Using historical data to guide user interface evolution

Stanislaw Osinski
Poznan Supercomputing and Networking Center
ul. Noskowskiego 10, 61-704, Poznan, Poland
e-mail: stanislaw.osinski@man.poznan.pl

ABSTRACT
In this paper we show how historical data, such as existing domain-specific databases or access logs, can be used to guide user interface design. We show how we employed such data in the process of improving usability of a web application used each year by over 40,000 16-year-olds to apply to senior high schools in various Polish cities.

Author Keywords
user interface design, usability, learnability

INTRODUCTION
Usability testing, no matter how formal the actual process is, seems to be the most popular tool for interaction designers to ensure that their UIs work well with real users. Observing the users is the best help in identifying general problems and misconceptions about the interface. However, results obtained from usability testing can not always answer detailed design questions, such as what the right default values should be, or whether the user behaviour patterns observed while small-scale testing accurately predict large-scale production use patterns.

One method of complementing the results of usability testing is observing patterns found in historical data, such as domain-specific databases and access logs of currently running systems. Such data can be particularly rich and useful in case of redesigns of existing interfaces. In such cases, historical data can be used on three levels: 1) during analyses of existing UIs to support usability observations, 2) during prototyping of new UIs to guide decisions that are hard to make based only on usability testing, 3) to verify how the redesigned UIs worked in practice.

In this paper we show how we employed historical data in the process of analysing and solving usability problems of a web application, as well as verifying if the redesign was successful in practice.

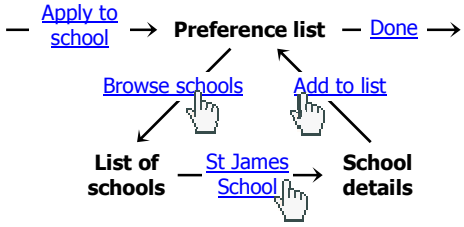

Figure 1. Preference list building by shopping basked model

THE CASE STUDY
Our case study is based on an application called *Nabor* [1], which is used each year by over 40,000 16-year-olds to apply to senior high schools in various Polish cities. The key characteristic of high school admission in Poland is that the process is highly competitive. A number of high schools are always on demand and must every year turn down a great majority of their applicants. Therefore, each candidate is allowed to apply to more than one school, which increases the chances of getting to a school matching his or her expectations. This, however, requires a computer system to coordinate the admission process across all high schools in a city.

The primary task to be carried out by a candidate during the admission process is to fill in his or her high school application form, which, apart from typical personal data, contains a list of schools to which the candidate would like to get admitted. The list is assumed to be ordered by the candidates according to their school choice preferences (the most preferred school at the top), and hence will be referred to as a *preference list*.

While filling in the high school application on-line was not obligatory, it was desirable as it would greatly speed up the application collection process. Unfortunately, the original web-based system for candidates was not usability-tested, which resulted in low numbers of successful electronic applications (lower than 25% as reported by school administration). Therefore, our primary aim was to redesign the interface in such a way as to increase the number of successful on-line applications.

The original UI: not usability-tested
An informal heuristic evaluation of the original design revealed that it was largely document-oriented, which could impair learnability and hence lower the success rate. One possible problem we noticed was related to the way

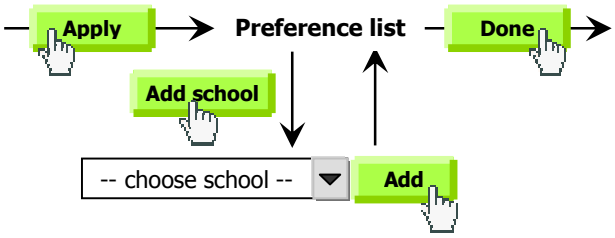

Figure 2. Preference list building by direct school choice

preference list building was designed. The design assumed a model based on the shopping basket metaphor, in which adding a new school to the list required: 1) browsing through the list of all schools in a city, 2) opening a description page of the school to be added, 3) clicking the Add to list link (Figure 1). In theory, this model would not necessarily have to cause problems, but user feedback and usability tests with a number of 16-year-olds confirmed our intuitions. Unfortunately, no access logs from that time were available to further support our observations.

