WEB FORM PAGE IN MOBILE DEVICES
Optimization of Layout with a Simple Genetic Algorithm

Luigi Troiano
University of Sannio, Viale Traiano, 82100 Benevento, Italy

Cosimo Birtolo, Roberto Armenise, Gennaro Cirillo
Poste Italiane S.p.A., Tecnologie dell'Informazione
Sviluppo Sistemi Informativi - Centro Ricerca e Sviluppo
Piazza Matteotti 3, 80133 Naples, Italy

Keywords: Optimization of layout, Web Form, User Interface for Mobile Device, Genetic Algorithm, Prediction, Search Based Software Engineering.

Abstract: Filling out a form on mobile devices is generally harder than on other terminals, due to the reduced keyboard and display size, entailing a higher fatigue and limiting the user experience. A solution to this problem can be based on reducing the input effort required to the user by auto-completion, and re-organizing the fields in order to provide first those with a higher prediction power. In this paper we assume to be able to predict the user input and we optimize the fields layout aiming at reducing on average the input actions.

1 INTRODUCTION

One of the main limitations in deploying a user interface or a Web application to mobile devices is the reduced size of their screen displays and keyboards. The display size represents, as a matter of fact, one of the main constraints in designing and organizing interface elements for mobile devices. Selecting the proper number and typology of elements, how to arrange them and splitting them in two or more pages are issues a designer has imperatively to face, mainly when the client device has a display with reduced capabilities. In order to ensure an appropriate level of user-experience, particular attention must be paid to designing those elements that gather information or to encouraging the user interaction. As the keyboard is the main mean the user interacts by, the reduced number and size of keys can affect also the user experience, as entering data becomes a tiring task. On the other side, forms are among the most interactive elements of any Web application. They collect user information by typing into text boxes, selecting an option by radio buttons and check boxes, etc., thus compelling a higher degree of user participation than simply clicking hyper-links. The greater is the level of user interaction, the more critical is usability. Indeed, filling a form on mobile devices can be so tiring that the task becomes frustrating to the user, especially when the application requires several forms to be filled. From the user's perspective, filling a form is a cost to pay before to gain access to the service. But, at the same time, forms represents a key feature to make on line applications really useful. On mobile devices a form usually appears as a list of labeled input fields, generally arranged according to a subjective priority, given by the designer. However, the input effort can be reduced by predicting the user data entry and auto-completion can drastically change the order that is suggested in order to maximize the input predictability. Assumed we have a model for predicting the user input, this work focuses on presenting a methodology aimed at ensuring an easy-to-use form layout for user forms, especially in the case of interaction through small mobile displays.In our work, the form layout optimization aims at balancing screen space requirements of fields with an easy accessibility and usability of the form. As will be explained in details in Section 3, starting from the screen size, the semantic relations between fields and past user interactions, a genetic algorithm attempts to optimize the fields distribution on the screen. In this way we can provide a form layout able to considerably reduce the *time to completion* that is the time to fill out a form. In Section 4 we describe the characteristics of the genetic algorithm used in our experimentation; Section 5 is aimed at presenting some experimental results

Troiano L., Birtolo C., Armenise R. and Cirillo G. (2009).
WEB FORM PAGE IN MOBILE DEVICES - Optimization of Layout with a Simple Genetic Algorithm.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Human-Computer Interaction*, pages 118-123
DOI: 10.5220/0002007001180123

and finally Section 6 outlines conclusions and future work.

