught English and one studied to become a teacher in Spanish. The text was 1-2 pages long and written by the student who participated in the observation. With the consent of the test subjects these three sessions were videotaped for later analyses.

The sessions all took place in environments, in which the teacher and student were not situated in an ordinary classroom but in an office room. The sessions all started with the test monitor explaining what would happen and how the observations were to be conducted. Then the teacher started reviewing the text sitting next to the student while the test monitor filmed the session.

4.2.3.2 Experiment Goals
- Record which activities take place between teacher and student while they review a text.
- Identify problems with text reviewing face-to-face.
- Record collaborative work between teacher and student.

4.2.3.3 Methods of Observation
The test monitor observed the sessions and took notes on paper. After the sessions he watched the films and tried to answer the questions below, intended to guide the observations.
4.2.3.4 Suggestions for Observation

- How do the teacher and student refer to where in the text they are currently correcting?
- Does the teacher use any device to point at the text?
- How does the teacher correct the text?
- What kind of markers does the teacher use for corrections?
- What happens when the student doesn’t understand the teacher?
- What happens when the teacher doesn’t understand the student?
- Does the teacher wait for confirmation from the student that he/she has understood before proceeding?
- If the student disagrees with the teacher over some correction made, how do they resolve the disagreement?
- How does the teacher get the student’s attention when reviewing the corrections made?

The above questions were meant to help the observer in the analyses of the experiments. Focus was on how the teacher corrects texts, how the student and teacher solve disagreements and misunderstandings and how they get each other’s attention. The reason for the focus on correcting texts was that this is considered the main task at hand. The other three focus areas were chosen based on what I read in the literature, as well as comments made by my supervisor and examiner.

4.2.3.5 Instruments

A video camera was used to capture each session as well as pencil and paper. The reason for filming the sessions was that it would have been difficult to be able to answer the entire list of questions while at the same time observing what took place.

4.2.4 Case Study Results

Three observations of text reviewing face-to-face were made. The following is a summary of the results. Following this are the results from each observation.

- The first observations made during the experiments were the basic actions taken by the student and teacher: The teacher started reading the text, then made a correction, reread the corrected part, continued and asked the student if he understood. The student started listening and when he did not understand a correction he asked for an explanation, or when he did not agree with the teacher. This meant that the student was more passive in his work than the teacher, as expected. It even turned out that in one of the tests the student did not ask any questions at all, the reason for this was probably cultural being that the student came from Peru and the teacher from Sweden. This could be due to the fact that in the student’s culture you do not interrupt a teacher while he is tutoring.
- In two out of three observations a dialog between the teacher and student could be observed. These dialogs contained both positive and negative evidence (Clark & Brennan, 1991) as the process of grounding continued. An example was the process of understanding the text to be reviewed by the teacher. This process was important since without knowledge of what the text was about it was difficult for the teacher to know whether or not to make a
certain correction. As one teacher expressed it: “I can’t understand one thing of what you have written, please explain what the text is about?”.

- When reviewing, the text was read out loud by the teacher in all observations.
- In all observations the teacher corrected errors while at the same time reading through the text.
- At all times both the student and the teacher were able to see the same text at the same time. This meant that the student also saw when the teacher was hesitant or even made errors. One of the students mentioned this as a problem: “Sometimes it was difficult to follow the corrections made by the teacher which happened when he made a correction but later on changed his mind and wrote something different”.

4.2.4.1 Observation 1

The first observation was of reviewing a text in English. Both the teacher and student were male, the teacher 24 years old and the student 22.

1. How do the teacher and student refer to where in the text they are currently correcting?
   Answer: The teacher reads the sentences out loud. The student points to the text.

2. Does the teacher use any device to point at the text?
   Answer: Pen

3. How does the teacher correct the text?
   Answer: The teacher first corrects the simple errors like spelling mistakes then continues with checking of word order, grammar and comprehension. If something is unclear he asks the student for clarification.

4. What kind of markers does the teacher use for corrections?
   Answer: Uses an upside down “!” proof-mark for showing where to put a word or sentence in the text. The teacher writes small words between lines.

5. What happens when the student doesn’t understand the teacher?
   Answer: The student waited until the whole text was corrected and then asked about different corrections made and why.

6. What happens when the teacher doesn’t understand the student?
   Answer: The teacher first asked for an explanation and then tried to convey the meaning back to the student in Swedish, thus checking if it was what the student had intended to say.

7. Does the teacher wait for confirmation from the student that he/she has understood before proceeding?
   Answer: Yes, the teacher waited a while, if no positive or negative evidence was received then the teacher continued.

8. If the student disagrees with the teacher over some correction made, how do they resolve the disagreement?
   Answer: Both teacher and student try to reach an agreement in Swedish before proceeding in English. They take turns giving arguments until the agreement is reached.
9. How does the teacher get the student’s attention when reviewing the corrections made?
   Answer: Asked the student for his opinion or read out the sentence again.

4.2.4.2 Observation 2
The second observation was of reviewing a text in Swedish. The teacher was male and the student female. He was a retired Swedish teacher and she studied Swedish B course.

1. How does the teacher and student refer to where in the text they are currently correcting?
   Answer: The teacher reads the whole text out loud. The student points to the text.

2. Does the teacher use any device to point at the text?
   Answer: Pen

3. How does the teacher correct the text?
   Answer: The teacher reads through the text and corrects it at the same time. If the student asks a question: for example about grammar the teacher gives an example of a correct sentence and explains why it should be like that. An example from the videotape is when the student asked why the teacher put in a hyphen between two words, the teacher gave an example sentence and explained the rules for using hyphens.

4. What kind of markers does the teacher use for corrections?
   Answer: The teacher used several different types of markers:
   “\(\)” was used for inserting words and sentences
   \(\) was used for changing word order
   \(\) was used for changing a particular word
   \(\) was used for inserting small words or symbols
   \(\) was used for changing position of a word in a sentence. These markings are proof reading markings used by printing plants.

5. What happens when the student doesn’t understand the teacher?
   Answer: The teacher explains until the student understands and gives examples.

6. What happens when the teacher doesn’t understand the student?
   Answer: The student tries to explain in different words what he means, the teacher helps out by giving either positive evidence that he has understood or negative evidence that he needs more explaining.

7. Does the teacher wait for confirmation from the student that he/she has understood before proceeding?
   Answer: Yes, the teacher waits a short while for positive evidence then continues, he also asked the student if he had understood the text correctly.
8. If the student disagrees with the teacher over some correction made, how do they resolve the disagreement?
   Answer: No disagreements were observed

9. How does the teacher get the students attention when reviewing the corrections made?
   Answer: This was not an issue since the student paid attention to the teacher all the time.

4.2.4.3 Observation 3

The third observation was of reviewing a text in Spanish. Both participants were male and in their mid twenties. The reviewer studied at Stockholm University to become a teacher in Spanish, the student had recently taken a Spanish course at KTH.

1. How does the teacher and student refer to where in the text they are currently correcting?
   Answer: The teacher reads out loud the parts that he believes need correction.

