A Process Improvement Approach to Improve Web Form Design and Usability

Sean Thompson
La Trobe University
sean@nostin.com

Torab Torabi
La Trobe University
T.Torabi@latrobe.edu.au

Abstract

The research presented in this paper is an examination of how the concepts used in process improvement may be applied to a web form to improve design and usability. Although much research is being conducted in improving the security and usability in how users input information to web sites, the HTML form remains as the primary source of user-web input. Thus far, a "process" oriented approach has not been explored in the literature as a methodology to improve user input interfaces on the web. Our approach is focussed on capturing as much user data as possible and using the process improvement engine as a tool to extract knowledge from the data so that the web form can be improved in terms of usability, clarity, security and user assistance.

1. Introduction

HTML web forms are the primary medium for user input on the web [19] and are used for a variety of reasons, including registration, e-commerce sales and security purposes. Sometimes, the process of filling these web forms can be complex. The early web applications focussed more on simpler tasks such as searches and browsing large volumes of data, however nowadays people use the web to perform complex inter and intra-business processes [11]. Although developers endeavour to ensure processes are as simple to use and as easy to understand as possible, even the most basic of forms will be filled out incorrectly by end users. This can be critically important because user failure to understand or fill out a form correctly may result in loss of site traffic and/or revenue. Camenisch et al [19] argues that data submitted in HTML forms is "error prone and fraudulent" and also suffers from usability problems. This research presents a methodology in which mining the data submitted by users can be used by a process improvement engine [12] to improve the design and usability of a web form. The data gathered for mining in this approach can be used to provide three possible benefits:

1. Improvement of clarity. When the form data is recorded, we may begin to see patterns emerging such as consistent errors by a significant number of users on one part of the form. In this case, we can scrutinize this section of the form and try to improve the design and usability which should hopefully reduce the percentage of error messages/failed submissions for the form. This in turn will also greatly reduce user frustration experienced when using the site.

2. User assistance. If a user is having some unique trouble with a part of the form, feedback can be sent to the user to help the user understand fully what they are doing wrong and why their submissions are failing.

3. Security flaws. If data is being submitted through the form which has extreme values, it could indicate a security flaw. Sometimes bots and search crawlers can submit data to a form causing consequences the developer did not intend. Having the data mined in this way can provide an indication of which submitted data is not legitimate so that it may be weeded out and the security flaw addressed.

A multitude of work has been conducted already in the broad area of *web mining* including finding relevant information, discovering new knowledge from existing information, personalization and learning about consumers or users [5]. Web mining, as data mining endeavours related to the web can be divided into three classes: content mining, usage mining and structure mining [1, 5]. According to this classification, the bulk of this approach would fit into *usage mining* as we are interested only in the data submitted by users via the form. Therefore, since this is the only data we are concerned with and it is captured by the web server upon submission, the issues and challenges facing other aspects of web mining such as the *abundance problem,* separating "noisy" data [7] and other issues like the ones covered in [6] are avoided in this case.

This approach is also related to the work presented in the area of *personalization,* a concept where as much historical user data is stored as possible, and this data can be used to "personalize" or adapt the content and presentation of the website for the user [3, 9, 10]. This concept was first described as "sites that semi-automatically improve their organization and presentation by learning from visitor access patterns" [4]. Essentially, data is gathered from user input/navigation and stored so that the information may be used to improve the experience for that user. This relates because our intention is similar, only on a mass scale. We seek to gather information about problems and errors users experience when enacting a web process and to then improve the nature of the process for future users, so similar problems are not faced.

Techniques that fall into the category of usage mining such as discovering browser navigation patterns are also relevant. The purpose of discovering users browsing behaviour is so decisions regarding modifying or restructuring could be tailored to better suit the user [3]. Again, we are concerned with discovering common behaviour which indicates a problem with clarity or presentation in the form. As with all types of data mining however, certain properties should be adhered to, in order to gain a good result [8]. This includes a large amount of "clean" data. Small amounts could produce erroneous feedback and all "noisy" or flawed data should be removed.