2 RELATED WORKS

The problem of designing a usable form as been recently addressed by literature. As highlighted by Wroblewski (Wroblewski, 2008), "a well–designed form is barely noticeable. But does not mean the design process is". Indeed, a form layout can seem obvious although its design requires a significant effort. This is common to many other user interface design problems. Eye tracking can be used to assess different alternatives and the resulting heat maps and fixations are analyzed in order to better understand which factors affect usability. For instance, an excessive distance between labels and the related input fields requires more time to users for completing the form. Instead, labels on the top of fields allow the user to capture both labels and input fields with a single eye movement, entailing a lower completion time. Furthermore if you consider a mobile device, usability becomes a key element. A usable interface brings many benefits: users are able and willing to use the various features and services supplied by the operators, the need for customer support decreases, and above all, user satisfaction improves (Jokela et al., 2006). Given the typical input limitations of a mobile device, Mobile Web Best Practices (W3C, 2008) suggest to reduce as far as possible the user input. Moreover they suggest to limit scrolling to one direction and to split pages into more but usable parts. According to Nielsen (Nielsen, 1994) usability is associated to five attributes: learnability, efficiency, memorability, error and satisfaction. These attributes must be prioritized according to user and task analysis, and then operationalized and expressed in a measurable way. Lentz and de Jong (Lentz and de Jong, 2005) analyze the form submission logs to obtain a well-structured layout optimized to prevent user errors in filling a form. They identify two factors as crucial for usability: the sequence of elements in the form and the spatial distribution of fields. Ahmad et al.(Ahmad et al., 2003) propose the utilization of fuzzy logic to obtain an automatic fitness evaluation method that can overcome the trade-off between requirements, often conflictive, and it is aimed at making a usable web page layout, in order to meet user preferences and to conform to standard guidelines at same time. In a previous work (Troiano et al., 2008) we have investigated the application of genetic algorithms as a viable approach to optimize a menu layout in order to maximize accessibility and compliance to guidelines and user preferences. A related study led by Google (Baluja, 2006) analyzes the problem of browsing web pages on small screens. In particular they adopt a Machine Learning Framework for segmenting a target front page. They segments the page into 9 equally sized regions, each able to be enlarged using the phone keypad.

3 OPTIMIZING THE FORM LAYOUT

The design of usable forms did not seem to receive enough attention in the literature, although designing a usable form is a challenging task (Lentz and de Jong, 2005). Indeed a non-usable form, can lead the user to make errors and to gain frustration, resulting in delivery failures and eventually loss of users and customers. The issue of creating usable forms becomes more critical in mobile devices, as the reduced screen and keypad size poses severe limitations and constraints to the user experience. In our work, the form layout optimization aims at finding a trade-off between fields occupation and position with form accessibility and usability. In particular we focus attention on mobile devices, and we assume that a form is made of a set of input controls (i.e. text fields, radio options, combo boxes, etc.) stacked vertically. Each control is labeled on the top (Wroblewski, 2008). Each field has a fixed width in order to occupy the whole display width; its height depends instead on the font size and on the maximum number of allowed characters. When we provide a form to fill out on a small screen device the question that arises is how to split the fields across different pages, which fields to place on each page and in which order. Answering this question, we should consider that a larger number of pages is limiting usability as well: the number of pages should be kept small. Another direction to improve usability consists in providing auto-completion features so that the input in some fields can predict the values of other fields. Fields with higher prediction capability should come first because the user generally fills out the field from the top to the bottom. Looking for a solution able to trade-off these different requirements can be regarded as an optimization problem, in particular this problem is combinatorial in nature as different placements of fields leads to different form layouts (e.g. re-arrangement of fields and page splitting). Indeed re-arranging items (e.g. fields) to fit into bins (e.g. pages) of fixed size is recognized as "*bin packing*" problem that is a combinatorial NP-hard problem (Dyckhoff, 1990). This suggests that a Genetic Algorithm (GA) can effectively overcome our problem. Therefore, starting from the screen size and the space required by each field in the form, the GA handles the fields occupation and determines if it is necessary to organize them in more pages. By this way, GA ensures that the screen space is not wasted and that there is enough room for the input text at the same time. Moreover, the distribution of fields takes into the account the logic grouping of fields and other designer preferences regarding the maximum and minimum number of fields in each page, the order of fields and where they should be positioned. We consider also the predictive power of each field of the form, as those with a higher prediction power should come first as filling them will make possible to pre-fill the following fields. For this purpose we adopted a Bayesian model aimed at predicting the values of form fields given the values on some of them. The model is trained according to a log of past submissions. Starting from these logs, we have built a Bayesian Network (BN) to map statistical dependencies between the data of past user interactions. Moreover we have assumed that each field of form correspond to a node of BN (i.e. discrete random variable) and so the different values filled in the same field become possible states of node. When user fills out data in some fields, which become "evidences", the system enforces Bayesian inference in order to calculate the posterior probability distributions of all the other network nodes. Therefore, for each field will be chosen as "prediction" the state having a posterior probability exceeding a certain threshold (0.7). In order to characterize the performance of a multi-words completion (Nandi and Jagadish, 2007) we take into account two quantities: *precision* and *recall*, where the precision p is the number of the completions accepted by the user over the number of all predicted completions, while the recall r is the number of completions over the number of all the possible predictions that is all the fields without user's evidences. Given the ability of predicting field values, the genetic algorithm attempts to arrange the fields so that they are able to predict a larger number of values, placing on top most predictive ones in order to reduce the user input and thereby to improve usability.