2. Does the teacher use any device to point at the text?
   Answer: Pen

3. How does the teacher correct the text?
   Answer: The teacher suggested corrections and asked the student for his opinion.

4. What kind of markers does the teacher use for corrections?
   Answer: The teacher doesn’t use any special markers, just cross for deleting a word and arrows for inserting words in sentences.

5. What happens when the student doesn’t understand the teacher?
   Answer: The student asks directly what he doesn’t understand.

6. What happens when the teacher doesn’t understand the student?
   Answer: First the teacher tries to figure it out on his own then he asks the student for clarification.

7. Does the teacher wait for confirmation from the student that he/she has understood before proceeding?
   Answer: Not all the time, mostly the teacher kept reading and correcting unless he wanted to explain a correction or ask for clarification.
8. If the student disagrees with the teacher over some correction made, how do they resolve the disagreement?
   Answer: The student gives arguments for his point of view. The teacher helps by giving an alternative correction and explanation.

9. How does the teacher get the student’s attention when reviewing the corrections made?
   Answer: The teacher asks a question directly to the student, thus reminding the student of what they are doing.

5 Design of the Prototype Application

Once the observations of text reviewing face-to-face had been finished the next step was to analyse the data collected. Conclusions were then drawn by using Task Analysis (Dix et al., 1999, chapter 7). The design and programming of the prototype were based on a method called Iterative design and prototyping (Dix et al., 1999, chapter 5.5). After the development of the first prototype version was finished a heuristic evaluation (Dix et al., 1999, chapter 11.4.2) technique was used. The heuristics, i.e. guidelines, suggested in Dix et al. (1999) were used by me to look for basic faults in the prototype. Next followed the usability testing which was based on test sessions including observational techniques and query techniques (Dix et al., 1999, chapter 11). After the results from one test were collected a new cycle started with programming changes to the application, heuristic evaluation and another usability test. In my opinion using Iterative design and prototyping helps the programmer to develop the prototype into still better versions.

5.1 Method used in analyses

5.1.1 Task Analysis

“Task analysis is the study of the way people perform tasks with existing systems” (Dix et al., 1999, chapter 7) Task analysis is not only useful for analysing systems but also when analysing procedures and behaviour. For example: designing a new system that is partly based on an existing system or procedure. Finding out user requirements is a task well suited for the method. This is due to the method’s focus on how an existing system is used, what tasks that exist. Through analysis of the results from task analysis conclusions can be drawn about what parts of the old system to implement in the new system. Once the task analysis is complete the results help in determining how the new system will be used. Which old procedures that can be used in the same way as before and which new procedures that need to be taught to the user.

In the experiments conducted in this project, interactions between teacher and student were studied as well as their relations. It was also important to identify the task done by teacher and student, i.e. how they performed text reviewing. Dix et al. (1999) discuss the different roles that can be identified through tasks. It is important to support these existing roles when developing a new system. The users need to “feel at home” when using the new system. They should be able to recognise what actions are available without having to read a manual or take a course.
It is important to mention that in reality Task Analysis is an intuitive method which describes well how I reasoned when drawing conclusions regarding my observations.

5.2 How the prototype was designed

5.2.1 Prototype Design
A brief overview of the design work flow is given below. Chapter 6 - Prototype implementation covers the details.

The client server model was used when designing the program. The idea with this model is that an application is set up on a web server; this is the server part of the model. When a user visits the webpage, this is the client part, and logs on, the server application registers this event and creates a new process for text reviewing. This process waits for another user to enter before starting.

When developing the user interface the usability principles suggested in Dix et al. (1999, chapter 4.3) were followed. They list three main categories of usability principles: Learnability, Flexibility and Robustness. Learnability is how easy it is to start using the application. Flexibility means in how many different ways the user can exchange information with the application. Robustness depends on, among other things, how well the application allows a user to recover from errors. The aim of using these principles was that the target group should be able to use the application without tutoring.

5.2.2 Design work flow
The first stage in the design of the prototype application was to write down a list of the tasks it should support. This list was a result of the task analysis and the tasks were chosen according to the different roles.

With the list of tasks finished a prototype specification (see chapter 6.1) was written and a first paper design drawn, see figure 2 on the next page.
The first version of the client prototype’s graphical user interface (GUI) was based on this design. The GUI is the part of the application that is used to interact with it. The GUI usually contains different buttons, text fields and menus. Before settling with the design both supervisor and examiner were consulted which led to changes. When discussing how to make the GUI more intuitive it was decided that changing the names of the fields, buttons and menu items was important. The resulting paper design is shown in figure 3 on the next page, changed names are marked with a frame. This design was then used for the first version of the prototype application. After the parts of the GUI were identified a communication protocol for transferring the texts between clients had to be chosen. This is described further in chapter 6 Prototype Implementation.
6 Prototype Implementation

Deciding how to implement the prototype turned out to be a challenge. In Java, which was chosen since I am most familiar with this programming language, there are several different ways in which you can program network applications. The first version of the prototype was based on a communication protocol using Sockets. Unfortunately I did not have enough knowledge about Sockets and since the aim of the project was not to deepen my knowledge in Java, Java’s Remote Method Invocation (RMI) protocol was used instead. The reason for choosing RMI will be further described in chapter 6.4.2 and RMI will be described in chapter 6.4.3.

For understanding of the following chapters, basic knowledge in programming is expected of the reader.

6.1 GUI specifications

From the results of the observations and from the literature conclusions were drawn which led to the following specifications for a prototype application.

6.1.1 Support for different roles of the user

When observing the students and teacher while reviewing texts it became clear that the prototype needed to support two different roles: the role of writer and the role of reviewer. Dix et al. (1999, chapter 7) discuss how roles can be identified through the actions taken by the participants, thus the prototype needed to support actions for both the writer and the reviewer.

6.1.2 Support for viewing the text

Support for displaying the text, both the original text and the text while being revised. In all the observations the teacher and the student had simultaneous
access to the original text and the corrected text. They both sat in front of the text and the corrections were made directly on the paper containing the original text. Rodriguez and Brunsberg (2004) report similar findings that the original text needs to be available together with the text being reviewed.

6.1.3 Support for grounding between users
Support for communication between teacher and students, either by voice using a microphone, through a text chat system or both. In two out of three observations there was a dialog between the teacher and student which helped them understand each other. This dialog implies that the need for direct communication is important. The results from the observations also showed that this dialog helped the reviewer understand what the text was about.

6.1.4 Support for marking words and sentences
Support for different ways of marking text when revising and correcting. In all the observations the teachers used different types of notation for correcting texts, thus a web application should let the teacher decide by himself what kind of notation to use. As stated by Rodriguez and Brunsberg (2004) when given the possibility to choose their own ways of making annotations the users take it. Thus it is important that the interface is flexible.

