Human-Computer Interaction -- INTERACT'03
M. Rauterberg et al. (Eds.)
Published by IOS Press, (c) IFIP, 2003, pp. 848-851

Online Form Validation: Don't Show Errors Right Away.

Javier Bargas-Avila, Glenn Oberholzer
STIMMT AG, Webernstrasse 5, 8610 Uster, Switzerland
javier.bargas@stimmt.ch, glenn.oberholzer@stimmt.ch

Abstract: Online form validation can be performed in several ways. This reports discusses an empirical study with 77 participants, which has found evidence that the best way of validating online forms is by reloading the form with erroneous fields highlighted. ISO recommended immediate error feedback came in last in this study. Additionally, a modal theory for form completion derived from the results postulates that users are either in Completion or Revision Mode when filling out online forms. These modes affect the users' way of interaction with the system.

Keywords: Online Forms, HTML, JavaScript, Validation Mechanisms, Interaction Design, Interaction Processes

1 Introduction

The use of the World Wide Web as a sales channel places a lot of emphasis on interaction. HTML as the base technology for websites offers only limited interaction mechanisms. New technologies like Macromedia Flash™ or Java™ promise to make up for the shortcomings of regular HTML. Nevertheless, HTML is still the most frequent way to implement interaction – mostly due to its standardization and widespread acceptance. The exchange of data between users and servers happens almost exclusively through forms in applications like registration processes, shopping carts and guest books.

Every human data entry is prone to errors – validation of data is often a must to ensure data quality and consistency. Several methods are used to achieve this. To the authors' knowledge, no research has been conducted discovering the most effective and most user friendly form validation method for HTML forms. The study presented in this paper explores this matter.

2 Theory

2.1 Form Validation

Due to the passive nature of HTML, several helper technologies are used to validate forms on the Net (i.e. Goodman 2001). Validation happens either server-sided or client-sided (Wilde, 1999). If validation is performed on the server, the technology employed is irrelevant for the client. Client side validation is performed mainly using JavaScript. The validation mechanisms used in the study are explained in the following sections.

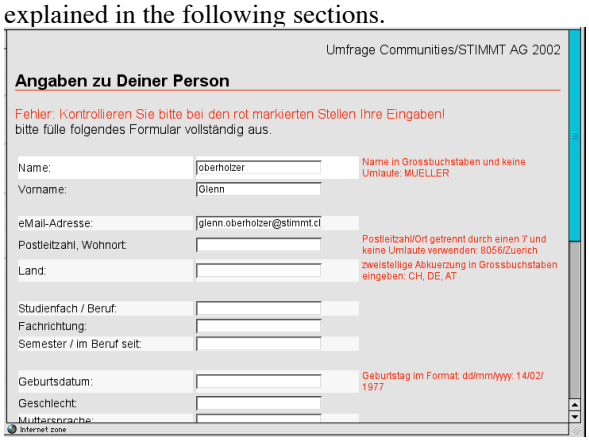

Figure 1: Server-Side Validation – simultaneous display of error messages (SV).

2.2 Server-Side Validation – Simultaneous Display of Error Messages

Validation is performed on the server and a page is returned to the client, which highlights the fields with missing or wrong entries (Figure 1). From an HCI point of view, this method represents an elegant solution; erroneous fields are explicitly highlighted therefore reducing the cognitive load of the user. Errors and the correct syntax do not have to be

memorized and reconstructed. Also the locus of control is directed to the critical parts of the form.

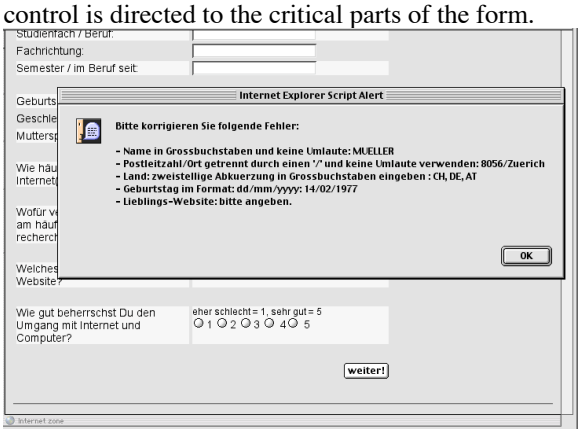

Figure 2: Client-Side Validation using JavaScript Simultaneous Display of Error Messages (JAIO).

2.3 Client-Side Validation – Simultaneous Display of Error Messages.

Validation is performed on the client. The user is presented with one pop-up-window displaying all field names with wrong or missing entries at once. Ideally, the correct syntax is repeated (Figure 2). This puts high cognitive load on the user (memorization). The problems with recall effects are well documented .