4 ADDRESSING THE PROBLEM BY A GENETIC ALGORITHM

This paper proposes the exploiting of a GA in order to facilitate the user when filling the form by properly re-arranging fields and predicting user-input. The algorithm proposed is inspired to the Simple GA given by Goldberg (Goldberg, 1989). The structure is outlined in Fig.1.

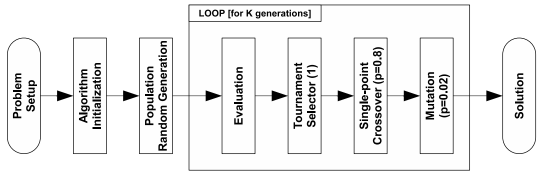

Figure 1: Algorithm structure.

After the problem is setup in terms of fields, layout preferences and display requirements, the algorithm is instanced and the initial population is randomly generated. The algorithm body is made of *evaluation stage* where a fitness score is assigned to each population individual and *genetic processing* where individuals are genetically processed by selection, crossover and mutation. After K generations the best individual is obtained that is the individual whose phenotype guarantee an optimized layout of form fields on different pages according to the specified mobile device constraints.

4.1 Chromosome

In our work we use a chromosome made by N genes. Each genes represents a specified field that user has to fill. Therefore the chromosome's phenotype is a set of N fields (*Field 1*, *Field 2*, ... , *Field N*) that compose an e-service such as the money order. In this example the phenotype is made by: *Sender Name*, *Receiver Name*, *Address*, *ZIP code*, *City*, *Amount*, *Number of checking account* and *Description*. Each gene is encoded by an integer $a_i \in [0,(N*Q-1)]$ with $i = 1...N$, where N is the number of fields and Q is the number of available positions in a page. A gene codes the number of page (p) of the specified field and the position in the page (q) as depicted in Fig.2 and shown in Eq.1.

$$q_i = a_i \bmod Q; \quad p_i = a_i \operatorname{div} Q \qquad (1)$$

where p_i is the page number the *Field i* belong to, q_i is its position within the page p_i, `mod` is the module operation, and `div` is the integer division operation.

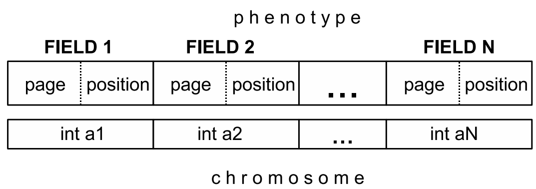

Figure 2: Chromosome

We could introduce illegal individual in order to avoid having two fields in the same position that is $q_i = q_j$ ($\forall i, j \in \mathbb{N}$), but it would cause an increased complexity of the problem which has to consider a lot of illegal individuals among the population. Therefore we does not introduce illegality and we solve this problem assuming that if two fields have the same position q we dispose them according the lexicographical order given by the chromosome. For instance if *Field1* and *Field2* (see Fig.2) have the same value of page ($p_1 = p_2$) and position ($q_1 = q_2$) the algorithm will arrange *Field1* before *Field2*.