6.1.5 Support for “What you see is what I see”
Support for locating where the teacher is currently reading in the text. This is easy when the teacher and student are co-located since all they need to do is point to the text. Once separated by distance they need some other way of letting each other know what they are referring to in the text. Rodriguez and Brunsberg (2004, chapter 5.2.6) report that the students used several different ways of helping the writer find the corrections:
- Ordinal position – is when the reviewers use adjectives such as the last, the beginning, the next, the sentence before.
- Numbering position – is when the reviewers enumerate the sentences or paragraphs.
- Topic related location – is when the reviewer mentions what the sentence is about: its content or theme.
- Quoting the beginning of the sentence – is simply a quote of the first part of a sentence.

6.2 Implemented specifications
This chapter describes how the five specifications were implemented in the prototype, for a screenshot of the resulting GUI see appendix 11.4.

6.2.1 First specification: Different roles
Deciding how to support the roles of writer and reviewer meant choosing if the roles should be set explicitly or if the users should both be able to do the same tasks. I chose to make it explicit; when the users log on they choose if they want to be writer or reviewer. The reason for this was that it made the GUI simpler to learn since the writer can only use his buttons and vice versa for the reviewer. The
downfall of this choice is that the author cannot edit the text being reviewed. This implies that the prototype is not suitable for cooperative writing tasks, although it is not complicated to extend the functionality of the prototype in a later stage.

6.2.2 Second specification: Viewing the text

To support the second specification two windows were placed vertically above each other. The top window is used for displaying the original text and the bottom window displays the text while being corrected. See figure 4.

![Figure 4. Text areas for original and reviewed text.](image)

The reviewing window (bottom window) does not update continuously or automatically, rather it is up to the reviewer to decide when to update the author’s view. The choice between automatic updating and manual updating depends on if you believe that showing the whole revision process is beneficial for the writer or not.

In his doctoral dissertation Rodriguez (2003) discusses how Col·laboració (a web based program for collaborative writing) is best used for collaborative writing: “Users might want to minimize the amount of editing, avoiding insertion of a text that will probably change later. It might be better to discuss it first in the comment space and later make it part of the section.”. In the context of my prototype it would imply first discussing a correction in the chat window (see figure 5 on the next page) and then making the correction. The correction will be entered in the reviewing window and when the reviewer is satisfied he can update the window for the writer to view the correction. From the above conclusion and the results of the observations the choice of manual updating was made. This allows the reviewer, who is presumed to be experienced with reviewing, to give the writer a coherent picture of the revision process instead of bits and pieces as would be the case in a continuous stream of updates.
6.2.3 Third specification: Grounding between users

For the third specification a text chat system was created that lets the reviewer chat with one or more users. See figure 5.

![Figure 5. Text area for chatting.](image)

6.2.4 Fourth specification: Marking words and sentences

By presenting the reviewer with the ability to change size, colour and font of the text he is currently reviewing, he can choose in which way he prefers to mark corrections. In figure 4 the buttons labelled “Options for text marking” and “Mark text” is used by the reviewer. The former to choose his favourite markings and the latter to use them.

6.2.5 Fifth specification: “What you see is what I see”

The fifth specification can be implemented in several different ways as explained by Rodriguez and Brunsberg (2004, chapter 5.2.6.).

In my prototype the window for reviewing texts displays the same part of the text to the writer as the reviewer sees as soon as the reviewer presses the update button. This is accomplished by using the position of the caret from the reviewer’s side and sending it to the writer’s window. The result is called WYSIWIS “what you see is what I see” (Dix et al., 1999, chapter 13). By setting the caret at the same position as for the reviewer all the participants who are logged in to the prototype see the same part of the window.

6.3 Planned Functionality

More and more people are connecting to the internet using broadband connections and thus network speeds increase. This in turn opens up for voice and video
communication. I believe that especially voice communication would be important for this type of application. Since in all observations there was turn taking incorporating both positive and negative evidence which is lost when transferring the communication to text only (Clark & Brennan, 1991).

I plan to add printing capabilities to the program. This would allow the end-user, either be it the writer or reviewer, to make a print out of the current work and this in turn would allow for easier reading of longer texts (Dix et al., 1999, chapter 1). The reason for not adding this function from the beginning was that the aim of the research was reviewing of shorter texts, 1-2 pages in length.

Another planned function is the capability to choose the language of the GUI. One way to do this would be through a menu called “Language” where the user can choose between several different languages or load his own language. The application would then read a text file which would contain names in the new language for the different parts of the GUI.

During the first demonstration of the prototype the issue of how the reviewer could mark the text came up. One of the participants suggested a function that would allow the reviewer to choose his own proof-marks. These marks could be implemented as icons with the ability to insert them into the reviewing window. Adding this functionality would improve the application’s flexibility.

6.4 Technologies for Distributed Communication

6.4.1 Sockets in Java
All communication over the web is based on the TCP/IP protocol. In Java a TCP network connection is represented by a “Socket” which is a Java class. Using a socket the client application can create a connection with a remote host, for example a server application. When using sockets the communication channel is stream-based, this means that the data is flowing between client and server and is managed with stream classes in Java. These stream classes are then enhanced with buffering, byte reading or string reading capabilities.

6.4.2 Implemented Protocol
The first version of the prototype application used socket communication between the client and server side. When the server was started on the remote host it created an instance of a ServerSocket, which is a socket for the server that opens up a port number and then waits for clients to connect to it. Once the server was started it was possible for a client to connect to it by creating a Socket and using this Socket to connect to the ServerSocket through the remote host’s IP address and port number. If the connection succeeded it was now possible for the client to transfer data to the server and vice versa. The data sent and received was strings taken from the chat part of the prototype’s GUI. These strings were sent and received with buffered streams and written and read to the GUI with data streams. See figure 6 on page 27 for communication flow chart.
This protocol worked fine for sending and receiving strings, i.e. for chatting, but when the GUI was enhanced with different textual styles it was not possible to continue to use the protocol. It could have been changed to accommodate sending Java objects such as documents with styled text. The only reason for not changing the protocol was that there already existed a communication protocol in Java called RMI that can handle sending and receiving documents. RMI will be described in chapter 6.4.3.
Socket protocol

Remote side

1. The remote host computer is a UNIX server on which both the webpage and the Java server application resides.
2. When the Java server is started it creates a ServerSocket object, which is an instance of the Java class ServerSocket. This object opens a specified port number and waits for client Sockets to connect.

<table>
<thead>
<tr>
<th>Remote host: Computer on the Internet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java server: Application running on the remote host</td>
</tr>
<tr>
<td>ServerSocket: Object created by the Java server</td>
</tr>
</tbody>
</table>

Local side

3. The local computer with a web browser is used to browse the webpage and run the applet.
4. When the Java applet is started it creates a Socket object which is an instance of the Java class Socket. This object connects to the ServerSocket by using the IP address of the remote host and port number as arguments when the object is created.
5. Once the Socket is connected to the ServerSocket, the applet can send and receive data through the connection, using different data streams, for example string streams for sending text.

<table>
<thead>
<tr>
<th>Socket: Object created by the Java client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java client applet: Application downloaded and run on local computer</td>
</tr>
<tr>
<td>Local computer: Home computer with web browser</td>
</tr>
</tbody>
</table>

Figure 6. Communication flow chart for Socket protocol.
6.4.3 Java RMI

RMI is short for remote method invocation, the first part of the name means it is used for remote communication between Java applications. The second part of the name represents the fact that when using RMI it is possible to call methods in remote applications. This is done in the same manner as you would call a method in a local application. Thus you need not know the exact details of the communication protocol since RMI takes care of choosing how to transfer the data.