2.4 Client-Side Validation using JavaScript – Sequential Display of Error Messages.

Validation takes place on the client. The user is presented only with one incorrect field feedback at a time. Even though the cognitive load is low due to the sequential nature of the validation, the work process is interrupted and attention is drawn from the task at hand. Additional clicks and confirmations have to be performed.

2.5 Client-Side Validation using JavaScript's OnBlur Event

Again, validation takes place on the client. The User is presented with an error message immediately after leaving a field she has filled out incorrectly. The error message appears in a Pop-Up window. Users notice the mistakes immediately after they happen. This enables an instant correction and prevents a tedious revision process later. However, this form of validation is not well known on the World Wide Web. The International Standard Organization implicitly recommends this way of form validation (ISO, 1996). The 9241 standard demands immediate correction of erroneous entries.

3 Experimental Hypotheses

3.1 Precise Formulation of the Problem

The goal of the study is to compare the efficiency of commonly used methods on the WWW to find the most efficient way to implement form validation. Four validation mechanisms are tested:

1) Server Side Validation (SV)
2) JavaScript Validation with simultaneous display of errors (JAIO: JavaScript All in One)
3) JavaScript Validation with sequential display of errors (JOAT: JavaScript One At a Time)
4) JavaScript Validation using the OnBlur Event (JOBE: JavaScript with OnBlur Event).

When looking at efficiency, the following questions are investigated: Which method leads to the least number of mistakes? Which method allows the most time efficient completion of forms?

3.2 Hypotheses

Two hypotheses are derived from the above stated questions (Table 1).

Hypothesis 1	Validation methods SV and JOBE lead to significantly less consecutive errors than JAIO and JOAT.
Hypothesis 2	Validation methods SV and JOBE lead to significantly shorter completion times than JAIO and JOAT.

Table 1: Hypotheses

Reasoning Hypothesis 1

SV presents the user with a comprehensive feedback on the errors made after the user has completed the form. The entire form is displayed again and the erroneous field are visually highlighted and commented. Users can correct all mistakes without further clicks.

JOBE on the other hand provides instant feedback as long as the question to be revised is still in the locus of control. When leaving the field, the system reports errors in a pop-up window. This way, the error can be corrected immediately and the process can continue.

Due to the comprehensive and always visible nature of feedback with SV and immediate feedback with JOBE, the authors assume that these methods will provoke fewer errors than JAIO and JOAT.

Reasoning Hypothesis 2

The reasoning for this hypothesis is derived from Hypothesis 1: A greater number of Consecutive Error leads to additional cognitive processing and interaction processes (clicks and entries). This should

result in longer completion times for JAIO and JOAT compared to SV and JOBE.

4 Methodology

4.1 Experimental Design

An online questionnaire is used for the study. Officially, it is conducted as an „Online-Community Study". Only the last part of the questionnaire actually concerns itself with validation methods. The experiment consists of three parts: Community Questionnaire (37 questions) and Demographic Questionnaire as well as a Subjective Evaluation Questionnaire (11 questions for subjective rating of the form). The Demographic Questionnaire includes 15 straightforward questions (i.e. name, address etc.) with five built in obstacles (see Table 2).

77 subjects are recruited for the experiment. They are randomly placed in one of four groups, each of which is presented with only one validation method:

1) Server Side Validation (SV): n=18
2) JavaScript All-In-One (JAIO): n=17
3) JavaScript One-At-A-Time (JOAT): n=19
4) JavaScript OnBlur-Event (JOBE): n=22

4.2 Obstacles

Key to the study are the obstacles built into the demographic questionnaire:

Field	Obstacle
Name	Had to be entered in capital letters: [FIRST NAME NAME].
Email	Had to include an «@»: [name@place.org].
ZIP/City	Had to be separated by an «/»: [9999/City].
Country	Had to be the two digit country code in capital letters: [CH].
Date of Birth	Day-Month-Year separated by «/» Day and Month two digits, Year four digits: [14/02/1977].
Favorite Website	Had to start with «http://», had to include the domain name and had to be completed with an «/» at the end of the URL: [http://www.place.org/] or [http://place.org/].

Table 2: Obstacles in the demographic questionnaire.

Of course, these validation criteria are not likely to exist on a real web site – hopefully. The unrealistic setting is taken into account, because the goal of the study is to research classic error correction and not the validation of good form design.

4.3 Experimental Procedure

Testing is carried out in a usability lab during a two week period in September 2002. The subjects are first introduced to the setting and given instructions. The error rate is recorded electronically as are all relevant time measures. All demographic data is extracted directly out of the completed forms.

5 Results

A variance analysis of the demographic factors shows no significant differences between the experimental groups' gender distribution (F=8.93, p>0.45), age distribution (F=.971, p>0.41), computer knowledge (F=.352, p>0.78). Effects due to demographic differences can therefore be excluded.

