Adaptive Interactive Dialogs through the Web: Addressing User's Interaction Requirements Dynamically

Martín González Rodríguez, María del Puerto Paule Ruiz,
Juan Ramón Pérez Pérez, Aitor de la Puente Salan, and
María del Carmen Suárez Torrente

Laboratory of Object Oriented Technologies (OOTLab), Department of Computing Science,
University of Oviedo, c/Calvo Sotelo s/n, 33007 Oviedo, Asturies, Spain
{martinc, paule, jrpp}@pinon.ccu.uniovi.es,
aitor@petra.euitio.uniovi.es, macamen@correo.uniovi.es

Abstract. Addressing the interaction requirements of the users of a web site at its the design stage seems to be an impossibly task as there are too many cognitive, perceptive and motive factors involved. If a web site pretends to be usable for dozens or even hundreds kinds of different users, it must be generated dynamically, depending on the interaction requirements of the current user.

1 Introduction

Traditional design of user centered interfaces is based on the identification and definition of the target audience for the application under development. Some design guidelines include the identification and understanding of the target audience as the most important steps to start the design of a product. The idea is that once the target audience has been defined; an interface that effectively satisfies their needs can be designed.

However, the quest for the typical user is opposite to the individuality and diversity that makes up some much of our identity. If the design of the interaction mechanisms of an application aims to make interfaces accessible and appealing to all users, it shouldn't rely on an abstract generalization [4]. In the design of applications targeted to a wide range of users, it is almost impossible to determine the typical user without falling in serious misconceptions. Maybe is it possible to describe the typical user of generic applications such Microsoft Explorer?

2 Guidelines versus Human Diversity

The uncertainty about the interaction requirements of the real user of an interface makes any decision taken at design stage completely useless. To show this, we would

J.M. Cueva Lovelle et al. (Eds.): ICWE 2003, LNCS 2722, pp. 190–193, 2003.

like to analyze an everyday task in web engineering: the design of the navigation bar (table of contents) for a web portal. We are going to consider the design of the syntactical level for the user interface of the navigation bar, that is, the location and appearance of the bar.

If the designers decide to include buttons in the navigation bar, they will get in trouble simply defining the size of these widgets. Small buttons will be difficult to be used by elderly users or by people with visual disabilities (short-sighted people, for example), while big buttons will look ugly for people with good perceptive systems.

Another pending question that designers must solve is where the navigation bar should be included.

When the World Wide Web became a profitable business and a massive corporate presence was a reality in the net, many graphic designers coming from the printed medium, translated their knowledge to the electronic one, porting many of their design guidelines and principles. One of them, was the relevance that the left side of the visual space of a publication has on a user of a Roman writing system. Notice for example that the headlines included in the covers of western magazines are consider as eye-catching section, acting as a real advert to promote the magazine's content [1].

When the headlines of printed magazines were ported to the web, they were grouped together in a bigger visual object which acted as a table of contents for the site (the navigation bar). Although the navigation bars can be found in any side of a web site, the left side location is still the most popular for graphic designers, as it is supposed to act as attraction pole for –western– readers.

If the designers of our example are working on a web portal for western readers, this guideline will solve the dilemma. However, the solution is not as simple as it seems, as this guideline miss and important feature of the electronic medium that isn't present in the printed one.

Some years ago, we conducted an experiment to corroborate the relevance of the left side of a visual space over the right side [2]. The experiment consisted in a web site from which the volunteers had to obtain certain information. The only way to obtain that information was by mean of the navigation bar. However, the web site had two identical navigation bars, one at each side of the visual space (left and right side).

As both bars were identical in appearance and both conduced to same information, the selection of the preferred bar wasn't performed in terms of cognition but on the relevance of location the bar.

Obviously, the selection of the users was registered by data-gathering agents embedded inside the web pages. The results obtained were amazing, as they didn't showed a clear preference for the left side bar as it was expected. In fact, only 179 users out of 342 selected the left side bar (52,9%). It is a quite relevant fact that 47,5% of the users selected the right side bar.

The experiment was repeated again with similar results, but this time, once users got the information they were looking for, they were asked to indicate whether they were left or right handed. The results obtained probed a clear correlation between right handed users and selections of the right side bar. This was the feature missed by graphic designers when they ported their general design principle to the electronic medium. In the printed medium, readers are attracted through their perceptive system only (visual attraction). However, in the electronic medium, the motive system of the users also plays a crucial in the scoring of the relevance of an objet, as users must select the object by mean of a mechanical action which depends on their laterality.