Another related approach which is not incidentally related to data mining is the system presented in [2] which describes a dynamic environment in which a piece of software is embedded into a web form which can dynamically supply the user with helpful information such as validations checks and help messages. The assistance functions are defined by the developer as a set of assistance rules which are compiled into a program and embedded into the web form. This approach is useful for developers to easily customize user assistance; however it appears if there is any problem inherent in the form that this system is not capable of picking it up.

The research presented in this paper begins after this introduction with a description of some of the concepts taken from the process improvement model and a description of how these concepts may be applied to benefit web forms. This description is presented with our approach in section 2. Section 3 outlines a case study which takes a sample web form and applies our approach to it along with an evaluation of results. Section 4 presents a summary and a conclusion.

2. Approach

In this body of research we consider the act of users inputting information to a web site to be a type of "process". For the purposes of this research and its associated process improvement model we consider a "process" as a sequence of "activities". These activities essentially modularize the process into a set of related smaller and simpler tasks which *actors* perform [13]. In the context of this research, an "activity" would be a single web form displayed on a single web page. The act of a user going to the relevant URL, filling the HTML form and submitting it constitutes an activity in this sense. A process, in this context, could encompass multiple forms sequenced to achieve the same outcome where each individual form is an activity, or if only one form is required, this one activity/form could represent the process also as a whole.

The concepts involved in the model aimed at deviation detection in enacting processes we presented in [12] is the test system used in this approach. The model is fundamentally a three tiered approach which begins with an interface where constraints and boundaries may be defined for different process activities. These include several types of *inconsistencies* which are distinguished from *deviation* in [14] as being a concept regarding the status of states, where a *deviation* is a concept relating to transition between states. Once we have our "activities" or web forms we can define our own set of inconsistency types to test for and also set our own boundary values to reference recorded actual values against. Some inconsistencies we could check for in this case are:

- User Time Violation. If the user takes too little time to fill the form out, it could be a bot or a malicious user not interested in filling the form properly. If they take too much time, this could indicate a problem the user has understanding the form.

- Total Time Violation. The longer it takes a user to complete their goal on the computer, the more frustration they experience [15]. If a significant number of users are taking a large amount of time to fill the form, this could indicate the form is either unclear or too long.

- Consistent Error Message. If a large number of users seem to be receiving the same error, this indicates a problem with the corresponding part of the form which will require examination.

- Excessive User Submissions. If the user attempts to submit the form too many times unsuccessfully, this may cause frustration.

- Excessive Total Failed Submissions. If many users are submitting the form unsuccessfully, it could indicate ambiguity or a software malfunction problem.

Boundary values can be set for numerical data using concepts involved in Statistical Process Control (SPC). The method here is to compute the mean value and the standard deviation for a given data set. A 3**σ** (where **σ** is the standard deviation) range is then applied on either side of the mean value. This has proven to be a suitable range for picking up "out of control" values as well as triggering very few false alarms [16, 17]. SPC, a successful tool in quality control for production lines and manufacturing has now expanded into electronics and software engineering [20]. However, as will be shown in the test data in section 3, SPC requires a large data set in order to compute reliable boundaries [18]. Exactly how much data constitutes enough to adequately apply SPC will be situation specific and sometimes difficult to estimate. Therefore, the use of SPC is only tenable when an ongoing and large volume of usage is expected for a particular form.