4.2 Fitness

The fitness function for an individual x is aimed at modeling the possible disposition of fields in several pages, taking into account different requirements such as predictability of single field for the other ones, semantic information of the field, area, and preference settings compliance. Thus it is defined as convex combination.

$$fitness(x) = \sigma \cdot P(x) + (1 - \sigma) \cdot C(x) \tag{2}$$

where $\sigma \in [0, 1]$, $P(x)$ is the degree of predictability, and $C(x)$ is the degree of constraints' compliance. In particular, $P(x)$ is defined as a measure of prediction power of a specified disposition.

$$P(x) = \begin{cases} P_1(x) & \text{Case 1} \\ \frac{P_1(x)+P_2(x)}{2} & \text{Case 2} \end{cases} \tag{3}$$

where in *Case 1* the predictability occurs within the page forcing to dispose in the first position of each page the field with the highest prediction power in the page (see Section 5.1), while in *Case 2* the predictability occurs within the page and across the pages forcing the arrangement of the fields with a higher prediction power in the first page (see Section 5.2). Instead constraints' compliance is defined as the weighted mean:

$$C(x) = \sum_{j=1}^{m} w_j \cdot c_j(x) \Bigg/ \sum_{j=1}^{m} w_j \tag{4}$$

where m is the number of preferences, w_j is the constraint importance, and $c_j(x)$ is the compliance of x to a specific preference c_j. Therefore, we assumed a compensation among optimization criteria. Each preference has a priority $w_j \in [1, 5]$, where 1 is the lowest priority (i.e. not very important), 5 the highest (i.e. very important). The default *priority* value is 1. Preferences can be of different kind, in our work we considered the following types:

- *Number of Pages* (*min, max*) defines the *min* and *max* number of pages;
- *Semantic requirement* (*field1*, *field2*) defines if a specific field *(field1)* must be coupled with another field *(field2)* in the same page.
- *Preferred Position in the Page* (*field, atBeginning*): if *atBeginning* is equal to "*true*", *field* should be in the first position, otherwise if *atBeginning* is equal to "*false*", *field* should be in the last position.
- *Preferred Page* (*field, atBeginning*): *field* should be in the first/last page of form (*atBeginning* = "*true/false*").
- *Dimension requirements* (*dimensionsOfFields, priority*) defines the height of each field.

5 EXAMPLES OF APPLICATION

In order to evaluate the behavior of our algorithm we consider money order service of Poste Italiane. User has to fill out 8 different fields in order to submit the money order and these 8 fields are the phenotype of chromosome (See Fig.2).

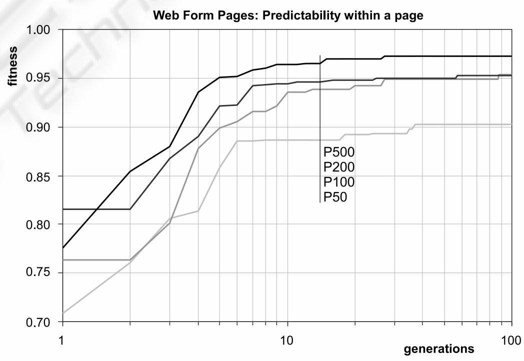

Figure 3: Fitness trend varying population size

We arrange the disposition of these fields in different pages according to the different constraints such as: the number of pages has to be between 2 and 4; some fields are coupled, i.e. *Number of Checking account* with *Receiver Name* and *City* with *ZIP code*; the arrangement has to minimize vertical scrolling in a page, avoiding the horizontal scrolling and minimizing the blank areas; some fields have to occupy a specific position, i.e. *Description* on the latter pages, preferably in the last one, *Amount* on the bottom of a page and *Sender Name* on the top. In addition we wish to minimize the number of fields to fill out and to improve usability. According to Eq.3 we consider 2 different cases, each characterized by a different prediction employment, although always aimed at reducing the user input typing: (i) predictability within a page; (ii) predictability within and across pages.

