# Direct Connect: Evaluation of a Filesharing Application

**Jens Schou**
Göteborgs Universitet
Computing Science
jenschou@mdstud.chalmers.se

**Gustav Engström**
Chalmers Tekniska Högskola
Industriell Ekonomi
i00engu@hotmail.com

**Therese Nordqvist**
Göteborgs Universitet
Computing Science
thenor@mdstud.chalmers.se

### ABSTRACT

Designing user-friendly software interfaces enables more people to use the software. We believe that many freeware applications, such as filesharing programs, have an interface that is difficult to understand for a first-time user. Thus, many potential users are not able to use the software. In this paper, Direct Connect ++ (DC++), a filesharing application, was examined in terms of learnability, i.e. how easy it is to understand for a first-time user. By observing real users and performing a Cognitive Walkthrough, a widely used web design evaluation tool, the authors identified a number of fundamental design flaws in DC++. These include poor feedback, lack of information and need for prior knowledge. A number of suggested improvements to remove some of these design flaws are presented in this paper. We believe that by improving the user interface of DC++, many more users would be able to use the program.

### Author Keywords

File sharing, filesharing, Cognitive Walkthrough, Cooperative Evaluation, evaluation, Direct Connect, DC++, HCI.

### ACM Classification Keywords

H.5.2 User Interfaces: Evaluation/methodology.

### INTRODUCTION

Many interfaces have been designed without putting the user in focus. This holds true especially for web-based applications. Designing user-friendly interfaces first came to the developers' attention with the commercial web-based applications. Today it is widely recognized that business-to-consumer systems must facilitate easy navigation and usability to retain customers [1].

Freeware are applications that are not driven by commercial interests. Some of the most widely used freeware programs are filesharing programs [2] such as Kazaa and DC++. This paper examines the most popular filesharing program, DC++ [3], and aims at suggesting improvements enabling more people to use the application; using general usability design principles we will look at the learnability aspect of the design of DC++ [4].

Learnability can be defined as how a system allows novice users to use it and attain an optimal level of performance [4]. Usability is a broader term, often referring to how user-centered and easy to use a design is [4].

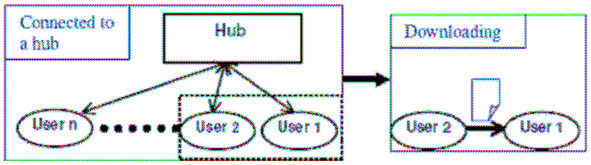


**Figure 1. Connecting to a hub, then downloading a file from another user. Notice that transfers are made directly between user clients, not through the hub.**

### Filesharing Concepts

Filesharing can be defined as the activity of making files available to other users for downloads over the Internet. Filesharing is often a peer-to-peer system where files are stored on personal computers and provided to linked users. The filesharing applications simply provide solutions for Internet users to get together and form networks that enable them to share files. Many modern filesharing applications, among them DC++, use the concept of hubs or super nodes to facilitate the networking [3]. The hub is the central computer of a filesharing network. When a DC++ user wants to share files he must allow his computer to be a server to a hub. By doing so, he will be able to download files from other users connected to that specific hub. Also, specified directories of his hard disc will be available to those same users. Files are not downloaded from the hub but from other individual users. The hub simply provides the route to the user and does not have to store any of the files available on the network (see Figure 1)[5].

### Objectives

The objective of this paper is to evaluate the user interface of DC++ in terms of learnability. What defines our study in relation to previous work done in the area is the different focus. Learnability of commercial software has been studied many times before. Also, non-commercial software has been studied with a different usability focus. This study is to our knowledge the first that explicitly focuses on the learnability of non-commercial software. Choosing learnability is motivated as it is one of the most important design principles of human-computer interaction [4]. Examining DC++ is particularly interesting as it is an software application with millions of users with various computer experiences.

We want to see if and how the design of DC++ distances the application from non-experienced users. Where other studies have focused on general usability issues [6] or specific issues not explicitly related to learnability [7], we want to use a "commercial" perspective on a non-commercial product. That is, we want to identify what could be done to maximize the number of users. Because of this perspective, we have chosen to study very basic functions of the application.