RMI systems are often made up of a server and a client application. The server application resides on a remote host. When it is started it creates remote objects, references to these remote objects are then made available through a special rmiregistry. This registry is a separate process that is started on the remote host before the server is started. Its purpose is to keep track of the remote objects created by the server and allow clients to get references to them. Once the server is started it waits for clients to connect and get references to the remote objects through the rmiregistry and then make remote method calls on the objects. When programming the client application these calls look exactly the same as an ordinary method invocation in a Java application. This is the main advantage of using RMI instead of creating your own socket protocol (Java RMI, 2004-02-26) (Java Tutorial, 2004-02-26).

See figure 7 on the next page for communication flow chart. The next chapter describes the different parts of the prototype’s RMI implementation.
Java RMI communication protocol

Remote side

Remote host: Computer on the Internet

RMIregistry: Holds references to remote objects

Web page: Contains the Java applet

Java server application: Responsible for creating remote objects

Local side

Java applet: Client application which is downloaded from the webpage to the web browser on the local computer

Local computer: Home computer with web browser

Figure 7. Java RMI Communication flow chart.

The following is a description of the different parts in the flowchart.

1. The RMI registry is started first, it takes a specified port number as argument. This allows the client to find the references to the remote objects created by the server.
2. Next the Java server is started, it creates the remote objects.
3. The remote objects are bound to the registry.
4. The remote host runs a web server on which the web page with the Java applet resides.
5. The Java applet is the client application and is downloaded to the local computer when the user enters the webpage.
6. The applet is displayed in the user's web browser, thus running the applet is just like viewing any other web page on the web.
7. When the applet is started it tries to get references to the remote objects from the rmiregistry.
8. If it succeeds it receives a “stub” that is a short cut to the remote object.
9. With the “stub” as a reference the applet can call methods from the remote objects, as well as send data to other clients through the remote methods.
10. The server also uses “stubs” for short cuts to each client that connects to it, thus it can broadcast the data sent by one client to all clients by using its “stubs” as references to the clients. Stubs are further described in the following chapter.

6.5 Prototype Programming

The prototype application consists of seven Java classes. Three on the server side: ChatServer, ChatServerImpl, UserInfo and four on the client side: ChatClient, Chatter, ChatterImpl and MyStyledDocument.

Each class is briefly described below, for an overview refer to the UML (Pooley & Stevens, 1999) class diagram in figure 8 on the next page.

6.5.1 Server side classes

The ChatServerImpl class implements the remote interface ChatServer, which contains the available remote methods. By implementing a remote interface the ChatServerImpl’s methods are available across networks, thus between different Java virtual machines. This makes the ChatServerImpl a remote object that can be reached by the clients logged on to the system. ChatServerImpl is the main class which gets started first before any users can log on to the system. When a user logs on an object of the UserInfo class is created and stored in a hashtable. This is later used when the ChatServerImpl wants to find out which users are logged on, allowing for broadcasting texts or messages to all users (Writing Chat Systems with Java 1.1’s RMI, 2004-03-16).

6.5.2 Client side classes

The main class on the client side is the ChatClient applet. An applet is a Java program that can be used through a web page with an ordinary internet browser such as Internet Explorer. When a user enters the webpage the ChatClient applet starts and creates the graphical user interface that is used for reviewing texts. This includes creating two MyStyledDocument objects which are used for holding the texts. The applet also creates a ChatterImpl object and a ChatServer object. The former is used by the ChatServerImpl to communicate with the client, the latter is used by the client to communicate with the ChatServerImpl.
Figure 8. UML Class diagram for the prototype application.
6.5.3 Problems encountered
During the first development stage there were no problems encountered. It was not until the first version of the prototype was tested that such ones appeared. The choice of using Java RMI for the communication between client and server had unintentionally made the programming more difficult when it came to compatibility. When testing the prototype between two clients running different versions of the Java VM (virtual machine) it turned out that the texts could not be transferred. The reason was that when sending the texts between the clients a procedure called serializing is conducted by the Java object stream. Serializing is when a Java object is “coded” and sent to the other client where it gets deserialized, i.e. decoded. It was this serializing and deserializing that did not work between different versions of Java VM. After two weeks of reading Java forums, manuals and Sun’s Java pages as well as testing different debugging techniques the clients could still not transfer texts adequately. Between clients using the same Java VM it worked fine. To be able to continue with the usability tests a Java VM link was added to the web page from which the prototype is run. This made it possible for the participants to download the same version of Java VM before starting the test. With the same version on both the clients connecting to the server the usability tests could be conducted as planned.

Two months later, when this report was almost finished, I reprogrammed the communication part of the prototype. Java RMI was still used although instead of allowing it to serialize and deserialize the texts; RTF (Rich Text Format) documents were used. RTF is a document format which allows styled text to be exchanged between different applications, for example in Microsoft Word documents can be saved in the RTF format. This meant that the texts were stored in temporary RTF documents in the client and then sent as byte arrays to the other clients. Due to how Java RMI serializes and deserializes primitive data, byte arrays can be sent without having to worry about version problems. Up to this date the prototype has been tested on JDK 1.4.1 on one client and JDK 1.4.2 on the other client without problems.

7 Testing of the Prototype Application
Each test conducted had the user profile, i.e. participants with common background, and methodology in common with the other tests (see chapter 7.3.4 and 7.3.5). The methods used are described below and how and why they were used follows in the next chapter. As mentioned earlier in chapter 5 the usability tests were part of the development process.

7.1 Methods for usability testing
Rubin (1994, chapter 2) lists six points that are common for all usability testing:

- Development of problem statements or test objectives rather than hypotheses.
- Use of a representative sample of end users which may or may not be randomly chosen.
- Representation of the actual work environment.
• Observation of end users who either use or review a representation of the product. Controlled and sometimes extensive interrogation and probing of the participants by the test monitor.
• Collection of quantitative and qualitative performance and preference measures.
• Recommendation of improvements to the design of the product.

These six points were used as a basis for the usability tests in the project. Three different tests were conducted: exploratory, assessment and validation tests.

7.1.1 Exploratory test

Exploratory testing is conducted in the beginning of the development. The aim of the test is to explore the users’ mental model, i.e. what the user thinks of the prototype and also if our assumptions about the user are correct. Without knowledge about the users’ mental model it is difficult to know if the user understands the different parts of the GUI.

Rubin (1994, chapter 2) lists some typical user-oriented questions of which the ones deemed relevant are listed here:

• What do users conceive and think about using the product?
• Does the product’s basic functionality have value to the user?
• Are the operations and navigation of the user interface intuitive?
• What type of prerequisite information does a person need to use the product?
• Which functions of the product are “walk up and use” and which will probably require either help or written documentation?

How the questions were used will be described in chapter 7.3.