The second tier of the approach relates to the monitoring and recording of the data as the user inputs it into the form. This is no problem at all, since the web server handles all the submitted data which can be then recorded in a suitable fashion so that it can be mined. Other data we may require such as the users IP address or browser type and version etc… are easily attainable using a server side language without the users explicit input. We structure the recorded data along with the referential data in the model so that the values in each can be easily compared with each other. For example, a relational database structure is illustrated in figure 1 showing how recorded timestamps may be referenced to our defined minimum and maximum times to find out of bounds time values:

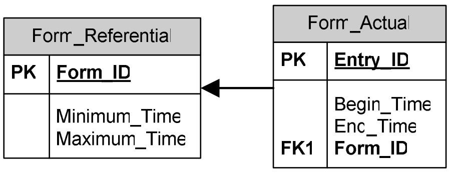
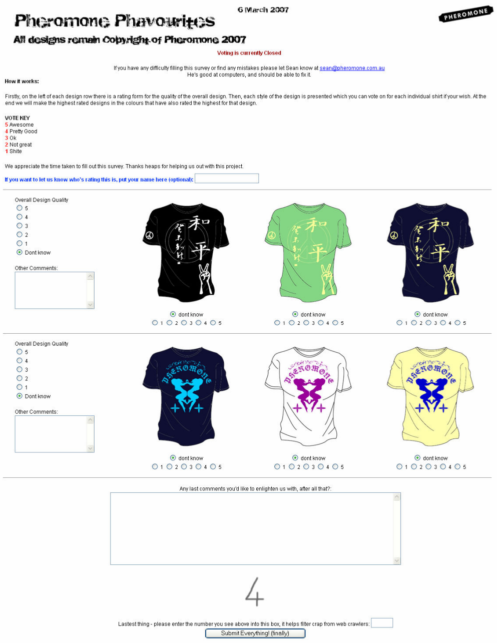


Figure 1 – Referencing Data

The third tier is simply a matter of comparing actual values to referential ones and mining for patterns in accordance with the inconsistencies we have defined in the first tier. If for example, we find actual time entries which exceed the Maximum_Time set in the referential definition, it is up to the developers to judge the best course of action. It is reasonable to expect a relative few entries will be out of bounds with the time range, however if there is a significant number outside this range then it could indicate problems with the clarity of the instructions or some other issue requiring attention.

3. Case Study

To better illustrate the methodology involved and to provide an evaluation of the results attained using this approach, a case study has been performed on a simple web form process. A condensed portion of the form is illustrated in figure 2:



Pheromone Phavourites

6 March 2007

PHEROMONE

All designs remain Copyright of Pheromone 2007

Voting is currently Closed

If you have any difficulty filling this survey or find any mistakes please let Sean know at sean@pheromone.com.au
He's good at computers, and should be able to fix it.

How it works:

Firstly, on the left of each design row there is a rating form for the quality of the overall design. Then, each style of the design is presented which you can vote on for each individual shirt if your wish. At the end we will make the highest rated designs in the colours that have also rated the highest for that design.

VOTE KEY
5 Awesome
4 Pretty Good
3 Ok
2 Not great
1 Shite

We appreciate the time taken to fill out this survey. Thanks heaps for helping us out with this project.

If you want to let us know who's rating this is, put your name here (optional):

Overall Design Quality
○ 5
○ 4
○ 3
○ 2
○ 1
● Dont know

Other Comments:

● dont know
○ 1 ○ 2 ○ 3 ○ 4 ○ 5

● dont know
○ 1 ○ 2 ○ 3 ○ 4 ○ 5

● dont know
○ 1 ○ 2 ○ 3 ○ 4 ○ 5

Overall Design Quality
○ 5
○ 4
○ 3
○ 2
○ 1
● Dont know

Other Comments:

● dont know
○ 1 ○ 2 ○ 3 ○ 4 ○ 5

● dont know
○ 1 ○ 2 ○ 3 ○ 4 ○ 5

● dont know
○ 1 ○ 2 ○ 3 ○ 4 ○ 5

Any last comments you'd like to enlighten us with, after all that?:

4

Lastest thing - please enter the number you see above into this box, it helps filter crap from web crawlers:

Submit Everything! (finally)

Figure 2 – T-shirt rating form

The test case is a company called PHEROMONE™ who design, manufacture and distribute men's t-shirts. The form is a collection of their latest t-shirt designs with controls to enable the user to rate certain designs, provide comments, enter their name (optional) and a security number box check. To better test this approach, no client side validation scripting (such as JavaScript) was used in this form.