### Previous Research

Earlier research shows that many web-based applications are lacking in both general usability and learnability [1]. Most of this research has been conducted with commercial applications in mind. However, there have been a few studies on freeware applications. From our experience it seems that the focus of those studies tends to be on usability aspects other than learnability. Arguably, this is because of the initial view of the intended user. When they emerged, many freeware applications (including filesharing) were used only by computer experts. Naturally, the user interfaces were then constructed with the expert user in mind [6]. Today, there has been a dramatic shift in user characteristic as millions of users are using different freeware applications on an everyday basis. Also, the usability of filesharing has been studied before [7]. This study was conducted using Kazaa, the most widely used filesharing application some years ago. The conclusions drawn were that Kazaa had serious usability flaws. The focus of the study was the lack of security in filesharing applications and poor usability was blamed as the number one reason [7].

## EXPERIMENT

Good design should concentrate on the tasks that need to be accomplished for the user to reach his or her goals [1]. Cognitive Walkthrough (CW) is a usability inspection method that has proven useful in designing web-based applications [7]. Both of these statements are good reasons to use CW in our study, however; CW assumes the user knows what to do on a step-by-step basis. The user of a filesharing program might have the goal of downloading a song, but might not be aware of the concept of a hub or file searching. In such a case the user might have problems translating his goal into a valid action sequence.

We believe that using only CW makes a reliable but maybe not a valid study. In our study CW is conducted by experts taking the role of inexperienced users. Including real users in the study should increase its validity. The solution was to execute both a CW and a Cooperative Evaluation with a number of users, observing how they managed to navigate through the program.

We believe that the combination of analysis through CW and the user perspective of Cooperative Evaluation, leads to a higher level of reliability and validity than any of the two methods could do on their own.

### Experiment Design

Since we had decided on using two different evaluation techniques we first had to decide upon some common ground to base our evaluation on, to ensure that our two methods would be evaluating the same thing. The following is a short description of the three most basic items in our evaluation, and how we decided their requirements.

- System to evaluate:

  *DC++ v0.670*

  Two members of our team were already familiar with filesharing and the application DC++ in particular. We therefore decided to use the newest version of DC++.

- Intended user:

  *A person familiar with computers running Microsoft Windows, but even though he has heard about them, is unfamiliar with filesharing programs.*

  Filesharing is quite an advanced task, that won't be attempted by many persons new to the Internet [8]. However it is more interesting to evaluate how a user unfamiliar with a system would use it. The reason you try a new system is often that you have heard others talk about it; we therefore assume our user to be new at filesharing but having heard something about the concept. Furthermore DC++ is only available for computers running some version of the Windows OS.

- Representative tasks:

  *Start the application for the first time and connect to a hub.*

  *Download a favorite song.*

  *Configure for automatic connection to a hub at next startup.*

  To get a better coverage of DC++ we decided to evaluate three distinct tasks, covering different aspects that would be important for new users. Before you can start using the application you need to set some settings and connect to a hub. The most representative task in a filesharing application is, in our opinion, to download something. Lastly we also added a task making frequent use easier.

#### *Chosen Methods*

The *Cognitive Walkthrough* [4] method simulates step-by-step user behavior. It consists of four items; three of them were already decided upon and described above, the fourth item is an action list for each task to be evaluated. We created action lists going through all the necessary steps to accomplish the tasks, choosing the most readily available option if several were present. After creating the action lists we evaluated the system by asking the four Cognitive Walkthrough questions at each step.

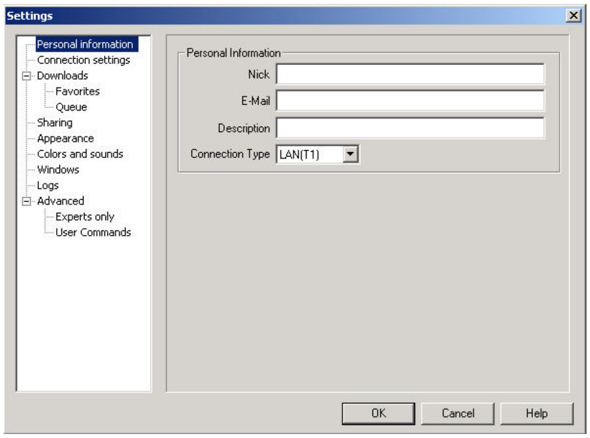
The *Cooperative Evaluation* [4] method consists of a test person using, or attempting to use, the program and an evaluator recording what the user does, in some way. The user is encouraged to talk out loud about what he or she is doing and even ask questions to create a dialog with the evaluator. The actions, mistakes, comments and questions can be recorded in different ways. We used pen and paper as a recording device. As we wished to get an idea of how the user would use the system with as little help as possible, we narrowed down the amount of help we gave. We only steered them in the right direction when they, for some time, seemed to be seeking for a certain function in the wrong place. We also encouraged them to use the built-in Help-function.