Exploratory tests are most easily conducted on a prototype with functionality like menus, buttons and fields. This allows for the user to explore the prototype in a manner similar to what it will be like in the real application.

7.1.2 Assessment test

Assessment testing is conducted further on in the development process, when the first design has been established through a previous exploratory test. If the exploratory test showed that the first design is sound then testing the implementation with an assessment test is a good idea (Rubin, 1994, chapter 2). Common tasks in an assessment test are to test actual tasks that a user would perform with the final program. The aim is to discover faults and errors in the implementation.

When compared with an exploratory test the test monitor’s interaction with the interviewer is reduced although the aim is still to gather information.

7.1.3 Validation test

Validation testing is the last of the tests before the application is released. It is intended to test the application’s usability.

The objective of this type of test is to determine how the application compares with a predetermined usability standard. This standard can either be a company standard, historical standard or as is the case in this project a project-related
standard. If the application meets the standard’s criteria of usability it is ready to be released, if it does not the reasons must be established.

Project-related standards come from the usability objectives developed early in the project. These usability objectives in turn were developed through the surveys, interviews and ideas put forward by the developer (Rubin, 1994, chapter 2). Rubin (1994, chapter 2) describes usability objectives stated in terms of performance criteria, this can be speed and accuracy; meaning how fast a user can perform certain operations and how many errors are made during the task. Usability objectives can also be described in terms of preference criteria, through questions that ask the users to rate the different parts of the application as well as the whole.

Another objective of the validation test is to make sure that there are no unwanted surprises hidden in the application that might cause trouble when it is released, so called “disaster insurance”.

The validation test is similar to the assessment test except for three differences (Rubin, chapter 2, 1994):

- The participants are given tasks to perform with either very little or no interaction with a test monitor. The test monitor’s role is to observe the test.
- The collection of measurable data is central to the test.
- Before the test is conducted standards for the tasks of the test are either developed or identified.

This last difference means that when comparing user performance with a standard, you need to determine in advance what is to be acceptable performance and not acceptable. It is also important in advance to decide what will be done if the standards are not met.

### 7.2 The usability tests

Setting up the usability tests included writing a general test plan (see appendix 11.3 or chapter 7.3 for a detailed description) for all three types of tests, this plan included descriptions of the purpose, problem statements, user profile, methodology, test environment, test monitor role and evaluation measures of the test (Rubin, 1994). The plan also included a questionnaire and a list of issues to consider for the test monitor when observing the tests.

The test plan was used for all three types of tests; the difference between how the plan was used depended on the role of the test monitor. Starting with: active participation in the exploratory test, a more observing role in the assessment test and finally a distant observing role in the validation tests.

Before the tests started the test monitor read an orientation script (see appendix 11.2) to the participants. This orientation script gave the participants information about what was going to happen and what was expected of them. It also reassured them that the test was aimed at testing the program not their performance.

Next follows descriptions of one exploratory test, two assessment tests and two validation tests.
7.2.1 Exploratory test

This first test was an exploratory test, therefore I used what is called a “super user” (Rubin, 1994). That is a user who is very experienced with programs. With this kind of user you can find severe problems fast. Since the user is experienced he/she usually will not get stuck on smaller problems.

The test was conducted according to the test plan. Two video recordings of the session were made, one from the reviewer’s side and one from the writer’s side. The reason for this was that it was not possible to observe both participants at the same time. A log file was also saved including the dialog from the chat window. The recordings and the log were used for later reviewing of what took place during the test. The reason for this is that it is not possible to catch every bit of useful information during the test and thus there is a need to be able to review the test session later. It was also important to be able to keep a video record of how the program was used when changes to the prototype were to be done. This could help in deciding which parts needed changing and which parts did not.

The role of reviewer was taken by the “super user” and I myself took the role of writer. Due to lack of time only one exploratory test was conducted.

7.2.2 Assessment test

The second and third tests were conducted as assessment tests (Rubin, 1994) of the prototype. The idea was to sit with the participants and let them try to explain what they saw and what they did while using the program to correct a text. This method is called “thinking aloud” (Rubin, 1994, p.217). The participants were a teacher in English and a retired Swedish teacher.

The test started with the test monitor explaining what was expected from the participant. Next the participant tried to understand how to use the prototype to correct a text without the help of the test monitor. The text was an English text and a Swedish text with some deliberately entered errors. The reason being to test how well the participant could understand the graphical user interface rather than doing a full validation test.

The participant then tried to explain the different actions he took while using the prototype. If the participant needed help he was first supposed to read the help menu and then ask the test monitor if anything was still unclear.

7.2.3 Validation test

The fourth and fifth tests were validation tests aimed at testing the usability of the prototype which was done through comparison with the usability standards. They were based on the research earlier in the project up until the assessment testing. Now they were used in the validation tests. The usability standards for this project were the following:

1. The number of errors made on each task should not exceed one error. This implies that the program must be very easy to learn and understand how to use.
2. The results from questions 5-8 in the questionnaire are graded from 1 to 5 where 1 is easy to use and 5 difficult. The standard was chosen to lie between 1 and 2.
3. The amount of help needed to get started using the program for the first time should not exceed two or three general questions answered. A common question was for example: How do you use this button (pointing at a button)?

The validation tests started in the same manner as the exploratory test and the assessment test, the participants were told what was going to happen and what the prototype was used for.

In the first test the text was in Swedish written by a student studying Swedish as a second language. The text was three pages long, thus it was longer than intended. This posed no problem for the teacher who was a retired teacher in Swedish. I decided that for this test the test monitor should observe the reviewer side, hence he stood next to the teacher. In the meantime a program called CamStudio was running on both sides, recording the session as two audio video files (avi). This made it possible to review the films afterwards on the computer.

For the second test the one page long text was written in Spanish, and also written by a second-language student. The teacher was a student at Stockholm University studying to become a teacher in Spanish.

In this test the test monitor observed the writer side, thus he stood next to the student. The session was recorded in the same way as in the first test using CamStudio.

In these last two validation tests I took the role of test monitor, unfortunately I could not get hold of anyone to help me observe the other side of the test. Thus I switched between observing the reviewer and in the last test the writer.

### 7.3 Contents of the test plan

#### 7.3.1 Purpose of the usability tests

The reason for conducting the tests was to investigate:

- How well the end users could interact with the prototype and review texts with it.
- Which activities were different from the ones in face-to-face communication.
- Which new problems arose.
- Which ones from face-to-face communication still existed.
- Which ones were solved.

The choice of using three different test methods was based on the notion that for the first test no previous knowledge was available. I believed that it was more valuable to get feedback such as the thoughts of the participant rather than testing the prototype with a validation test. Thus I chose a progressive approach to the usability tests, progressive in the meaning that the participant’s role changed from active participant in the development during the exploratory test to a passive role in the development during the validation test.

The prototype application was modified in accordance with the results from each test. This meant that each pair of test subjects was faced with a slightly different version of the prototype and hence, the tests were part of the ongoing development process.
7.3.2 Questionnaire
Apart from basic questions about the participant’s background, age and sex the following questions were used:

On a scale from 1-5, where 1 is easy and 5 difficult.