As the user interface of any application should be usable and accessible by any kind of users (independently of their perceptive system or their laterality), each design decision should be taken at execution time, once that the interaction requirements of the real user have been finally detected. As designers are not present at execution time, the application should be able to emulate the human designer, creating the user interface on the fly.

Following this design principle, we have developed GADEA, a User Interface Management Expert System (UIMES) able to generate dynamic versions of the syntactical level of the interactive dialogs of a user interface at execution time.

3 The Architecture of GADEA

The UIMES GADEA tries to emulate the behavior of a human expert in Human-Computer Interaction (HCI) who is in charge of the design of an interactive dialog for a specific user, employing multimodal communication channels [5] (visual and auditory communication channels). The emulated expert will select the most suitable interaction style available for the target platform (the web or a standalone applications), accordingly with the unique cognitive, perceptive and motive requirements of the target user, adapting the appearance, contents and access mode of every widget included in the dialog.

The internal architecture of GADEA relies on three independent components who has been specially designed to cope with the problems derived of the three most important features of the system. Those components are DEVA, ANTS and CodeX (see Figure 1).

The CodeX (Code eXplorer) module represents the interface between GADEA and the client applications of this UIMES. This module must convert the user interaction requests into calls to specific methods inside the application's domain workspace. To reach this objective, this module automatically inspects the binary code of every client application at execution time, looking for any user process defined by the programmers during the design time. CodeX consider a 'user process' as those methods designed to satisfy a specific user's interaction request and will represent them as options or commands in the respective application user interface. This module is also in charge update every single piece information displayed by the user interface, keeping track of possible changes in the value of public variables and data structures associated to the interface by the application at both execution and design time. All this information is obtained automatically inspecting the client application's code by mean of the structural reflection mechanism of the Java platform.

The information collected by CodeX is sent periodically to DEVA (Dialog Expert Valuator for Adaptation) which represent the agent software in charge of the simulation of the behaviour of human expert in Human-Computer Interaction (HCI). This module converts the application's and user's interaction requests into adaptive interactive dialogs. Based on the general knowledge and guidelines provided by the HCI discipline as well as by specific knowledge stored about the current user model of the client application, DEVA uses its fuzzy inference engine to evaluate, to select and to adapt the best interactive dialog available for the current user. The dynamic creation of the user tailored interactive dialog will depend on the current user's cognitive, perceptive and motor skills, which might fluctuate over the time.

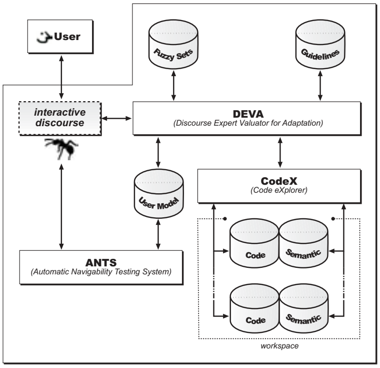

Fig. 1. General Architecture of GADEA, composed by three modules: DEVA, ANTS and CodeX.

Finally, the ANTS module will be the agent responsible for keeping the information about the users of any client application of GADEA updated. This component makes use of different kinds of automatic remote agents designed to observe the user behavior in any of the interactive dialogs designed by DEVA. Those agents obtain crucial information about the specific user's skills, which is stored in specific distributed user model.

References

1. Gordon, Druce; Designing Covers, Mac Format, Future Publishing, Bath, England. Issue: January. 1994.
2. González Rodríguez, Martín; ANTS: An Automatic Navigability Testing Tool for Hypermedia. Correia Nuno; Chambel, Teresa; DAVENPORT Glorianna; (Eds.) Multimedia'99. Springer-Verlag. ISBN 3-211-83437-0. 2000.
3. Nielsen, J.; Alertbox. http://www.useit.com/alertbox/. 2002.
4. Reynolds, C.; A Critical Examination of Separable User Interface Management Systems: Constructs for Individualization. ACM SIGCHI. 29(3). 1997.
5. Wahlster, Wolfang; User and Dialog Models for Multimodal Communication. Sullivan, Joseph W.; Tyler, W. Sherman (Eds.) Intelligent User Interfaces. Frontier Series. ACM Press. 1991.