### Experiment Results

Both our Cognitive Walkthrough and our Cooperative Evaluation indicated several both small and big design flaws with an adverse effect on learnability.

As quantified by our Cognitive Walkthrough: In total we have 14 steps in all our action lists, which means we have asked one of the CW questions in total 56 times. Of the answers to these questions 18 indicate a problem of some kind. Even though some of these are just small problems some are quite big obstacles for a new user and in total it indicates that a new user would have a frustrating experience with DC++.

The major problems identified by our combined methods were: (see Table 1)



Personal Information

Nick

E-Mail

Description

Connection Type LAN(T1)

**Figure 2. DC++ settings: the first page is simple, but some necessary settings hide in the tree to the left.**

| Problem Areas | Suggested Solutions |
|---|---|
| Prior knowledge of filesharing is assumed for setting up the system. There is no indication of which settings are necessary and why. Failure to set “shared folders” leads to not being able to connect to a hub. Much time is spent considering advanced settings that are unnecessary for new users (see Figure 2). | Place the settings 'give a nickname' and 'add sharing folder' on the first page of settings, indicating that they are the only required settings.<br>A more advanced solution is to optionally launch a wizard, helping with the setup of the system. |
| Prior knowledge of filesharing is assumed to use the system. E.g. a user intending to download a file needs to realize that it's necessary to first connect to a hub, then search for the file, then download it. | When the system is used for the first time, display a brief introduction to filesharing.<br>Create a new default toolbar with the following buttons: “Public Hubs”, “Favorite Hubs” and “Search”. Constraining the available options to those most used (Norman's 5'th principle [4]). To avoid removing control from the user, the original toolbar should still be available. |
| Prior knowledge of filesharing terminology is assumed when using the provided help. | Add the above-mentioned introduction to the help. Adapt the help by removing jargon and technical terms where possible. |
| Information overload: the list of public hubs is very long. In addition, most hubs are not open for users with little or nothing to share. Consequently, connecting to a hub becomes problematic for users with little to share, or users having missed to share. | The list of public hubs should by default be filtered to only show hubs you can connect to, given your current settings (Norman's 5:th principle; Exploit the power of constraint [4]). If there are no hubs to display due to this filtering a text indicating how to solve this should be displayed. |
| There is a lack of informative feedback when executing a search. There is little to differentiate between the case when nothing is found and the case when there is a problem, e.g. not being connected to any hub. | Give clear visual feedback if there is no active connection when attempting to search. Give clear visual feedback when a search is in progress, e.g. displaying a progress bar indicating how many users have been searched so far. Give clear visual feedback when a search is completed. (Schneiderman's 3:d rule; Offer informative feedback [4]) |

**Table 1. Problems found, with corresponding suggested solutions.**

## DISCUSSION

We have evaluated the freeware application DC++ in terms of learnability. Our research has found several violations against fundamental principles of learnability in the design of the application. These include lack of feedback, need for prior knowledge, and too much information at the same time. In addition, we have tried to ameliorate the faults we found by applying principles of learnability, leading to a list of suggested improvements. We believe that implementing most of our suggestions would lead to an improvement in terms of learnability and hence usability for new users.

### Effects of the Chosen Methodology

Since we worked with two different methods at the same time we have had some information sharing between them. This has probably resulted in different results for each individual method, compared to that same method being executed as the sole method of evaluation. However we do not believe that this presents any problem for the validity of our study. Rather the extra information available because of executing two methods in tandem resulted in more faults found in both single methods. Additionally, all faults described in this paper have been found to break some fundamental principle of user-friendly design.

