

# Comparing Web Applications with Desktop Applications: An Empirical Study

Paul Pop

paupo@ida.liu.se

Dept. of Computer and Information Science  
Linköping University  
Sweden

## Abstract

*In recent years, many desktop applications have been ported to the world wide web in order to reduce (multiplatform) development, distribution and maintenance costs. However, there is little data concerning the usability of web applications, and the impact of their usability on the total cost of developing and using such applications. In this paper we present a comparison of web and desktop applications from the usability point of view. The comparison is based on an empirical study that investigates the performance of a group of users on two calendaring applications: Yahoo!Calendar and Microsoft Calendar. The study shows that in the case of web applications the performance of the users is significantly reduced, mainly because of the restricted interaction mechanisms provided by current web browsers.*

## 1. Introduction

The last decade has seen an explosion of the world wide web (web, for short) and web based services. According to [3] web sites are either *documents*, *programs* or *databases*: a document style web site is, for example, the static web directory provided by yahoo.com, web sites providing ecommerce services like amazon.com are databases, and web sites like hotmail.com belong to the category of program web sites.

In this paper we are concerned with the usability of web sites that are programs, or *web applications*. A web application is an application that runs on a server and presents its services to users through a web browser. Thus, for web applications the user interface consists of web pages, and the interaction is done via hyperlinks and web forms.

The number of web applications has seen a steady increase over the last years. The open directory project at dmoz.org lists more than 200 web applications in the “Computers, Internet, WWW, Web Applications” category. Such applications include web based email, bookmark managers, databases, personal information managers, web calendars and online banking. Moreover, the number of users that use such services is also increasing. For example, the web based email service from hotmail.com has more than 40 million users.

There are two main driving forces responsible for the creation of web applications. First, Internet portals like Yahoo, MSN and Netscape are investing in web applications as a way to attract more “eyeballs” that are converted to revenue through advertisements. Second, companies and organizations are implementing web applications as a way of reducing development, distribution and maintenance costs of *desktop applications*, costs that are known to be very high, especially in the case of multiplatform applications. Desktop applications are applications build according to the current WIMP and direct manipulation paradigms.

There has been a large amount of work on that evaluates the usability of desktop applications. For a survey of the work in this area the reader is referred to [4, 11]. In comparison, little amount of work has been done that evaluates the usability of web services. There are several attempts in this direction [5, 6, 10], but few scientific results and data are available. In [1, 2] guidelines for document style web sites are presented based on empirical studies. Database-like web services are addressed in [5, 14] that evaluate the usability of several ecommerce services.

Web applications have been addressed by [7, 12, 13] that describe this class of applications, and present their advantages and disadvantages. These works also acknowledge the lack of usability data and design methods for such applications. Previous results on hypertext and hypermedia [8] can potentially be of use in the case of web applications. Metrics to evaluate the usability of hypertext systems are presented in [8, 9]. Unfortunately, the usefulness of such results for web applications is greatly restricted by the fact that the work in this area has focused on optimizing a given hypertext system, sometimes in comparison with other hypertext systems. Also, another direction in this area has been to show that certain classes of applications, augmented with hypertext systems, offer improved usability.

Our aim in this paper is to compare web applications with desktop applications from the point of view of usability, measured as the user performance on a given set of tasks. Such a study, not only provides a measure of the performance degradation when switching from a desktop based application to a web based one, but can also be the source of guidelines for designing web applications and methods of efficiently mapping existing desktop applications into

web applications.

For the web and desktop applications comparison, we have selected a calendaring application, and conducted a pilot study on both the web and desktop based versions. We have been interested to measure the performance of a group of users on a number of significant tasks. The study has provided data on the performance degradation, discovered a number of usability problems of both the desktop and the web applications, and offered an insight into the factors that reduce the usability of web applications.

This paper is divided as follows. In the next section we presents the applications we have selected for comparison. Section 3 presents the study setup, and Section 4 introduces our results. A discussion of the findings is presented in Section 5, and the last section presents our conclusions and ideas for future work.