5.1 Predictability Within a Page

In this case study we consider the prediction contribution (see Eq.2) as the prediction power provided by the first control of each page for the remaining controls within the page. The prediction component is expressed in Eq.5 as the sum of the root mean square of precision p and recall r expressed in Section 3.

$$P_1(x) = \frac{1}{G} \cdot \sum_{i=1}^{G} \sqrt{\frac{p_i^2 + r_i^2}{2}} \quad (5)$$

where G is the number of pages of individual x, p_i and r_i are respectively the precision and the recall evaluated when we fill out only the field which holds the first position in the page i and we use as testing set the other fields within the page i. Therefore the fields on the top of each page are expected to be the most predictive within the page. In other words, the disposition of fields will be in according with an decreasing prediction power from the top to the bottom of pages. This will allow users to fill out the first fields obtaining a good prediction of values in the remaining fields within the page. Having the most predictive fields in the top of page should lead to reduce the user-inputs as users usually begins to fill the first part of the page especially on mobile devices. In our experimentation, we executed 5 runs for different problem configurations. The average behavior of the algorithm is depicted in Fig.3, where the fitness behavior is studied with 100 generations, varying the population size (i.e. 50,100,200 and 500 individuals). Elitism is 5. We note that generally the population size does not play a relevant role in the algorithm convergence, and also with small populations it is possible to obtain good results. The convergence behavior is confirmed by a test made without elitism.

5.2 Predictability Within and Across Pages

In this case study we consider a combination of the two prediction schemes. We consider as prediction factor ($P(x)$ in Eq.2) the arithmetic mean of $P_1(x)$ and $P_2(x)$ as seen in Eq.3.

$P_1(x)$ is expressed in Eq.5 as the prediction power provided by the first control of each page for the remaining controls within the page, while $P_2(x)$ is the prediction between filled pages and the following ones (See Eq.6) and is defined as the sum of the root mean square of precision p and recall r.

$$P_2(x) = \frac{1}{G-1} \cdot \sum_{i=1}^{G-1} \sqrt{\frac{p^2(e_i,t_i) + r^2(e_i,t_i)}{2}} \quad (6)$$

where e_i is the set of evidences given by the fields of the pages $\{1,2,...,i\}$, t_i are the other fields within the pages $\{i+1,...G\}$, G is the number of pages of individual x, and $p(e_i,t_i)$ is the precision evaluated considering as evidence set e_i and as testing set t_i, at the same time $r(e_i,t_i)$ is the recall with those evidences and those test samples.

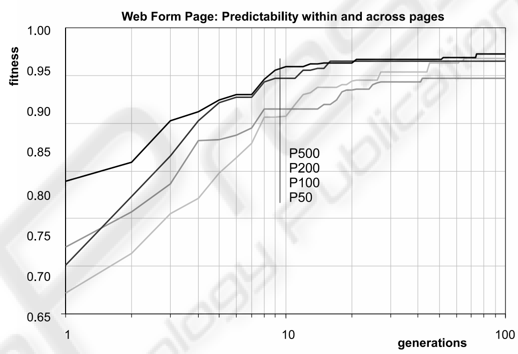

Figure 4: Fitness behavior after 100 generations varying population size.

In this case study we wish to obtain a form layout in which top fields of each page are the most predictive ones within the page, and at the same time, the whole fields of first pages are more predictive than fields in remaining pages. In Fig.4 we can see the fitness behavior after 100 generations. Each curve represent the average of 5 different runs for each population size.

5.3 Qualitative Analysis

As result of our algorithm we optimized a typical form of Poste Italiane made of 8 different fields.