Our choice of expert analysis evaluation technique could be challenged; another good option would have been to use a heuristic evaluation. This might have resulted in findings of a more general nature, rather than the quite specific issues we have detected.

Due to a lack of resources we used pen and paper to record our user observations. We therefore had to rely on quick writing and memory to get down all comments and actions made by the user. If we had had the possibility to record mistakes, observations, and remarks made by the test subjects in a more efficient way, we might have found more and/or different faults in the application.

In short we feel that our method of combining two fundamentally different evaluation techniques worked very well for our study.

## SUMMARY AND OUTLOOK

The goal of our evaluation has been to find concrete improvements affecting learnability for the filesharing application DC++. We have found and suggested several such improvements.

Filesharing is quite a new concept in our society. With enthusiasts leading the way it is spreading to the general public, resulting in increased exposure. In turn this will lead to an increased familiarity with the concept. When observing our test subjects it was obvious that a lack of familiarity with filesharing was one of the biggest reasons for some of their common mistakes. We therefore believe that even without any development in the applications, new users will have less problems with filesharing as familiarity with the concept increases in our society.

The developers of DC++ regularly update their application and they have the clearly stated objective to increase their number of users [3]. In addition the application itself has a mechanism for users to supply the developers with design suggestions. It is therefore our conclusion that, even though much work still needs to be done, DC++ will continue to improve in terms of learnability and general usability.

Coupling new users increased familiarity with the concept of filesharing with improved filesharing applications in terms of learnability, we believe filesharing will continue to increase in popularity.

### Further Research

The main result of this study is the table of design faults and suggested improvements we supply in the body of this paper. If the suggestions were to be implemented, an interesting follow up research subject would be to re-evaluate the application and compare the two evaluations to find out how big an impact our suggestions really would have on learnability.

Filesharing is an activity that gains from having more users. The more users a specific filesharing application has, the higher is the chance of finding a specific file. Therefore a filesharing application should have as many users as possible. The actions of the developers of DC++ also imply that they want to increase the user base. The application is updated on a regular basis [3], their homepage states "…the simplicity of the software user interface will help you get started in no-time." [3]. Still our study shows that inexperienced users will have trouble learning to use DC++.

This discrepancy leads to an interesting research assumption; Are freeware applications less user-friendly than commercial applications are, and if they are, why is that so? Commercial software usability improvements are arguably primarily driven by commercial interests, i.e. maximizing profit. Further research on identifying mechanisms driving usability improvements in freeware is needed to answer the above question. Our evaluation of one freeware application indicates several faults in terms of learnability and hence supports the notion that freeware indeed is lacking in usability.

## ACKNOWLEDGEMENTS

This paper was originally written for a project in the Human Computer Interaction course in the spring term of 2005. It was later modified and shortened with advice from Morten Fjeld.

## REFERENCES

1. Donnelly, V. *Designing Easy-to-use Websites; A Hands-on Approach to Structuring Successful Websites.* Addison-Wesley, ISBN 0-201-67468-8. Page 9-10 & 86-94. 2001.
2. "Download.com", http://www.download.com/3101-2001_4-0-1.html?tag=dir, 2005-02-21
3. "DCPlusPlus", http://dcplusplus.sourceforge.net, 2005-02-21
4. Dix, A., Finlay, J., Abowd, G. D. & Beale, R. *Human-Computer Interaction, 3:rd edition.* Prentice-Hall, ISBN 0130-461091. Pages 225-257, 258-288 & 318-365. 2004.
5. "P2P transfers - What is P2P (peer to peer) transfer and how does it work?", http://www.p2ptransfers.com/whatisp2p.htm, 2005-02-21
6. Berglund, E. & Priestly, M. *Open-source documentation: in search of user-driven, just-in-time writing.* ACM Special Interest Group for Design of Communications. 2001. P. 132-141. ISBN: 1-58113-296-6
7. Good, N. S., Krekelberg, A. *Usability and privacy: a study of Kazaa P2P file-sharing*. Conference on Human Factors in Computing Systems. 2003. P. 137-144. ISBN: 1-58113-630-7
8. Madden, M. & Lenhart, A., www.pewinternet.org, *Pew Internet Project Data Memo*, July 2000