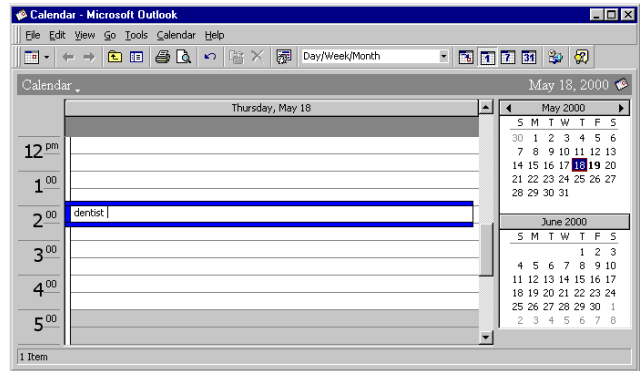
## 2. The Two Calendaring Applications

As described in the introduction, there is a large number of web applications currently in use. Our main goals in selecting an application were related to its cost and size. Thus, we were interested in applications that are either available for free in both forms, or our organization has licences available for them. In terms of size, we searched for applications that are large enough to be able to perform a meaningful study, but not too large, in order to limit the influence of features not addressed in the study on the overall performance of users.

We have found that a calendaring application meets our requirements, and have chosen Microsoft Calendar [15] as the desktop application and Yahoo!Calendar [17] as the web application to be evaluated. The purpose of a calendaring application is to provide support for the user to organize his or her meetings, appointments and other such events. More advanced features like “meeting planner”, “sharing” and “to do lists” were ignored for this study. The main tasks of such a calendaring application are:

- Provide an overview of a day, week, or year.
- Add, delete and move simple as well as recurring events.
- Find free slots for events, and events based on their name.
- Offer reminders for important events.
- Meeting planners.
- Sharing of events with other members of a group.
- Help in managing a “to do list”.

**Microsoft Calendar.** Microsoft Calendar (MC), presented in Figure 1, is a component of Outlook which is part of the Office suite from Microsoft. Arguably, this is one of the most used calendaring applications on desktop computers, mainly because it comes bundled together with the



1. The Microsoft Calendar Application

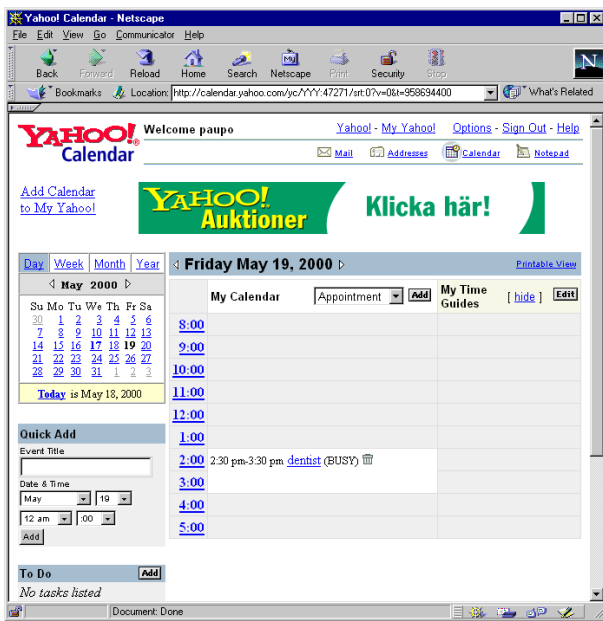
Microsoft Office, the most popular office suite.

From the extensive list of features offered by MC, those presented next are of interest because they make full use of the underlying interaction paradigms of desktop applications. Thus, the creation of events is possible simply by directing the mouse pointer and clicking in the desired time location, followed by text input using the keyboard. This straightforward input mechanism in MC is a typical direct manipulation interaction. Other tasks like moving and updating events, creating all-day events, recurring events, etc. also use heavily a direct style of interaction through “double-clicks” and “drag and drop”s rather than entering information in dialog boxes. However, while MC takes advantage of direct manipulation and WIMP paradigms, one drawback is that it is too loaded with features.