1. Are the functions of the program easy to use?
2. How difficult was it to use the chat for communicating with the other person?
3. (Author question) When following the reviewers corrections on the screen, how hard was it to find where in the text the reviewer made corrections?
4. (Reviewer question) How difficult was it to use the built in functions for marking corrections?
5. What problems did you encounter while using the program for reviewing?
6. Did you miss hearing the text being read out loud while reviewing, as is usually done face-to-face?
7. When you wanted to say something to the other person and had to use the chat to get their attention, was that a problem?
8. In general what do you think of the graphical user interface?

The first four graded questions were used to gather information about how difficult or easy it was to use the prototype. The remaining questions were used to collect comments from the participants. These comments were valuable sources of information when it was time to create a new version of the prototype.

7.3.3 Observation list
The following points were used by the test monitor in the same manner as the questions for the case studies:

- How do the participants handle confirming that they understood each other?
- How do the participants handle disagreements?
- How do the participants get each others attention?
- How does the reviewer mark the text that was corrected?
- Do the participants follow the built in function for finding where the corrections were being made in the text?
- Do the participants develop their own way of referring to the location in the text?
- What does the user need to know before being able to use the system?
- Which parts of the GUI need further explaining?
- Where, when and what parts of the GUI causes breakdowns in the reviewing process?

These points served as a guide to what was to be observed during the tests. By using similar questions for the usability tests and for the laboratory experiments it was later possible to compare the results. The results were a valuable aid in the development process. The reason was that they showed what needed to be changed in order to better support the observed behaviour of the test subjects.
7.3.4 User profile

A total of five pairs of author reviewer participants were tested, one pair in the exploratory test, two pairs in the assessment tests and two pairs in the validation tests, totalling 10 people. Due to problems finding voluntary participants only two women were included. The age of the participants ranged from 21 to 71 years, the majority was approximately 25 years old.

Common background was that they all had studied second language courses. Five of them also had experience with teaching a second language. 8 out of 10 participants were experienced internet users (more than two years of experience). All of them had been using a computer regularly for the last two years or more.

Choosing participants was not done in random, rather they were chosen according to their backgrounds. This choice was made based on the idea of making qualitative usability tests instead of quantitative. If the prototype will be further developed then tests with randomly chosen participants will be conducted.

7.3.5 Methodology

The usability tests consisted of three parts:

1. The test started with gathering of background information through the questionnaire (appendix 11.3.2.1). Then followed an introduction were the test monitor explained what would happen next, what the program was, what it supported and how they were to use it. This was intended to focus the test on the reviewing part.
2. The second part consisted of the reviewing session which was monitored directly by the test monitor and also recorded with a program called CamStudio (FreeWare). This program allows you to record anything that is shown on the screen. The reviewing was done using a text approximately one to two pages long.
3. The third part consisted of the participants answering the rest of the questions in the questionnaire.

7.3.6 Test environment/equipment

The test was conducted in the participants home, thus both the writer and reviewer were situated in a familiar environment since they each sat in front of their own computer. The test monitor resided with the reviewer or the writer. The only equipment needed was a computer running Windows with an internet connection, preferably a fast computer due to the overhead of running CamStudio while testing the prototype.

7.3.7 Test monitor role

The role of test monitor, i.e. the person who leads the test, was taken by me. The test monitor observed each one of the participants while the writer and reviewer interacted with the program. The test monitor also gave the participants instructions on how to use the program before starting the tests. Depending on the type of test the test monitor gave the participants different help with their questions. In the two validation tests he did not answer questions about how to use the prototype, in accordance with how a validation test is conducted.
8 Results from the Usability Tests

The results from the usability tests are not statistically valid since this was not the intention. Rather they should be viewed as part of the development process where the next step would be to create a new version of the prototype. The new prototype would then be tested with a larger group of participants before a possible final version could be released.

8.1 Exploratory test

8.1.1 Reviewer results

Comment 1:
“When you wanted to say something to the other person and had to use the chat to get their attention, was that a problem? Answer: “Yes, I missed some function for getting the other persons attention. “

Comment 2:
“Where, when and what parts of the graphical user interface (GUI) causes breakdowns in the reviewing process? Answer: “The chat distracted me while reviewing.”

These are two comments from the reviewer that were interesting. Since this was only a preliminary result from one user this could not be said to hold for all users.

The first comment shows that when using the chat there can be a problem getting the other persons attention. This is not surprisingly considering that the user is more familiar with reviewing texts face-to-face. When using only a text-based chat there is no other way of acknowledging the other persons presence than through writing.

The second comment is also about the chat, the user found it distracting while reviewing the text. It was not possible to tell this early in the test phase if this was just a result of this user’s preference for working undisturbed or if this was a problem in the user interface. Thus I chose to wait and observe the other users’ reactions to the chat.

8.1.2 Author results

These three comments are my own thoughts after using the program live for the first time.

I believe adding the same feature for finding the location in the corrected text to the original text in a future version of the prototype would remove the problems discussed in comments one and three.

Comment 1:
What problems did you encounter while using the program for reviewing?
Answer: I found it a bit difficult to find which word or sentence in the original text that had been corrected.

Comment 2:
Did you miss hearing the text being read out loud while reviewing, as is usually done face-to-face?
Answer: Not really, but it felt a bit strange to me not to hear but only see the corrections appear on the screen.
Comment 3:
Where, when and what parts of the GUI causes breakdowns in the reviewing process?
Answer: Not being able to find the location of the corrected word or sentence fast in the original text meant extra time had to be taken to read through the original text.

8.2 Assessment test 1

8.2.1 Reviewer results
The following is a list of suggestions for changes to the prototype put forward by the user. Some were implemented others such as the instructions part are planned functions.

- Text field labels: Change “Please write here” to “Please write here and press enter”, change “Original text” to “Author’s original text” for the reviewer and “Author: Paste your text here” for the author, change “Chat output for” to “Text area for messages” and “User's name”.
- Button changes: Change “Send corrections” to “Send revised text to author”, change “Choose markings” to “Options for text marking”, disable “Send original text” button for reviewer.
- Menu changes: Change “Help” to “Instructions”.
- Change tool tip help to window with instructions and write a better help text.
- Create startup information, for example “Welcome to ….”
- Create password field when starting program
- Change width of program window, make it wider.

8.3 Assessment test 2
The second assessment test was conducted with a group of people during a small presentation of the prototype. The following comments were either implemented directly in the prototype or added to the list of planned functions. (They were originally written in Swedish)
Comment 1: Transcript from the demonstration.
“ola: yes good, but I would rather have comments than corrections.
ola: It would look good if for example a teacher could annotate the text with his symbols.
ola: You could imagine a panel with symbols from which you could choose what to use to mark the text.”
Comment 2:
“Would it be possible to extend the function for marking the text? More options such as larger text and different colours.”
Comment 3:
“Would it be better to update continuously or not?”

8.4 Validation test 1
The following results from the validation tests shows that the chat was used as expected: as a substitute for speech. The results also show that the possible problem of lacking co-location was solved with the implemented function for WYSIWIS.
8.4.1 Observation results

1. How did the participants handle confirming that they understood each other?
   Answer: They both confirmed by writing “ok”.