The new UI: make it task-driven

The main challenge to be addressed by the new design was to make the interface learnable enough for a 16-year-old to apply to a school with the lowest possible mental effort. We therefore decided to create a strictly task-driven interface, making filling in the application form the central task.

We also decided to supplement the shopping basket model for preference list building with a simpler drop-down-based direct school choice (Figure 2). Informal usability tests of a prototype with a number of 16-year-olds showed that they favoured the direct school choice.

During the course of detailed design of the new interface we stumbled upon a number of questions, which could be answered with more confidence after analysing data found in historical databases of the *Nabor* system. One problem stemmed from the fact that the application process required that candidates provide up to 6 different addresses: their permanent and temporary address, their mother's and father's permanent and temporary addresses. The previous year's *Nabor* database provided a helpful design hint for this case: out of 5357 candidates, 4591 (85%) had all six addresses equal. This supported our initial intuition that a variant of the responsive disclosure pattern [2] would be most appropriate here (Figure 3).

We also used historical data while designing a printable version of the application form to make the form fit on one A4 page for the majority of candidates. According to the database, the maximum length of the preference list was 26, but for 99% of candidates, the length did not exceed 18. We therefore ensured that a page break would not occur when printing application forms with up to 18 preferences.

To additionally confirm our design-time decisions and gather data for future interface improvement, we ensured

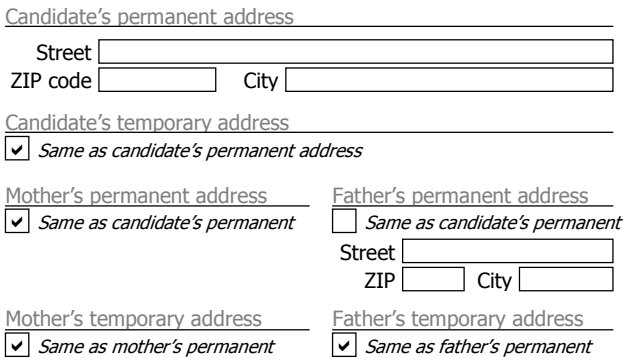

Figure 3. Selective disclosure pattern used for entering multiple addresses

that potentially useful data, such as the percentage of application forms completed on-line, is retained in the system's database. We also built a simple mechanism for counting the number of times particular elements of the user interface (links, form submission) were accessed.

The new UI worked better than the original one

The ultimate validation of the new design was done by over 41,200 16-year-olds applying to high schools in 15 Polish cities in 2006. According to the content of the *Nabor 2006* database, 43% of high school applications were completed on-line, which is a noticeable improvement over the 25% for the original design. Additionally, page access statistics show that less than 20% of the candidates used the shopping basked model (i.e. clicked Add to list) to build their preference lists and over 80% of them used the direct school choice (i.e. accessed the direct school choice page).

CONCLUSIONS

When redesigning user interfaces, historical databases and access logs can be a valuable addition to conventional usability testing. They can strengthen usability testing conclusions and intuitions, but also guide decisions that are difficult to make based solely on user observation. Finally, databases and access logs created during the operation of the new design can give insight into how the new system performs. It is important to bear in mind, however, that historical data may be incomplete and inaccurate (e.g. access logs do not take web page caching into account [3]), and are by no means a substitute for user involvement in the UI design process.

REFERENCES

[1] Pawel Gruszczynski, Bernard Lange et al.: *Building a Large-scale Information System for the Education Sector: A Project Experience.* 7th International Conference on Enterprise Information Systems, Miami, USA, 2005

[2] Jenifer Tidwell, *Designing Interfaces: Patterns for Effective Interaction Design*, O'Reilly, 2005

[3] Karl Groves, *The Misuse of Server Log Files for Usability Analysis*, http://www.karlcore.com/article/id/26