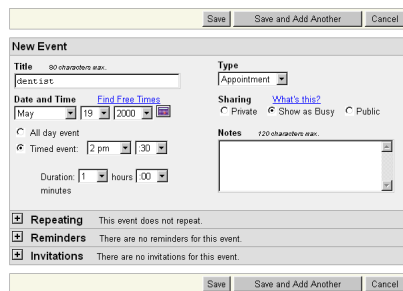
**Yahoo!Calendar<sup>1</sup>.** Yahoo!Calendar (YC), main window presented in Figure 2a, is a web application offered by the ubiquitous yahoo.com web directory. The choice of YC can be motivated by the fact the Yahoo is the leader in such web based services, its directory and other applications having the most straightforward interface. The interface of YC is simple, clean and relies only on functionality supported by all important browsers. It is possible, for example, to fully use YC without having JavaScript enabled. Another useful feature is the duplication of buttons (see Figure 2b) both at the top and at the bottom of forms in order to compensate the fact that parts of the form become invisible through scrolling. The only thing that hinders the user is the advertisement placed at the top of the window, the price users have to pay for the service.

YC can basically perform the same tasks as MC, except meetings scheduling. However, the notable difference is that all of the interaction is done through a web browser interface. Thus, the browser limits the interaction to selecting a link, using a form (inputting data, clicking a button,

1. A better choice for a web based application might have been the web based calendar from Microsoft, at [calendar.msn.com](http://calendar.msn.com), but the authors were not aware of its existence at the beginning of this study. Yahoo is, however, the market leader in this area.



a)



b)

## 2. The Yahoo!Calendar Application

drop down box, etc.), and does not permit a more direct style of interaction as in MC. For example, it is not possible to add an event just by pointing to a time, clicking and starting to write. In the case of YC, in order to add an event, we need to bring additional web page, like the one in Figure 2b, and input our data using the web form.

For the comparison of the two applications we have only considered a subset of their functionality. Section 3 details the tasks considered for performance comparison during the user study.

## 3. Empirical Study Setup

We have used an empirical study in order to determine the performance differences between desktop and web based applications.

In designing the study we have assumed that the users have no prior experience with the two applications, and thus provided them with a set of training tasks. The desired outcome of the training session is to accustom the users with

the two applications. Thus, the observers of the study have been instructed to offer help and advice whenever the users run into problems while trying to complete the training tasks. Also, while performing the training tasks in MC, the users were shown shorter ways of accomplishing the tasks through the use of more direct ways of manipulating the interface (point-and-write, selection, drag-and-drop, etc.), instead of using dialog boxes.

The training tasks used with both calendaring applications are the following:

- Add the event “dentist” to September 12, from 11am to 12.
- Delete the previously entered event “dentist”.
- Add the same event, but now from 11:15 to 12:30.
- Move this event to one hour later.
- Add a new event, entitled “lunch” from 11 to 12.
- Select and delete both events: “lunch” and “dentist”.
- Undo the previously applied delete command.
- Starting from today’s day, find a day in the week of October 16-21 when it is possible to schedule an appointment from 10am.
- Add a recurring event named “course” from 10am to 12, on every weekday in starting from September until the end of October.
- Find an event named “buy tickets” and delete that event.

The most important part of the study design is the selection of the actual tasks that are used for the performance comparison. We were interested to cover both meaningful tasks that are performed with a calendaring application, and tasks that are meaningful for our comparison. Below are presented the four tasks selected for the study. The first two are probably the most used tasks: add an event and go to a given date, while the other two tasks that have a better support within the desktop metaphor: move and undo.

- **Go To Task.** Starting from today’s date<sup>1</sup>, the users were instructed to go to the date of July 17.
- **Add Event Task.** The users had to go first to the date of June 5 (subtask not counted in the performance analysis) where they were instructed to add the event “Buy tickets” starting at 2pm, and lasting for half an hour.
- **Move Events Task.** As part of this task, the subjects were told to move two consecutive events starting at 9am to 10am. The current date was the date of June 5, that already had two such consecutive events from 9am.
- **Undo Task.** The users were instructed to undo the previous move task.

In order to determine the time per task and error frequencies, we have used the “record and playback” setting, in which the actions of users were recorded and then timing measurements were taken during the playback of each user session. Such an approach is the most suited in our case, in which is difficult to instrument the code of MC and YC in order to observe the tasks timings.