A significant evaluation of the error rate can only be performed when categorizing the errors into two subgroups as follows:

First Time Errors: The term «First Time Error» refers to an error made the first time a field is filled out by the participant. They occur due to the obstacles built into the form. The participant has no chance to prevent them. This rate should not differ among the four experimental groups.

Consecutive Errors: The term «Consecutive Error» refers to an error made after the form has been validated once. This means that a user has been presented with one of the feedback messages and still enters the data incorrectly. The authors assume that this rate differs significantly between the groups.

A variance analysis of the result confirms that the First Time Error rate is not significantly different between the groups (F=2.274, p>0.87).

The key question of the study is whether Consecutive Errors depend on the validation method used (Hypothesis 1). Table 3 states the results.

Method	*M*	*SD*
SV	2	1.88
JOAT	1.58	1.84
JAIO	6.12	3.12
JOBE	5.73	2.64

Table 3: Consecutive Error Rate

A variance analysis shows that there are significant differences in the Consecutive Error rate. (F=18.416, p<0.00). Post-Hoc analysis shows that both SV and JOAT differ significantly from JAIO and JOBE (p<0.00). Neither between SV and JOAT (p>0.59), nor between JAIO and JOBE (p>0.62) exists a significant difference.

These surprising results only partially support Hypothesis 1. As expected, server side validation (SV) leads to only few Consecutive Errors and presenting all errors at once in a JavaScript Pop-up window (JAIO) leads to many Consecutive Errors.

However, only presenting one error at a time when submitting the form led to an even lower error rate than SV (difference however is not significant). This is even more surprising as we assumed that clicking away the pop-up windows will interfere with the revision process. This is not the case.

Most astonishing are the results for the validation method using JavaScript's OnBlur Event (JOBE). Despite the fact that immediate error correction was possible, test participants made a great number of Consecutive Errors. A closer inspection of the taped sessions reveals that many participants were clicking away the error message but did not correct the fields. They continued to fill out the form and were presented with additional error messages when trying to submit the form. Participants were irritated by the fact of being presented with an error message before even submitting the form for validation. It appears that users were not in «Revision Mode» and therefore ignored error messages when completing a form.

Hypothesis 1 can only partially be confirmed. Validation SV and JOAT lead to significantly fewer Consecutive Errors than JAIO and JOBE.

Another important factor when choosing a validation method is the time needed to complete a form (Hypothesis 2). A faster completion time is better economically, because it takes users less time and the probability to successfully terminate the interaction process is higher.

Method	M	SD
SV	119.29	49.85
JOAT	142.71	104.04
JAIO	140.87	47.78
JOBE	134.02	72.25

Table 4: Time needed to send off form (in seconds).

Due to the high variance in time, no significant difference can be observed (F=.533, p>0.59). Also the single comparison between SV and JOAT (the methods that led to fewer mistakes) leads to no significant difference (F=.951, p>0.34). Mean differences can therefore only be regarded as tendencies. There seems to be a trend towards SV as the fastest mechanism.

Hypothesis 2 is rejected for statistical reasons.

6 Discussion

Choosing the right validation method can have a severe impact on the success or failure of a data entry process. The results confirm the very common approach of server sided validation by highlighting all erroneous fields to be user friendly.

6.1 Theory of Mental Modes

The poor results for the OnBlur-Event validation could hint to a modal model for the mental states of users filling out online forms.

At first, users are in «Completion Mode». Their goal is to submit the requested information to the system. The main focus is on filling in all the fields, knowing that there might be mistakes. As soon as users have finished completion, their mental mode switches. With the appearance of the first error message, users enter the «Revision Mode». Now, error messages are acknowledged, interpreted and necessary actions are taken to fix errors.

This modal theory is also supported by behavior in the real world. The natural process of filling out forms is also different from revision by third parties. Habituation might play an important role in this behavior. The advantage of end validation lays in the fact that the user is not constantly interrupted. To prove this theory, further studies are needed.

Also the surprising result that JavaScript Validation with sequential error correction is almost as good as server-side validation can be explained by the modal theory. After completing the form, the attention is explicitly directed to one specific field. This way, a user is guided through the revision process and hence produces fewer mistakes.

7 Further Work

Further studies are currently conducted to verify the modal theory for form validation. Also, the subjective satisfaction with these validation mechanisms must be researched. While subjective data was gathered in this experiment, the results have not shown clear indication towards one or the other method. It would also be interesting to research even different mechanisms .

References

Goodman, D. (2001). *JavaScript Bible* (Gold edition ed.). New York: Hungry Minds.

ISO-9241. (1996-2002). *Ergonomic requirements for office work with visual display terminals* (Parts 1-17), on www.iso.org

Wilde, E. (1999). Wilde's WWW technical foundations of the World Wide Web. Berlin [etc.]: Springer.