Human Guided Evolution of XUL User Interfaces

Juan C. Quiroz
Evolutionary Computing Systems Lab
University of Nevada, Reno
Reno, NV 89557 USA
quiroz@cse.unr.edu

Sergiu M. Dascalu
Evolutionary Computing Systems Lab
University of Nevada, Reno
Reno, NV 89557 USA
dascalus@cse.unr.edu

Sushil J. Louis
Evolutionary Computing Systems Lab
University of Nevada, Reno
Reno, NV 89557 USA
sushil@cse.unr.edu

Abstract

Graphical user interface design is a time consuming, expensive, and complex software design process. User interface design is both art and science in that we use both objective and subjective design metrics to evaluate interfaces. An automated process that relies on both subjective and objective metrics to guide the evolution of effective, personalized user interfaces could significantly change current GUI development and maintenance practice. This paper uses an interactive genetic algorithm to evolve XUL user interface layouts by combining objective and subjective metrics. The genetic algorithm encodes expert knowledge from prominent usability guidelines as objective heuristics. Further, the graphical user interface developer (or user!) biases and guides the evolution of the interfaces by subjectively evaluating and selecting the “best” and “worst” interfaces from a small set of displayed interface prototypes. We explore how the selection of individuals from the population to be displayed to the user for subjective evaluation affects the convergence of the genetic algorithm and show that our methodology can produce effective interfaces that reflect subjective user-preferred aesthetics.

Keywords

User interface design, XUL, style guidelines, interactive genetic algorithm

ACM Classification Keywords

H5.2. [Information interfaces and presentation]: User Interfaces – Graphical user interfaces; Style guides

Introduction

User interface (UI) design is an expensive, complex, and time consuming process. It is driven by guidelines of style and design principles (metrics), which are meant to be used as a set of rules to which UI designers should pay attention to and that can be used to evaluate a user interface design. However, guidelines do not always apply to the problem at hand since "very little knowledge in design generalizes beyond specific case studies" [8]. Thus, due to the lack of a formal process in UI design, we tend to see designers being guided by objective measures, obtained from guidelines (e.g., [1,2,4,9]), and by subjective measures, such as aesthetics and the look and feel of an interface.

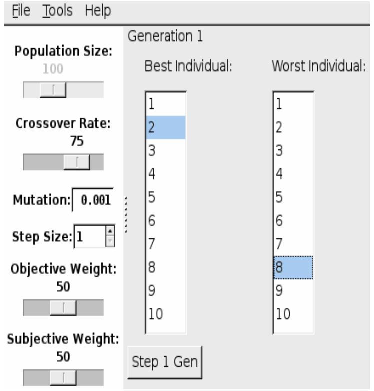
We present an approach that allows the user to incorporate expert knowledge, in the form of objective design metrics or guidelines (e.g., positioning of widgets) and subjective human preferences (e.g., choice of colors), into the UI design process through an interactive genetic algorithm (IGA). Genetic algorithms (GAs) are search algorithms based on the principles of genetics and natural selection [3]. GAs consist of a population of individuals, where each individual is a potential solution to the problem being solved. In contrast to a GA, an IGA allows the user to guide the evolution of solutions through human subjective input.

In our research approach, we encode UI layouts as individuals in an IGA, and evolve the UI layouts over a number of generations to explore the space of UIs. The UI layouts are displayed for the evaluation of the user (a user interface designer), who is asked to choose two layouts: the one considered the best and the one considered the worst. This is subsequently used to evaluate the current population and to create the next generation of UI layouts.

The number of individuals displayed is a subset of the entire population of the GA. The composition of the subset displayed for user evaluation creates rich dynamics that affect the behavior and the convergence of the population in the GA. So far, we have investigated three alternatives for displaying the individuals in the population for user evaluation: displaying the best individuals, displaying random individuals, and displaying both the best and the worst individuals. The results obtained show that displaying a subset consisting of the best individuals in the population yields better and faster convergence of the population to the user desired goal. In terms of related work, few reports are available in the literature. The evolution of website styles was explored in [6], while our work is focused on the evolution of layout and style of GUI widgets.