A process and corresponding activity was set up in the first tier of the process improvement engine to handle the data from this web form. Also, the five inconsistency types mentioned in section 2 were defined for the web form activity. Since there was no pre-existing data available from the form, a couple of cursory run-throughs were conducted by us to attain some appropriate but modest boundary values for the activity, thus setting up the reference model for which to compare the actual values from this form.

The data entered by each user was captured in the PHP server side code and inserted into the database along with other information about the user such as their IP address. Also captured was the timestamp they opened the URL and the timestamp when the form was submitted. Error messages are also captured. Once the data was collected, we ran our engine over it. A rundown of the data compiled from this survey follows:

Total Unique Valid Submissions: 41
Total Valid Submissions: 53
Total Submission Failures: 8

4 IP Addresses submitted the form 2 times
4 IP Addresses submitted the form 3 times

Average Time: 199.66 seconds
Quickest Time: 18 seconds
Longest Time: 420 seconds

Standard deviation of time taken: 70.199 seconds

The first thing to note is that the only error message and therefore the only submission failure possible on this particular form is the users failure to enter the correct security code into the input box at the bottom of the screen. This security check is a feature used by many websites across the internet to validate user form submissions nowadays and was extensively tested in development and works properly as intended. According to the data gathered, there were 8 submission failures (error messages) of this type from 61 total submissions (53 successful submissions). This equates to just over 13% of submissions who are misreading the security check at the bottom. There are a number of different remedies we can try to lower this number, such as changing the code to a more prominent colour like red, increasing the size of the font underneath, or adding more characters to the code.

As mentioned in section 2, there is a problem with the SPC boundaries on the form time constraints, indicating that the 53 (valid) submissions were not enough data for SPC to be adequately applied. Given the standard deviation for the time data set was 70.199 seconds and the mean time was 199.66 seconds, we get an inapplicable -10.937 for the lower boundary and 410.257 for the upper control boundary. This means that the only out of control value in this dataset was the 420 second form which excludes an 18 second and another 21 second submission time. There were 69 different t-shirts to rate in this form, it is reasonable to assume that the users could not have loaded the page, accurately considered the form and submitted the data in such a short amount of time. Therefore, in this instance, the boundaries we initially applied from our cursory run through of the form are better.

In any case, the engine has instantly identified two cases where the votes should be disregarded – the 18 second and 21 second submissions. This along with the issues regarding the security code show that even for a simple form such as this, the engine was successful in providing some useful information on improving the form and discarding spurious data.

4. Conclusion

In this paper, we have presented a different type of methodology on how a process improvement engine may assist in improving usability on a web interface. Our approach along with some background in the area was presented along with a case study which was tested using the model.

The approach succeeded in providing some useful feedback from the data gathered. The two main benefits in the example cast study was the improvement of the security code input at the end of the form and the removal of some erroneous data.

In any case, the effort and cost involved in implementing an engine such as this must first be weighted against the possible benefits improvement of the form could yield. It is therefore advisable that an approach like this is more suited to critical aspects of input such as e-commerce forms and user registrations.

References

[1] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, Pang-Ning Tan, “Web usage mining: discovery and applications of usage patterns from Web data”, ACM SIGKDD Explorations Newsletter, Volume 1 Issue 2, ACM Press, January 2000.

[2] Aoki, Y.; Shinozaki, M.; Nakajima, A; “Interactive Web forms based on assistance rules”, IEEE International Conference on Systems, Man and Cybernetics, Volume 7, 6-9 Oct. 2002 Page(s):8 pp. vol.7.

[3] M. Eirinaki, M. Vazirgiannis, I. Varlamis, “SEWeP: using site semantics and a taxonomy to enhance the Web personalization process”, Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining KDD '03, ACM Press, August 2003.