2. How did the participants handle disagreements?
   Answer: If the author disagreed on a correction made, they started a small discussion using the chat until agreement was reached. Compared with handling disagreements face-to-face, the discussion through chatting was short (subjective judgement of test monitor).

3. How did the participants get each others attention?
   Answer: Through the chat, in this test it was not an issue since both participants were alert all the time.

4. How did the reviewer mark the text that was corrected?
   Answer: The reviewer used red colour for marking changes made, blue colour for marking changes to be made by the author and green colour when he was uncertain on the meaning of a word or sentence.

5. Did the participants follow the built-in function for finding where in the text the corrections were being made?
   Answer: Yes, the author found it easy to follow the corrections as they were sent to him.

6. Did the participants develop their own way of referring to the location in the text?
   Answer: There was no need.

7. What does the user need to know before being able to use the system?
   Answer: Both participants were given a short summary of the different functions of the program.

8. Which parts of the GUI need further explaining?
   Answer: No further help was given and none was asked for during the test session.

9. Where, when and what parts of the GUI causes breakdowns in the reviewing process?
   Answer: The author found that he had to wait a long time while his text was being reviewed. Example from dialog:
   reviewer sent you a correction
   author: ok
   author: du har bara korrigerat första stycket va?
   reviewer sent you a correction
   reviwer: ja.. jag skickar dig allteftersom jag blir klar...
   author: ok
8.4.2 Answers to questionnaire

The answers from one author are included here for reference. In the next chapter in which I give the results from the second validation test the answers from one reviewer are included. The answers were translated from Swedish to English to increase readability.

1. Male [x] Female [ ]
2. Age: 22
3. How many years have you studied second languages?
   Answer: 4 years
4. Author [x] Reviewer [ ]
5. On a scale from 1-5, where 1 is easy and 5 difficult, are the functions of the program easy to use?
   Answer: 1
6. On a scale from 1-5, where 1 is easy and 5 difficult, how difficult was it to use the chat for communicating with the other person?
   Answer: 1
7. (Author question) On a scale from 1-5, where 1 is easy and 5 difficult, when following the reviewers corrections on the screen, how hard was it to find where in the text the reviewer made corrections?
   Answer: 2
8. (Reviewer’s question) On a scale from 1-5, where 1 is easy and 5 difficult, how difficult was it to use the built in functions for marking corrections?
   Answer: Not applicable
9. Did you miss hearing the text being read out loud while reviewing, as is usually done face-to-face?
   Answer: No
10. Was it a problem when you wanted to say something to the other person and had to use the chat to get his attention?
    Answer: No
11. In general, what do you think of the program?
    Answer: It is a good program, but it requires that you are used to using computers. It also requires patience, since sometimes you have to wait a long time. I learned more this way compared to reading a corrected text sent through email.

8.5 Validation test 2

8.5.1 Observation results

1. How did the participants handle confirming that they understood each other?
   Answer: They did not need to confirm, since the text was very short and easy to understand.
2. How did the participants handle disagreements?
   Answer: None came up.

3. How did the participants get each others attention?
   Answer: Through the chat and through comments marked with red in the edit window.

4. How did the reviewer mark the text that was corrected?
   Answer: With red and bold text.

5. Did the participants follow the built in function for finding where in the text the corrections were being made?
   Answer: Yes, the function was found to be easy to understand.

6. Did the participants develop their own way of referring to the location in the text?
   Answer: No, no need.

7. What does the user need to know before being able to use the system?
   Answer: The participants were told that the program is used for reviewing texts and told to explore the functionality. They only needed confirmation from the test monitor that it was ok to test the different buttons and fields.

8. Which parts of the GUI need further explaining?
   Answer: The buttons for marking the text were not crystal clear to the reviewer.

8.5.2 Answers to questionnaire

1. Male [X ] Female [ ]
2. Age: 24
3. How many years have you studied a second language? Answer: 1 year
4. Author [ ] Reviewer [X ]
5. On a scale from 1-5, where 1 is easy and 5 difficult, are the functions of the program easy to use?
   Answer: 1

6. On a scale from 1-5, where 1 is easy and 5 difficult, how difficult was it to use the chat for communicating with the other person?
   Answer: 2

7. (Author’s question) On a scale from 1-5, where 1 is easy and 5 difficult, when following the reviewers corrections on the screen, how hard was it to find where in the text the reviewer made corrections?
   Answer:

8. (Reviewer’s question) On a scale from 1-5, where 1 is easy and 5 difficult, how difficult was it to use the built in functions for marking corrections?
   Answer: 3

9. What problems did you encounter while using the program for reviewing?
Answer: I don’t think it is clear what the button “options for text marking” is used for. At first glance I thought it was to mark the text with the options. The “mark text” button was not clear either, my first thought was: what does it do? Does it mark the text like in a word processor?

10. Did you miss hearing the text being read out loud while reviewing, as is usually done face-to-face?
   Answer: No

11. Was it a problem when you wanted to say something to the other person and had to use the chat to get his attention?
   Answer: No

12. In general what do you think of the graphical user interface?
   Answer: I think the program is very easy to use, it felt like I knew how to use all the functions after only having used the program for a short while. In comparison with many other programmes that have functions that are almost never used.

9 Conclusions

9.1 Sub Goals
The results from the field study and laboratory experiments fulfilled the second and third sub goals set up earlier (shown below for reference):

- Learn how ordinary text reviewing is conducted: how a reviewer reads, corrects and communicates with the writer.
- Learn the writer’s requirements as well as the reviewer’s.

The results from the usability tests show that the prototype is fully functional and could be used for reviewing texts. This conclusion is based on the fact that the results show that the prototype met the usability standards. The results also show that the prototype could be further developed to support more of the user’s requirements as stated in chapter 6.3 Planned Functionality.

Compared with reviewing face-to-face there was one problem that can not be solved in a text-based chat system. This was the fact that while the reviewer read a sentence or part of the text the author did not know what he was doing and thus the experienced waiting time felt longer than in face-to-face reviewing.

9.2 Project conclusions
The main question for this master’s project was: How do you allow a reviewer to review texts from one or several writers in a setting where the reviewer is not located in the same room as the writer or writers?

Apart from answering the above question the aim of the project was to develop an application which would facilitate cooperation between writers and reviewers. The resulting prototype is in some aspects a complete novelty – for example in being able to be used through a webpage instead of having to install it on a computer. In other aspects the prototype uses functions that can be found in existing applications although not in the exact same manner, for example the synchronous reviewing function.
It is important to remember that synchronous reviewing is not the most common form of reviewing texts. It is still more common that a reviewer goes through a text on his own, writes his comments and then hands the text to the writer. Synchronous reviewing has both advantages and disadvantages compared with asynchronous reviewing. Advantages are that the writer can get his text reviewed without having to wait and the writer also can learn from the revision process, especially in a second language course. This was observed during one of the usability tests of the prototype as the writer later told me that he found it educating to be able to follow the revision process. One disadvantage for the reviewer is that he must both read and comment the text while the writer is waiting and might feel stressed, especially if the text is long. This is also a disadvantage for the writer who waits and gets impatient since he cannot follow when the reviewer reads the text. I have implemented a save function and a function to save the ongoing conversation to compensate for this disadvantage. These functions allow the reviewer and writer to take a break and continue later.