XUL UIs

The user interfaces evolved were written in XUL, the XML User-interface Language, a cross-platform markup language for user interfaces [10]. XUL is a powerful and extensive language, allowing the defining of the appearance of widgets through CSS style sheets and the use of JavaScript to implement the widget behaviors [10]. XUL is used as the target language



Population Size: 100

Crossover Rate: 75

Mutation: 0.001

Step Size: 1

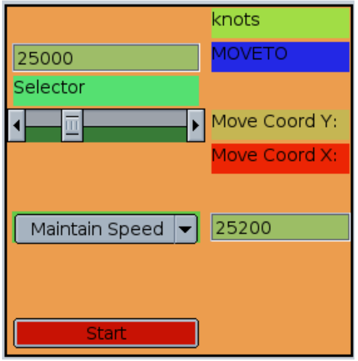
Objective Weight: 50

Subjective Weight: 50

Generation 1

Best Individual:	Worst Individual:
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

Main window of the user interface evolution software.



UIs at generation 0 start with random colors and positions for every widget. Widgets are organized in a grid of 2 rows by 10 columns.

because of its flexibility and the ease with which widgets can be manipulated. XUL is also suitable for the manipulation necessary to evolve the structure of UI layouts. Due to the simple syntax and structure of XUL, one can create a wide range of UIs, from simple UIs consisting of a couple of buttons to complex UIs that incorporate a plethora of widget controls.

UI Evolution Environment

The environment developed for our research provides a front end (shown on this column's left side margin) to an interactive genetic algorithm. The user specifies the UI to be evolved by loading a XUL file consisting of a list of the widgets that make up the UI to be evolved. This makes our tool powerful, since one can evolve as complex a UI as desired, with the same base code.

Once the UI is loaded, the user is able to customize the parameters of the IGA, including population size, crossover rate, mutation rate, selection algorithm, the number of individuals to display for user evaluation, and the frequency of user input.

Fitness Evaluation

The fitness of a UI in the population consists of a linear weighted sum of its objective and subjective components. The weights of these two components are complements of each other adding up to 1. Thus, given a weight of *x* for the objective component, the weight of the subjective component would be *1-x*, where *x* is a number between 0 and 1. The tool allows the user to set the relative weights of the objective and subjective components (criteria). These weights specify the importance that should be attributed to the corresponding component during the fitness evaluation of the generated UIs. For example, a weight of 0.5 for both the objective and subjective components would equally balance the user input and the objective design criteria during fitness computation. For the experiments discussed in this paper we used weights of 0.5 and 0.5 for the objective and subjective components respectively.

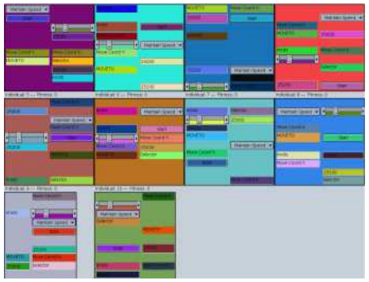
Objective Metrics

We have encoded three metrics taken from guidelines of style [1,2,4,9] and incorporated them as expert knowledge into the IGA: 1) there should be a high contrast between the background color and foreground color; 2) there should be a low contrast between widget colors; and 3) widgets should be aligned with each other. Widgets in our layouts are organized in a grid construct, thus implicitly enforcing the alignment guideline. A sample UI is shown on the left margin.

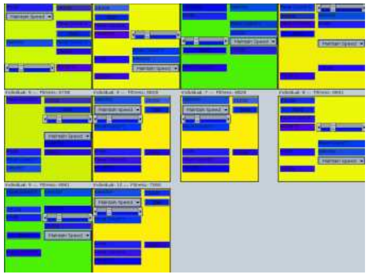
Research Questions

We address several questions with the research tool described previously. First, who in the population should we display for user evaluation? Most importantly, how does our selection of individuals for user evaluation affect the population dynamics? We have chosen three methods of selecting individuals to be displayed for user evaluation: displaying the best individuals in the population, displaying both the best and the worst individuals in the population, and displaying random individuals in the population.

Displaying the best and worst individuals in the population seems to cater to the way the user provides feedback by choosing the best and worst from the individuals displayed. Displaying only the best individuals for user evaluation is an interesting case because it shows where the population is heading; also,



UIs at generation 0 (shown are 10 UIs, members of the population). Each UI starts with a random position and color for each widget.



Evolved UIs at generation 200. The UIs have blue widgets, as was the assumed user preference (a subjective design criterion). Also, the UIs have high contrast between background and foreground (the result of an objective design and evaluation criterion).

as the population converges the best individuals tend to be similar. So having the user choose the best and worst UIs from very similar individuals leads us to believe that it might affect negatively the evolution process by misleading the population convergence and causing it to falter. Displaying random individuals provides us with a benchmark, which can show whether who we display for user evaluation can affect the population convergence and solutions found.