1. The study was conducted during the month of May.

A small *pilot study* with three users was conducted in order to fine tune the study setup. Several modifications to the initial study design were necessary as a result of the pilot study. We have decided to alternate the order in which the users used the two applications because when using first MC and then YC users were much more frustrated with the limitations imposed by the web browser compared to the order YC and then MC. Also, we have observed that it is very useful to have the initial calendar state as a file, and to load its contents in both MC and YC before each user session. The calendar contents were generated in order to support the tasks, the same contents being used for each user. In addition, the pilot study has discovered flaws in our record/playback setting, and thus we have finally opted for the ScreenCorder [16] application instead of Microsoft's Recorder. ScreenCorder records everything that is going on the screen as a movie file, without any perceived extra load on a Pentium II class computer. Then, we have measured the timings of the tasks using the MediaPlayer application from Windows that displays during playback the time in seconds, providing also useful features like stop and resume.

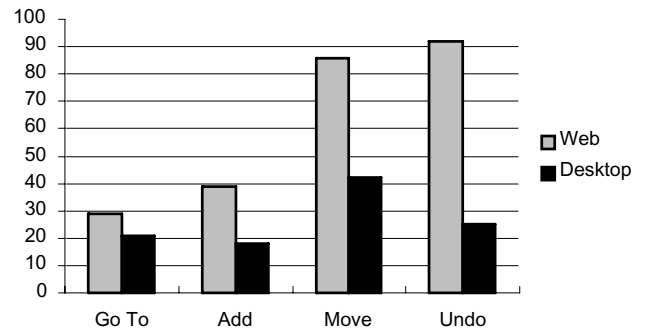
For the actual study, six subjects were recruited for the study among the authors' colleagues, five male and one female<sup>1</sup>. The users spent on average 48 minutes with the study, including the training session. The subjects were allowed to abort tasks during the study if they felt they could not complete them, or they spent more than 5 minutes on a task. At the end, the subjects were asked to fill in a questionnaire, the results of which are presented next<sup>2</sup>:

- The age of users ranged from 25 to 27 years, with a median of 26.3 years.
- All had more than 5 years experience with computers, and used computers more than 20 hours per week.
- One of them has stated that has used calendaring applications before, and three reported using services similar to the web applications considered in this paper.

## 4. Empirical Study Results<sup>3</sup>

The first result of the study concerns the user's performance in terms of time per task. Thus, Figure 3 presents the average times per task, in seconds, for each of the four tasks. On average, the users were 2.3 times slower on the web application compared with the desktop application. The times per task ranged from 18.2 to 25 seconds in the case of MC, and from 29.4 to 92.4 seconds in the case of YC. The largest difference was measured in the "Undo Task" that was

1. Under these circumstances, this study only qualifies as an extended pilot study.
2. All the users are graduate students at the Dept. of Computer and Information Science, Linköping University, Sweden.
3. No statistical analysis of the results has been done, like analysis of variance, etc., that might prove useful in identifying certain dependencies.



3. Average Time per Task, in Seconds

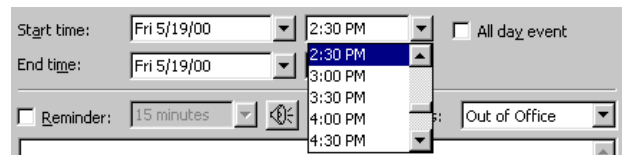
straightforward in the desktop application, while in the web application it required loading two additional web forms in order to set the previous times for the events.

Next, we were interested to compare the two applications from the point of view of error frequencies. The table below presents the number of errors for each of the tasks. A task was considered "OK" if completed successfully, resulted in "Error" if the task was completed, but the desired outcome was not obtained, and we denoted with "Abort" the case when a task was skipped by a user.