We aimed at arranging fields in different pages according to some constraints to satisfy as: number of pages has to be between 2 and 4; *Number of Checking account* and *Receiver Name* have to belong to the same page; *City* and *ZIP Code* have to belong to the same page; inserting *Description* on the latter pages, preferably in the last one; placing *Amount* on the bottom of page and placing *Sender Name* on the top of page. Furthermore we wish to arrange the fields according to the second case study (see Section 5.2). The proposed algorithm evolved for 100 generations made by 200 individuals and proposes a solution for the specified problem as depicted in Fig.5.

Figure 5: Filling out a form on a mobile device in order to submit a money order: The result of our algorithm.

The fields are split in three different pages (it satisfies the first constraint) with a specific order. Furthermore we can observe how the imposed constraints are satisfied, there is no scrolling in the page and the blank areas are minimized. Experimentally we prove that if we fill out the fields of the first page we can see how it automatically predicts the following two pages and within the single pages the first field is holden by the most predictive one which respects the specified constraints. The fitness value of this solution is 0.96848, confirming a good trade-off between the conflicting criteria.

6 CONCLUSIONS AND FUTURE WORK

Designing a graphical user interface for mobile devices must face many issues related to the limited screen and keyboard size. This entails that assumptions used for designing desktop applications should be revised in the spite of these constraints. Among them there are form scrolling features and data entry. Both can considerably affect application usability. Indeed, scrolling a form on mobile devices can become an annoying experience, as being better to move among pages. Higher usability on mobile devices can also be improved by limiting the data entry effort, employing predictive techniques regarding values the user will enter in the form. In this paper we investigated the use of genetic algorithms in order to automatically support the process of designing a more usable form for mobile devices, that is how to organize fields on different pages considering their predictive power in determining the following field values. Experimental results proved that this approach is robust and can support design decision. These results encourage in pursuing the research direction of investigating how to automatically organize a user interface, thus supporting a designer in choosing among different alternatives. In particular, we aim at exploring regards the optimization of more general Web pages for mobile devices using genetic algorithms, and meta-heuristics more in general.

ACKNOWLEDGEMENTS

This work was partially supported by MIUR Project Automatic System For The Visually Impaired (SAPI), n.1642-2006.

REFERENCES

Ahmad, A. R., Basir, O. A., and Hassanein, K. (2003). Fuzzy inferencing in the web page layout design. In Bézivin, J., Hu, J., and Tari, Z., editors, *WSMAI*, pages 33–41. ICEIS Press.

Baluja, S. (2006). Browsing on small screens: recasting web-page segmentation into an efficient machine learning framework. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 33–42, New York, NY, USA. ACM Press.

Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, 44:145–159.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley.

Jokela, T., Koivumaa, J., Pirkola, J., Salminen, P., and Kantola, N. (2006). Methods for quantitative usability requirements: a case study on the development of the user interface of a mobile phone. *Personal Ubiquitous Comput.*, 10(6):345–355.

Lentz, L. and de Jong, M. (2005). Name and address forms on the web: research into usability. In *Proceedings of the STC's annual conference*, pages 268–272, Arlington, VA, USA.

Nandi, A. and Jagadish, H. V. (2007). Effective phrase prediction. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, pages 219–230. VLDB Endowment.

Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann, San Francisco, CA, USA.

Troiano, L., Birtolo, C., Armenise, R., and Cirillo, G. (2008). Optimization of menu layouts by means of genetic algorithms. In van Hemert, J. I. and Cotta, C., editors, *EvoCOP*, volume 4972 of *Lecture Notes in Computer Science*, pages 242–253. Springer.

W3C (2008). Mobile web best practices 1.0. Technical report, W3C Recommendation. available at http://www.w3.org/TR/mobile-bp/.

Wroblewski, L. (2008). *Web Form Design: Filling in the Blanks*. Louis Rosenfeld.