Generative UI Design in SAPI Project

Giuseppina Russo
Poste Italiane
Chief Information Office
P.zza Matteotti 3
80133 Napoli, Italy
russog1@posteitaliane.it

Cosimo Birtolo
Poste Italiane
Chief Information Office
P.zza Matteotti 3
80133 Napoli, Italy
birtoloc@posteitaliane.it

Luigi Troiano
University of Sannio
RCOST – Dept. of Engineering
Viale Traiano
82100 Benevento, Italy
troiano@unisannio.it

CHI 2008, April 5 – April 10, 2008, Florence, Italy
ACM 978-1-60558-012-8/08/04.

Abstract
In this paper we provide an overview of the SAPI project, an initiative of Poste Italiane in collaboration with RCOST-University of Sannio, ITSLab and CRIAI, supported by MIUR, Italian Ministry of University and Research. The project is aimed at developing a software platform, that implementing a novel approach to the user interface generation centered on search based techniques, will provide automatic and semi-automatic adaptation of user interface to vision impaired people.

ACM Classification
H.5.2 User Interfaces; I.2.8 Problem Solving, Control Methods, and Search; D.2.11 Software Architectures.

Keywords
Autonomic Computing, Evolutionary Algorithms, Intelligent Services

Introduction
Vision impairment, or low vision, is a problem that affects with a different level of severity large portions of population, stating an issue for user interface design and engineering to concern about. Vision impairment is due to a class of diseases including: macular degeneration, cataract and glaucoma, and color vision deficiency. These disorders can limit the access to information society, especially for old-age people. Improving interface design can reduce such a barrier. The common solution is to

develop and adopt accessibility guidelines, aimed at removing barriers to any severity level of vision impairment. However, this can lead to solutions that are not aesthetically satisfactory for normal users, or for people with a lower severe vision disorder. So, an accessible version is often provided along the standard user interface. Moreover, interfaces are generally designed by people without vision disorders that can only have a vague idea of vision limitations, so that providing an accessible user interface can result in a non-easy task to accomplish, especially when interface has to be deployed to a large set of devices (e.g. PDAs, laptops, televisions) as those available on the consumer electronic market, under different environmental conditions (e.g. domestic, workplace, public). This would require to think about a large number of user interfaces, that makes any attempt of finely adapting interfaces very time consuming and cost ineffective. Generative techniques (e.g. those based on evolutionary algorithms), represent a promising research direction and solution to the need for self-adapting user interface to specific vision limitations, or even in assisting the interface design, covering a larger class of vision disorders and target devices.

Aim

Objective of SAPI (Sistema Automatico Per Ipovedenti - Automatic System for the Visually Impaired) project [6] is the development of a software platform aimed at delivering services and experimenting novel generative techniques for the automatic provision of user interfaces to different classes of users with visions disorders by means of different devices.

The approach adopted by SAPI is inspired to the principles of "Universal Design" which replaces the concept of "Average User" with respect to the various user skill levels, needs and preferences [6]. Traditional methodologies to the user interface development, such as those based on incremental developments centred on focus groups, appeared soon inappropriate, as they are very time and budget consuming. In order to produce a larger set of interfaces, better tailored to user needs and device features, SAPI is focusing research activities in experimenting generative techniques (in particular based on evolutionary algorithms) as a promising and viable solution.

Innovation aspects

User Interface (UI) design is an expensive, complex, and time consuming process usually driven by documented style guidelines and design principles. Many of these guidelines and design principles are difficult to translate into code, and good UI design is driven in large part by human aesthetics and experience. Furthermore, "very little knowledge in design generalizes beyond specific case studies". Thus UI designers tend to be guided both by objective measures gleaned from UI style guidelines and design principles, and by subjective measures such as the "look" and "feel" of an interface [5].

Designers usually use guidelines to organize the layout and the features of user interface. Existing guidelines, such as Apple's Human Interface Guidelines [1] and Sun's Java Look and Feel Guidelines [7] are either too specific or too vague, so they do not always apply to the problem at hand [9]. Recent works [4][5][9] use meta-heuristic and evolutionary techniques to organize a structural element of a interface such as menus or interface layout [4][9] or non structural features such as fonts or colors [5].

Therefore, designing an interface entails a number of decision problems with respects to the structure, attributes and logics. For instance, what is the widgets layout, how to split the user interaction among different frames, choosing the colour palette, are common issues to be addressed during the interface design. As they can be basically reduced to choosing the most appropriate solution among different alternative, each presented as a combination of simpler alternatives, these issues can be regarded as optimization problems aimed at maximizing some utility function. This perspective makes possible to build a bridge between interface design and search algorithms, in order to adopt a generative approach in designing and building user interfaces. This approach takes several positive aspects, among them:

- A larger number of alternatives can be explored, often resulting in surprising solutions, thus supporting pro-actively human creativity and decision making;
- Different quality attributes and guidelines can be considered at a time (by means of a suitable utility function), thus facilitating the trade-off among conflicting criteria;
- Designers are made free to focus on more adding-value tasks, leaving algorithms to finely optimize their choices;
- Interfaces can be automatically adapted to a larger set of devices, and a more specific set of user preferences.