[4] M. Perkowitz, O. Etzioni, “Adaptive Web Sites: An AI Challenge”, in Proc. of the Fifteenth International Joint Conference on Artificial Intelligence, Nagoya, Japan, 1997.

[5] Raymond Kosala, Hendrik Blockeel, “Web mining research: a survey”, ACM SIGKDD Explorations Newsletter, Volume 2 Issue 1, ACM Press, June 2000.

[6] Minos N. Garofalakis, Rajeev Rastogi, S. Seshadri, Kyuseok Shim; “Data mining and the Web: past, present and future”, Proceedings of the 2nd international workshop on Web information and data management WIDM '99, ACM Press, November 1999.

[7] Lan Yi, Bing Liu, Xiaoli Li; “Eliminating noisy information in Web pages for data mining”, Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining KDD '03, ACM Press, August 2003.

[8] Ron Kohavi, “Mining e-commerce data: the good, the bad, and the ugly”, Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining KDD '01, ACM Press, August 2001.

[9] Anindya Datta, Kaushik Dutta, Debra VanderMeer, Krithi Ramamritham, Shamkant B. Navathe, “An architecture to support scalable online personalization on the Web”, The VLDB Journal — The International Journal on Very Large Data Bases, Volume 10 Issue 1, Springer-Verlag New York, Inc., August 2001.

[10] Magdalini Eirinaki, Michalis Vazirgiannis, “Web mining for web personalization”, ACM Transactions on Internet Technology (TOIT), Volume 3 Issue 1, ACM Press, February 2003.

[11] Marco Brambilla, Stefano Ceri, Piero Fraternali, Ioana Manolescu, “Process modeling in Web applications”, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 15 Issue 4, ACM Press, October 2006.

[12] Thompson, S; Torabi, T, Joshi, P; “A Framework to Detect Deviations During Process Enactment”, 6th IEEE International Conference on Computer and Information Science, IEEE Computer Society Press, Melbourne, Australia, July 2007.

[13] Mark Dowson, Brian Nejmeh, William Riddle; “Concepts for Process Definition and Support”, Proceedings of the 6^{th} International Software Process Workshop, IEEE Computer Society Press, October 28-31 1990, Hakodate, Japan.

[14] Gianpaolo Cugola, Elisabetta Di Nitto, Alfonso Fuggetta, Carlo Ghezzi; “A framework for formalizing inconsistencies and deviations in human-centered systems”, ACM Transactions on Software Engineering and Methodology (TOSEM), Volume 5 Issue 3, ACM Press, July 1996.

[15] Valerie Mendoza, David G. Novick, “Usability over time”, Proceedings of the 23rd annual international conference on Design of communication: documenting & designing for pervasive information SIGDOC '05, ACM Press, September 2005.

[16] W.A. Florac and A.D. Carleton, “Measuring the Software Process: Statistical Process Control for Process Improvement”, Addison-Wesley, 1999.

[17] Jalote, P.; Saxena, A.; “Optimum control limits for employing statistical process control in software process”, IEEE Transactions on Software Engineering, Volume 28, Issue 12, Dec. 2002 Page(s):1126 – 1134.

[18] Qing Wang, Nan Jiang, Lang Gou, Xia Liu, Mingshu Li, Yongji Wang, “BSR: a statistic-based approach for establishing and refining software process performance baseline”, Proceeding of the 28th international conference on Software engineering ICSE '06, ACM Press, May 2006.

[19] Jan Camenisch, abhi shelat, Dieter Sommer, Roger Zimmermann, “Applications and system issues: Securing user inputs for the web”, Proceedings of the second ACM workshop on Digital identity management DIM '06, ACM Press, November 2006.

[20] João W. Cangussu, Raymond A. DeCarlo, Aditya P. Mathur, “Monitoring the software test process using statistical process control: a logarithmic approach”, ACM SIGSOFT Software Engineering Notes , Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering ESEC/FSE-11, Volume 28 Issue 5, ACM Press, September 2003.