A second and more challenging question, is determining what the size of the population subset for user evaluation should be. The size of the display subset can affect the convergence of the population and the diversity in the population. A large display subset presents the user with a wider variety and higher insight into the current state of the population, at least in the earlier generations of the population. However, this increases the computation and psychological burden on the user since the user is forced to evaluate a lot of individuals every generation. There is also the screen space constraint, limiting the amount of individuals that can be displayed at one time. A small display subset also has disadvantages: 1) it might not provide the user with sufficient insight into the current state of the population; 2) it might not present the user with sufficient variety from which the user can pick something to his/her liking; and 3) in later generations the UIs presented might all be too similar, which would make it difficult to choose the UI the user likes the best and the UI the user likes the least.

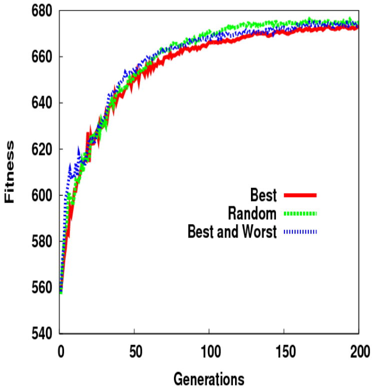
One of the first assumptions we made was that the user chooses the best and worst UIs from the individuals presented for user evaluation. By doing so, we assign the highest fitness to the best individual and the lowest fitness to the worst individual. Every other individual is assigned a fitness value through interpolation by comparing it to the user chosen best and worst individuals. Other methods include ranking all individuals in the population [7], but by choosing only the best and worst we can reduce user fatigue by lessening the amount of user input needed every generation. However, we would like to explore whether the GA is able to converge to a satisfactory optimum by picking only either the best or the worst individual in the population, further reducing the amount of input to one selection every generation and have this serve as the feedback which guides the evolution of the UIs.

Case Study: Lagoon UI

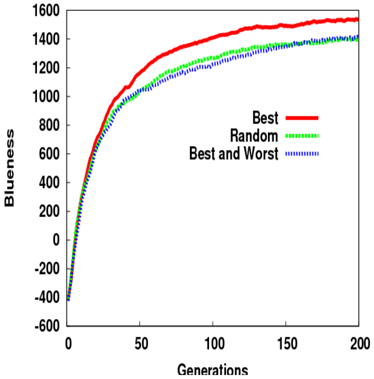
Lagoon is a real-time 3D naval combat simulation game developed at the Evolutionary Computing Systems Lab (ECSL) at UNR as a platform for AI research [5]. We have tested our UI evolution approach with a small panel from the complex Lagoon UI, the "MoveTo" panel that controls combat ships, shown in the generated UIs on this column's left side margin figures. The widgets in the MoveTo panel were written in XUL and loaded into our research tool. The MoveTo panel was chosen because it has a variety of widgets, yet it is simple enough to be used in our initial tests.

Experimental Setup

Experiments were conducted to test two hypotheses: (1) displaying the best individuals for user evaluation results in the best IGA performance; and (2) the user is able to evolve individuals that reflect his/her preferences by only picking the best and worst from a small subset of UIs displayed. We tested three methods for selecting a subset of *n* individuals from the population to be displayed for user evaluation:



Fitness convergence of average individuals.



Blueness convergence of average individuals.

displaying the best *n* individuals, displaying the best *n/2* and the worst *n/2* individuals, and displaying *n* random individuals.

We ran the GA over 30 independent runs with each of the three displaying methods discussed. We used tournament selection, a population size of 100, and 10 individuals displayed every generation for user evaluation. Instead of having a user evaluate individuals for hundreds of generations for all GA runs, we simulated the user input. We made the assumption that the user would always choose the UI that had the "most blue" widgets, which was implemented in a greedy fashion.

Results

Figure 1 shows a fitness convergence plot of the best individuals in the population. We can see that by displaying the best individuals for user evaluation we are able to find a better optimum. This supports our first hypothesis that displaying the best individuals results in the most effective GA performance. The top plot on this column's left margin shows the performance of the average individuals in the population, indicating that all three display methods perform similarly in terms of the population average.