This approach have started to be investigated only recently relatively to some aspects. For instance, Quiroz et al. [5] encode user interfaces as individuals in an Interactive Genetic Algorithms (IGAs), and run over a number of generations to help explore the space of UI designs. They show the user the best and the worst UIs in the population and use a simple interpolation to determine the fitness of every other individual. IGA combines both computable metrics as objective heuristics and human subjective input to guide the evolution of UIs. They organize the layout of widgets using a grid based system and organize the element of UI in rows and columns.

Ichikawa et al. [2] describe manipulation of Web page color for color-deficient viewers. The authors design a fitness function to preserve detail and to minimize the distance between an input color and its corresponding remapped color. They first decompose the page into a hierarchy of colored regions. These spatial relations determine important pairs of colors to be modified. Therefore they minimize the fitness function using a genetic algorithm.

As preliminary experimentation highlighted this approach to be promising, SAPI focuses research efforts on generative techniques aimed at optimising user interfaces and to adapt them to needs of visually impaired users.

As an example of application, consider the problem of fitting a colour palette to the needs of users with vision disorders [10]. These users perceive colours differently from normal users, entailing that, although palette colours provide an appropriate contrast ratio, the perceived colours could not. Therefore, designers are demanded to identify palette variations providing better contrast ratios, still preserving original chromatic choice. In SAPI, this problem has been addressed by exploiting a genetic algorithm to indentify the palette providing better contrast and minimizing chromatic differences. Experimentation showed algorithm being able to converge towards an appropriate solution. This approach, can be also applied to the optimization of palettes for normal users. Indeed, not always chromatic choices meet contrast requirements. Genetic algorithms can support the designer in choosing a palette variation able to increase colour contrast.

Another example is provided by the selection of menu structures that meeting a set of preferences regarding order, number, duplicates and position of items, maximize functionality accessibility [9]. Again, a genetic algorithm can be implemented in order to identify best menu layouts.

Generative techniques are not only limited to genetic algorithms. In some cases, other evolutionary algorithms can better address some classes of problems. For instance, genetic programming [3] is promising to better address the problem of menu generation when preferences are transformed into constraints (e.g. some items must proceed or follow other items). In this case, a solution is required to meet all constraints in order to be valid, and genetic algorithms demonstrated to produce a larger number of invalid structures. Genetic programming can better control the generation of alternative menu layouts, thus speeding up convergence towards a valid solution. Also, other traditional search techniques and meta-heuristics, can be used for implementing a generative approach, as it is more related to ability of exploring the space of alternatives, than to which computational technique is employed.

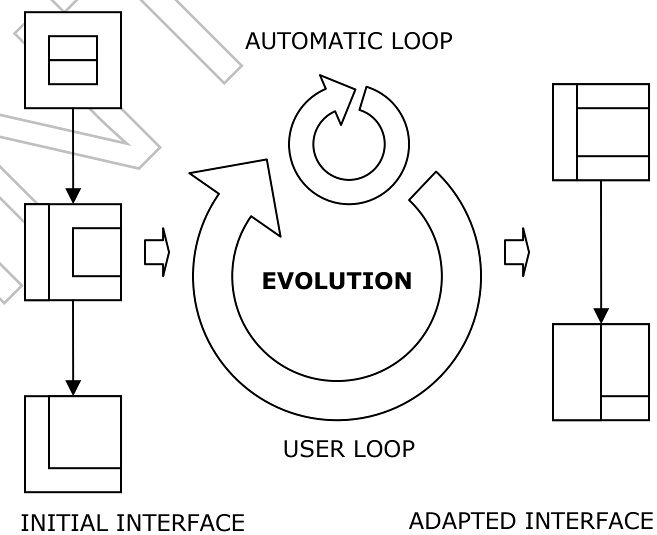

Figure 1 – Interactive interface evolution

The utility function can be implicitly determined by the interface usage. A relevant source of information is provided by the user feedback, both at design and runtime. At design time, user feedback is given by interface designers that selecting alternatives provide information for driving the generation of following alternatives. At runtime, end users will react differently to different interfaces. Also assuming a necessary learning time, this effect is still present at regime, thus providing insights on which solution is better. In both cases, user is put in the optimization loop, becoming an integral part of alternatives search. This leads to experiment the application of interactive evolutionary algorithms as outlined by Takagi [8], and depicted in Figure 1.

In this case evolution is given by two loops: the inner loop aims at automatically screening a set of alternatives, in order to identify potential solutions; a smaller set of solutions is evaluated by the user interaction, leading to the adapted interface.