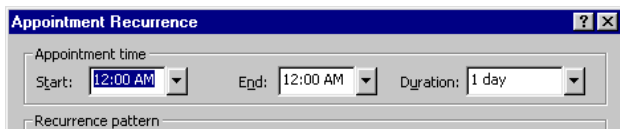
Status	Go To		Add Event		Move Events		Undo	
	MC	YC	MC	YC	MC	YC	MC	YC
OK	6	6	6	6	5	5	6	3
Error	0	0	0	0	1	1	0	2
Abort	0	0	0	0	0	0	0	1

The table shows that the first three tasks were completed successfully by the users, for both applications, with a single exception: a user misunderstood the "Move Events" task and this resulted in an "Error". The interesting case is in the "Undo" task, which all the users completed successfully in the MC, but had problems with YC. 4 out of 6 users have tried to undo the "Move Events" task using the "Back" button of the web browser, which clearly does not offer any undo facility for the YC, as it refers only to the state of the browser, and not the web application. This finally resulted in two erroneous outcomes and one abandon.

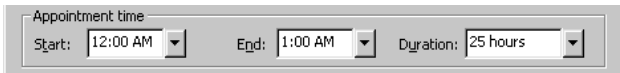
Besides the time per task and error frequency results we have also discovered some potential usability problems with the MC and YC applications. Let us consider the dialog partly depicted in Figure 4. During the training tasks



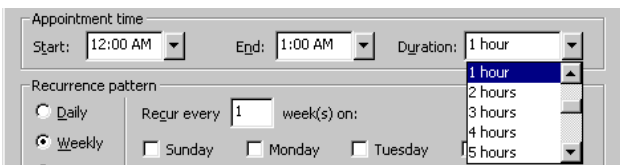
4. Problems Adding an Event at 11:15



a)



b)



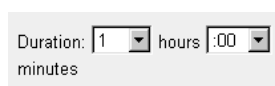
c)

## 5. Problem with the Duration of Events

two users had problems performing the training task “add event dentist from 11:15”. They complained not finding any times starting 15 minutes past, and did not realized that the control is a combobox that also allows the possibility of text input. The confusion resulted from the fact that the combobox in this case looked like a dropdown list, the same text, e.g. 12:00 AM, being in both the edit box and the list.

Another problem that confused 4 out of 6 users, is depicted in Figure 5, where the parts dialog box for the recurrent events are presented. The initial state of the dialog is presented in Figure 5a, where the duration of the event has a default value of “1 day” from 12:00 AM to 12:00 AM, next day. What puzzled users first was the need of two controls for setting the duration: “End” and “Duration”, and their relationship to each other. Moreover, another source of confusion was the fact that when changing the “End” to 1:00 AM, the duration changed to “25 hours”, instead of “1 hour”! They were able to get the desired behaviour, of an one hour event, only by using the “Duration” field. In our opinion, this problem can be easily corrected if the default value for the duration is not 24 hours, but, for example, 1 hour. Thus, when selecting a new time through the “End” control, the time will refer to the current day, and not to the next day, situation that confused the users.

The layout of web pages is partly controlled by the users of web browsers. Thus, the layout of web applications has to be very flexible. The layout of the YC interface is very simple and flexible, but problems can still be found, like the one presented in Figure 6. In Figure 6 we present a detail of a web form used for adding events (see Figure 2b). During the training tests few users were confused by the “minutes” text that lays under the “Duration” caption. This text actu-



6. Problem with the Layout in Yahoo!Calendar

ally refers to the second filed, and not to the first one as some users thought. The problem results from the size of the caption text, for sizes bigger than 12 points the text breaks ungraceful, creating the problem. A more careful layout should solve this problem.

## 5. Discussion

At the beginning of the study, we have formulated the hypothesis that the performance of users will be significantly reduced in the case of web applications. The hypothesis was confirmed, results showing a performance degradation of more than two times when switching to the web application.

In our opinion, there are three factors that contribute to the reduced performance in the case of web applications, in this order of importance:

- the limited interaction mechanisms provided by web browsers,
- the mismatch between the user’s mental model of the web application and the actual application, and
- the delays resulted from downloading the web pages from the web server.