Reading through this report might give the impression of a very structured way of working, which is partly correct. Due to my lack of experience in research and report writing I started out reading the literature and searched for methods and techniques which seemed to fit my expectations of what I needed to do. Now that the report and research are finished I have learnt that the methods I had chosen from the beginning would have been better suited for a larger project.

Finding test subjects was not as easy as I had expected from the start and thus the test subjects who participated in the two studies and in the usability tests were family members, friends and students at KTH. Although they fitted the target group a future version of the prototype developed and tested with test subjects chosen in a more random fashion would be beneficial. This could for example be tests conducted in university courses in second languages or Komvux (high school for adults) courses. A possible test scenario could be to let the teacher in a second language course show typical writing errors to the class using the prototype as a medium. The teacher could be located in his office or even at home and the students in their homes while the teacher displays a text written by a student and reviews and corrects it synchronously showing and explaining to the class what needs to be corrected.

During the course of this project it became clear to me that this type of research seldom has one correct answer to a question and that it is not always the answers that are interesting but what is revealed during the research.
10 References

- Rodríguez H., 2003, Designing, evaluating and exploring Web-based tools for collaborative annotation of documents, Department of Numerical Analyses and Computer Science, Royal Institute of Technology.
- Rodríguez H., 2002, Web annotations systems supporting collaboration, research report, Department of Numerical Analyses and Computer Science, Royal Institute of Technology.
- Rubin J, 1994, Handbook of Usability Testing, Wiley, United States
11 Appendix

11.1 Research design for observing interactions between teacher and student

11.1.1 Practical preparations
Three groups consisting of one teacher and one student are to be observed while reviewing a text. The text will be 1-2 pages long. With the consent of the test subjects these three sessions will be videotaped for later analyses. The sessions will all take place in a “laboratory” environment, The teacher and student will thus not be in an ordinary classroom rather in an office room.

11.1.2 Goals
• Investigate what activities take place between teacher and student when reviewing a text.
• Identify problems with text reviewing face-to-face
• Observe collaboration between teacher and student

11.1.3 Methods
• Behavioural observation (Kerlinger & Lee, 2000, chapter. 31, p. 727)
• Remembered observation (Kerlinger & Lee, 2000, chapter. 31, p. 740)

11.1.4 Instruments
• Videotaping
• Pencil and paper

11.1.5 Suggestions for observation
1. How do the teacher and student refer to where in the text they are currently correcting?
2. Does the teacher use any device to point at the text?
3. How does the teacher correct the text?
4. What kind of markers does the teacher use for corrections?
5. What happens when the student doesn’t understand the teacher?
6. What happens when the teacher doesn’t understand the student?
7. Does the teacher wait for confirmation from the student that he/she has understood before proceeding?
8. If the student disagrees with the teacher over some correction made, how do they resolve the disagreement?
9. How does the teacher get the students attention when reviewing the corrections made?
11.2 Orientation Script

Thank you for participating in this test which is completely voluntary and can be ended whenever you want to. The main stages of the test are the following:

1. Introduction with background to the test, questionnaire to gather background information about your language level and instructions for use of the prototype.
2. Test session where you review a text together with the writer or reviewer depending on your role.
3. Evaluation of the prototype through a questionnaire in which you are asked to fill out flaws, errors and problems caused by the prototype’s design during the test session.
11.3 General usability test plan

11.3.1 Purpose
The reason for conducting this test is to investigate:

- How well the end users can interact with the prototype and review texts with it.
- Which activities that are different from the ones in face-to-face communication.
- What new problems arise.
- Which problems from face-to-face communication still exist.
- Which problems that are solved.

This is then compared with my earlier observations of text reviewing on paper.

The time it takes to complete a revision, learn how to use the program and recover from errors will be used as measurements together with the participants’ answers to the questionnaire.

The results from this test will be analysed and new versions of the program developed pending the outcome. The results will also be used as a base for further research in synchronous text reviewing on the web.

11.3.2 Problem statements

11.3.2.1 Questionnaire
1. Male [ ] Female [ ]
2. Age: _ _
3. How many years have you studied second languages?
4. Author [ ] Reviewer [ ]
5. On a scale from 1-5, where 1 is easy and 5 difficult, are the functions of the program easy to use?
6. On a scale from 1-5, where 1 is easy and 5 difficult, how difficult was it to use the chat for communicating with the other person?
7. (Author’s question) On a scale from 1-5, where 1 is easy and 5 difficult, when following the reviewers corrections on the screen, how hard was it to find where in the text the reviewer made corrections?
8. (Reviewer’s question) On a scale from 1-5, where 1 is easy and 5 difficult, how difficult was it to use the built in functions for marking corrections?
9. What problems did you encounter while using the program for reviewing?
10. Did you miss hearing the text being read out loud while reviewing, as is usually done face-to-face?
11. Was it a problem when you wanted to say something to the other person and had to use the chat to get his attention?
12. In general what do you think of the graphical user interface?

11.3.2.2 Observation part
- How did the participants handle confirming that they understood each other?
- How did the participants handle disagreements?
- How did the participants get each others attention?
- How did the reviewer mark the text that was corrected?
- Did the participants follow the built-in function for finding where in the text the corrections were being made?
- Did the participants develop their own way of referring to the location in the text?
- What does the user need to know before being able to use the system?
- Which parts of the GUI need further explaining?
- Where, when and what parts of the GUI causes breakdowns in the reviewing process?

11.3.3 User profile
A total of five pairs of author-reviewer participants will be tested, totalling ten participants. Common background will be that they all have studied second language courses. Five of them will also have experience with teaching a second language.

11.3.4 Methodology
The usability test will consist of three parts:
1. The test starts with the gathering of background information through the questionnaire. Then follows an introduction where the test monitor will explain what will happen next, what the program is, what it supports and how to use it. This is intended to focus the test on the reviewing part and not on how easy the program is to learn to use.
2. The second part consists of the reviewing session which will be monitored directly by the test monitor and also recorded with a program called CamStudio (FreeWare). This program allows you to record anything that shows on the screen. The text to be reviewed will be approximately one page long.
3. The third part consists of the participants filling out the questionnaire.

11.3.5 Test environment/equipment
The test will be conducted in the participants’ home, thus both the author and reviewer will be in a familiar environment. The test monitor will reside with the reviewer. The only equipment needed will be a computer running Windows with an internet connection.

11.3.6 Test monitor role
The test monitor will observe one of the participants, the reviewer, while the author and reviewer interact with the program. The test monitor will also give the participants instructions on how to use the program.
11.4 Screenshot of GUI
11.5 Equipment used in the development

Hardware and software used in the developing process:

- Programming: JBuilder 7, JDK 1.4.1
- Hardware: PC running windows XP