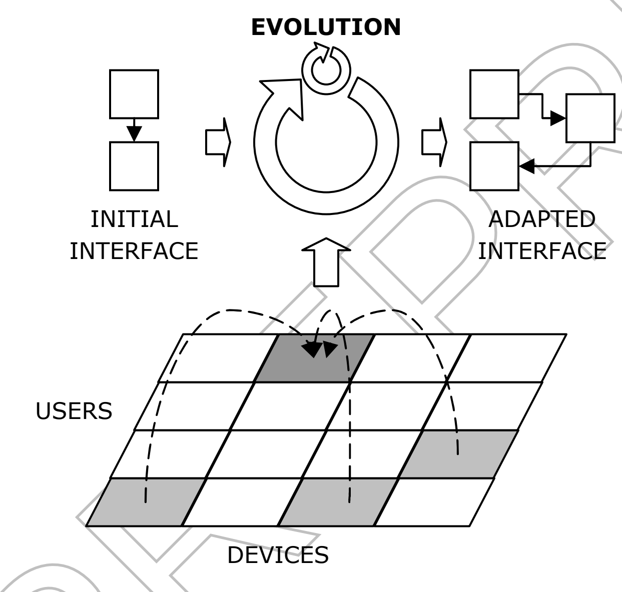

Figure 2 – Transposing interaction data

In interactive search algorithms, a major issue regards how to best use data obtained by the user interaction, as it is not feasible to enquire the user with many alternative interfaces. Large companies, such as Poste Italiane, can benefit of the vast amount of data coming from the interactions of users. However, users differs for age, education, skills, gender, and vision disorders. Therefore data cannot be used directly for evolving the interface, if we are interested to adapt the interface to specific needs and preferences, as outlined in Figure 2. Instead, it is necessary to identify a way for transposing data from one user class to another, as the richer the dataset is, the more effective the evolution can be.

Transposing data is a major challenge in SAPI. Indeed, there exist no model to perform this task, and a wrong transposition can heavily affect the evolution result. We aim to investigate the application of both statistical and inferential models.

Adaptation requires to make a clear distinctions between presentation and logics. Both can be target of optimization. For instance, an online telephone call involves three actions: (i) selection of a telephone operator, (ii) specification of cost and (iii) method of payment. This task can be performed by the user using a single display on a PC with a 15" monitor or, the same service adapted and customised for a visually impaired user using a palmtop may require the three actions to be presented in sequential order on the device display with adaptation of both the presentation and logics.

The Team

SAPI project is developed by the Innovative Services Development Centre of Naples, part of the Chief Information Office of Poste Italiane. The project team is composed by 17 members, most of them with an IT engineering and software development background. SAPI team is constantly supported by research partners, namely RCOST-University of Sannio, CRIAI and ITSLab, responsible for identifying and developing research directions.

Poste Italiane is the leading operator in Italy in the postal services arena. It is also an innovative and competitive operator in the area of financial and payment services. It is able to offer integrated communications, logistic and financial products and service throughout Italy with 14,000 post offices and a staff of 150,000 employees. Fortune Magazine ranked Poste Italiane in the top 10 most admired logistics companies, CISCO in 2007 gave it "best corporate IP network" award.

RCOST is the Research Centre On Software Technology of University of Sannio, targeting excellence in researching software technologies. It is internationally recognized for research in software engineering and computational techniques, taking part to numerous projects both at national and European level.

CRIAI is a research centre in the area of Information and Communication Technologies founded in partnership with the University of Napoli Federico II, with an outstanding record of partnerships and projects.

ITSLab is an international technology provider working in the development of information systems and solutions, focusing on integration of Information, Telecommunication and Multimedia technologies.

References

[1] Apple. Apple human interface guidelines. (2006).

[2] Ichikawa, M., Tanaka, K., Kondo, S., Hiroshima, K., Ichikawa, K., Tanabe, S., and Fukami, K.: Web-Page Color Modification for Barrier - FreeColor Vision with Genetic Algorithm, LNCS 2724, Springer-Verlag, (2003), pp. 2134-2146.

[3] Koza, J.R., Genetic Programming: On the Programming of Computers by means of Natural Selection, MIT Press, Cambridge, (1992).

[4] Oliver, A., Regragui, O., Monmarché, N., Venturini, G.: Genetic and interactive optimization of web sites. The 11th international World wide web conference, Honolulu, Hawaii, USA, (2002) pp. 7-11.

[5] Quiroz, J.C., Louis, S.J., Dascalu, S.M.: Interactive evolution of XUL user interfaces. Proceedings of Genetic and evolutionary computation GECCO 07, London, (2007), pp. 2151-2158.

[6] Russo G., Manfredonia E., Troiano A., Gragnani F.: Il Progetto SAPI, Sistema Automatico per Ipovedenti. Atti della Conferenza Italiana sul Software Libero, 11-13 May 2007, Cosenza, Italy, pp. 149-161.

[7] Sun Microsystems. Java look and feel design guidelines. (2001).

[8] Takagi, H., Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation, Proceedings of the IEEE, Volume 89, Issue 9, Sep 2001, pp. 1275–1296.

[9] Troiano L., Birtolo C., Armenise R., Cirillo G. Optimization of menu layout by means of genetic algorithms, in Proc. of EVOCOP '08, Naples, Italy, LNCS 4972, Mar 2008, pp. 242–253.

[10] Troiano L., Birtolo C., Miranda M.: Adapting Palettes to Color Vision Deficiencies by Genetic Algorithm, in Proc. of Genetic and evolutionary computation GECCO '08, Atlanta, USA (submitted).