Let us discuss in detail each of these three factors, starting with the first. The web browsers are tools designed for browsing hypertext documents. While the current commercial browsers have their problems, they are very successful at providing support for such tasks. However, they fall short when used for anything other than browsing and filling in simple forms. During the study we have observed users becoming very frustrated when going from MC to YC, and trying in YC interaction styles that are not supported by browsers. The second factor deals with the user mental model. Thus, the users do not realized that in the case of web applications they are actually interacting with a web browser, and not with an application. They have problems realizing the border between the web application and the web browser. Buttons like “Back” and “Forward” are confused with “Undo” and “Redo” for the application, and sometimes they completely loose their work by stepping out of the web application while following a link (e.g., an advertisement). The last factor, internet delays, was not of major importance in our case because of the high bandwidth connection used during the study and the high availability of the yahoo.com web site. However, in realistic settings, where users use modems for connections and the web server is unreliable, download delays can severely limit the performance.

## 6. Conclusions and Future Work

Desktop application have many drawbacks: they have to be downloaded before their use, have to be developed on multiple platforms, their administration and maintenance is

problematic. In order to overcome these type of problems an increasing number of web applications have been implemented in recent years.

We have presented in this paper a comparison between desktop and web applications, based on an empirical study that investigates the performance of a group of users on two calendaring applications: Yahoo!Calendar and Microsoft Calendar. The study shows that users are twice as slow when using web applications compared with using desktop counterparts, mainly because of the limited interaction mechanisms provided by the web browsers, and the lack of clear delimitation between browsers and web applications. Therefore, any application development should also consider the aspect of usability before deciding on a particular implementation style.

Several ideas seem promising and could be addressed by future work:

- Identifying a method of accurately predicting the performance degradation resulted from implementing an existing desktop application as a web application. Such a method should rely on the mapping of task on the specific interaction mechanisms of each of the paradigms.
- Developing guidelines for web applications based on empirical results, with a focus methods to map existing desktop applications onto web ones.

## References

- [1] J.A. Borges, I. Morales, N.J. Rodriguez, "Guidelines for Designing Usable World Wide Web Pages", *Proceedings of ACM CHI Conference on Human Factors in Computing Systems*, 1996.
- [2] T. Comber, "Building Usable Web Pages: An HCI Perspective", *The First Australian WWW Conference*, 1995.
- [3] P. Greenspun, "Philip and Alex's Guide to Web Publishing", *Morgan Kaufmann Publishers*, 1999.
- [4] M.G. Helander, T. Landauer, P. Prabhu, editors, "Handbook of Human-Computer Interaction (2nd. Edition)", *Amsterdam: Elsevier*, 1997.
- [5] M. Hurst, "Holiday '99 E-Commerce: Bridging the \$6 Billion Customer Experience Gap", *www.goo-dreports.com*
- [6] J. Nielsen, "Designing Web Usability: The Practice of Simplicity", *New Riders Publishing*, 2000.
- [7] J. Nielsen, "Does Internet = Web?", *www.useit.com/alertbox*, 1998.
- [8] J. Nielsen, "Multimedia and Hypertext: The Internet and Beyond (2nd. Edition)", *Morgan Kaufmann Publishers*, 1995.
- [9] J. Nielsen, "The Matters that Really Matter for Hypertext Usability", *Proceedings of the Second ACM Conference on Hypertext*, 1989.
- [10] L. Rosenfeld, P. Morville, "Information Architecture for the World Wide Web", *O'Reilly & Associates*, 1998.
- [11] B. Shneiderman, "Designing the User Interface: Strategies for Effective Human-Computer Interaction (3rd. Edition)", *Addison-Wesley*, 1998.
- [12] H. Shubin, M. M. Meehan, "Navigation in Web Applications", *interactions*, 4(6), 1997.
- [13] H. Shubin, R. Perkins, "Web Navigation: Resolving Conflicts Between the Desktop and the Web Workshops", *Proceedings of ACM CHI Conference on Human Factors in Computing Systems*, 1998.
- [14] R. Tilson, J. Dong, S. Martin, E. Kieke, "A comparison of two current e-commerce sites", *Proceedings on the sixteenth annual international conference on Computer documentation*, 1998.
- [15] "Microsoft Calendar", *www.microsoft.com/office/outlook*
- [16] "ScreenCorder", *www.matchware.net*
- [17] "Yahoo!Calendar", *calendar.yahoo.com*