Figure 2 shows the convergence of the best individuals in the population to UIs with blue widgets, which is the user assumed preference. It can be seen that all three display methods result in similar performance, except for in later generations where displaying the best and worst individuals increases slightly. The bottom plot on the left margin shows the performance of the average individuals in the population. Here we see that displaying the best individuals for user evaluation gives the best convergence to blue UIs, that is, the most effective user bias. The blueness convergence plots show the effectiveness with which the user is able to guide the evolution of UIs to their preferences. This supports our second hypothesis, that the user is able to evolve UIs that reflect their preferences by only selecting the best and worst individuals from the subset displayed, instead of ranking all individuals in the subset, as has been done in other IGA studies and applications [7].

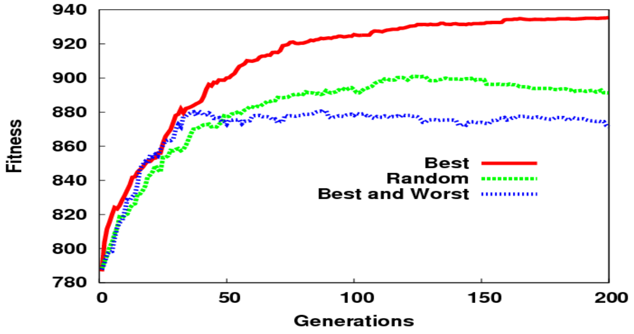


Figure 1. Fitness convergence of the best individuals.

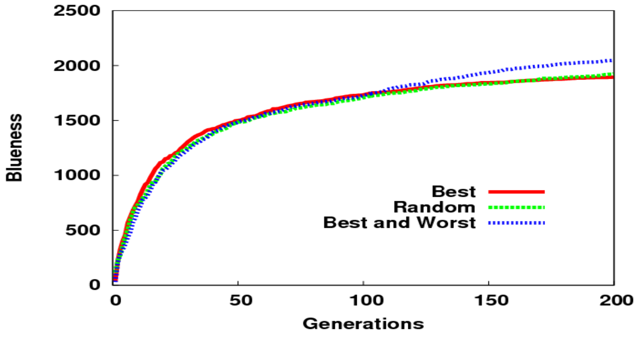


Figure 2. Blueness convergence of the best individuals.

Conclusions and Future Work

We have presented an evolutionary approach to UI design which incorporates both expert knowledge and human subjective input. The experiments presented demonstrate that displaying a subset consisting of the best individuals in the population results in the best IGA performance. We also showed that the user is able to effectively guide the evolution of UIs by picking only the best and worst individuals from the displayed subset.

Evolutionary UI design is a promising direction of future work. First, we would like to conduct user studies in order to assess the utility of the tool. We also plan to see the type of UIs evolved by the users and find out whether users find the tool useful. Currently, widgets in the UI evolve their color and position; we would like to expand on the characteristics that are evolved and also to incorporate further metrics into the objective design and evaluation criteria.

Finally, we wish to expand the degree of human input and specification of the UIs. We wish to enable the user to specify higher-level constraints and declarations and have the tool come up with the necessary widgets and evolve the layout of these widgets. There should also be the ability to specify high level grouping of widgets, such as the grouping of a label with a corresponding textbox. Ideally, we would like the user to input the type of data that needs to be represented by the UI, and then have our tool evolve both the widgets used to represent the data and the layout of the widgets chosen.

In our view, using evolutionary computing for user-guided generation of UIs has tremendous potential in terms of increasing UI design productivity while following desired guidelines of styles and user preferences.

Acknowledgements

This work was supported in part by grant number 00014-05-1-0709 from the Office of Naval Research, USA.

References

[1] Apple Human Interface Design Guidelines. http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHIGuidelines/

[2] GNOME Human Interface Guidelines 2.0. http://developer.gnome.org/projects/gup/hig/

[3] Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[4] Java Look and Feel Design Guidelines. http://java.sun.com/products/jlf/ed2/book/

[5] Lagoon. http://lagoon.cse.unr.edu

[6] Monmarché, N., Nocent, G., Slimane, M., Venturini, G., and Santini, P. Imagine: a tool for generating HTML style sheets with an interactive genetic algorithm based on genes frequencies. In *Proc. of IEEE Intl. Conf. on Systems, Man and Cybernetics* (1999), 640-645.

[7] Takagi, H. Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation. *Proc. of IEEE*, *89*, 9 (2001), 1275-1296.

[8] Thimbleby, H. User Interface Design with Matrix Algebra. *ACM Trans. on Computer-Human Interaction 11*, 2 (2004), 181-236.

[9] Windows XP – Guidelines for Applications. http://www.microsoft.com/whdc/Resources/windowsxp

[10] XULPlanet.com. http://xulplanet.com.