

Adapting User Interfaces to the User Interaction Requirements in Real Time

Martín González, Marcos González, Jorge Manrubia
The Human Communication and Interaction Research Group (HCI-RG) –
Department of Computer Science, University of Oviedo, Spain
martin@uniovi.es, {marcosgonzalezgallego, Jorge_manrubia}@yahoo.es

Abstract. User interfaces are commonly designed to match the requirements of the so-called ‘typical user’ who represents an abstract generalization of each user of an application. In systems employed by different kinds of users (such as Web systems), the identification of the requirements of their users is very difficult, if not impossible. Our proposal is to avoid the construction of interactive dialogs during the design stage, building them dynamically once the specific cognitive, perceptual and motor requirements of the current user of the system are known: that is, during the execution stage.

1 Introduction

Traditional design of user centered interfaces is based on the identification and definition of the users of the application. Some design guidelines recommend the identification and understanding of the users as the first step to start the design of the product. This design principle shows that only once the requirements of the users have been identified, 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 for all the users, it shouldn’t rely on abstract generalizations [1].

The identification of the typical user of applications targeted to different kinds of users is almost impossible without falling in serious misconceptions. Maybe is it possible to describe the typical user of Microsoft Office? Is it possible to identify the typical user of generic Web sites (such as Google), visited by thousands of different users everyday? For some authors such as [2], the answer to those questions is definitely ‘no’ as ‘there is no average user’. For him, ‘either compromises must be

Please use the following format when citing this chapter:

Gonzalez, Martin, Gonzalez, Marcos, Manrubia, Jorge, 2006, in IFIP International Federation for Information Processing, Volume 204, Artificial Intelligence Applications and Innovations, eds. Maglogiannis, I., Karpouzis, K., Bramer, M., (Boston: Springer), pp. 156–162

made or multiple versions of the system must be created' in order to satisfy the specific interaction requirements of every user.

2 Low Level Adaptive Systems

Designing different interfaces for each kind of user can be extremely costly. Furthermore, this approach causes excessive development costs when maintaining versions across multiple platforms [3]. A much cheaper solution is the development of a unique user interface able to be adapted to each type of user. Literature includes several references to techniques designed to adapt user interfaces.

Several systems and prototypes generate adapted contents depending on the information stored in general user models. Techniques for the presentation of adaptive content usually include conditional inclusion, contents removing and transformation of contents. Kobsa et al [4] developed and adaptive learning framework tailored to the student's presumed level of familiarity with the subject. This system was called KN-AHS (KoNstanz Adaptive Hypertext System).

Weber and [5] developed another adaptive learning system in the form of an adaptive interactive textbook called ELM-ART. This system was able to learn from the student's actions, performing visual adaptive annotation of links based on the actions performed in previous sessions. Another important adaptive learning system is AHA [6] (De Bra and Calvi, 1998). This system not only learns from user actions but also from tests.

In the EU-funded AVANTI project ([7]; [4]), a web-based information system was developed providing universal access to disabled and elderly users. This project represented an important attempt in the use of techniques for the adaptive presentation in order to satisfy the needs of each individual user, including a broad spectrum of users with perceptual and motor disabilities.

PERSONA [1] is an ambitious (and never implemented) proposal for an adaptive user interface management system based on agents called 'tailors'. Each 'tailor' includes a collection of similar components that perform the same tasks within the user interface, but which are specifically designed for one specific kind of users.

Working on the highest levels of the user interface, these adaptive systems do not pay much attention to the syntactical and lexical levels of the interface responsible for the configuration of the interactive objects (size, color, etc.) and how these objects are going to be accessed by the user [8]. The main handicap in the management of the user model in some of these systems is their data acquisition algorithm, which is based on periodical updates. However, a continuous observation of the user behavior could highly increase the amount of features that could be adapted in the user interface, as well as the global accuracy of the system.

3 The Architecture of GADEA

GADEA tries to emulate the behavior of a human expert in Human-Computer Interaction designing all the interactive dialogs required to establish an effective communication between the application and the specific user.

This emulated expert selects the most suitable interaction style available for the target platform (Web-based or standalone applications), adapting the appearance, contents and access mode of each dialog to the user requirements. The result is a multimodal communication channel [9] containing a combination of visual and aural interactive dialogs.

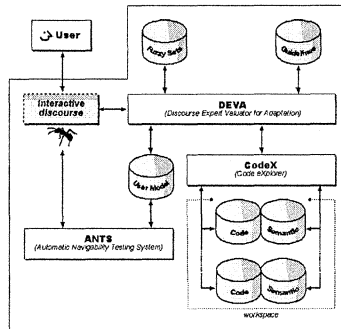


Fig. 1. GADEA uses three main modules: DEVA, ANTS and CodeX. The User Model is shared by ANTS and DEVA. While ANTS is in charge of keeping it updated, DEVA uses it to create user-tailored interactive dialogs.

The architecture of GADEA is based on a two layer design (see figure 1). The upper layer constitutes the user interface's front-end; containing two modules: DEVA (*Dialog Expert eValuator for Adaptation*) and ANTS (*Automatic Navigability Testing System*). The lower layer manages the interaction between the first layer (the user interface) and the application(s); containing one module: CodeX (*Code eXplorer*).

3.1 CodeX (Code eXplorer)

When the user launches the application, CodeX inspects the application's binary code using computational reflection in order to obtain the list of user processes registered by the programmers at the design stage. Those user processes are the actions available to interact with the application. This list is sent to DEVA, which includes the processes in the application's menus. If the user selects a menu option, DEVA will send a notification to CodeX in order to execute the corresponding method.

Whenever a user process requires information from the user, the programmer must design at least one interactive dialog for that process. These dialogs are data containers built at the design stage but filled by the user at the execution time. For

example, if an application requires to know the user favorite color, the corresponding interactive dialog will include a data object filled with the color request message (something like 'What is your favorite color?'), and an empty data object for the user favorite color that will be filled with the data provided by the user at execution time.

3.2 DEVA (Dialog Expert Valuator for Adaptation)

When CodeX executes a user process including an interactive dialog, this dialog is sent to DEVA which will convert this data container into a visual or aural display (depending on the kind of user, it could generate a combination of both). This display uses interactive objects (buttons, text fields, menus, etc.) to interact with the user, filling the information included in the data container with the information provided by the user. The conversion algorithm uses general human-computer interaction guidelines which are adapted to the interaction requirements of the current user of the application using the information stored in the user model.

Depending on the requirements of the current user, DEVA selects the most suitable communication channel (for instance, aural channels for blind users) inserting the most suitable interactive objects (for example, certain disabled users cannot use popup menus, radio buttons or checkboxes) and suitably configuring each of them (for instance, short-sighted users require bigger objects).

In the first step, the interactive objects are selected a search algorithm that obtains the list of the interactive objects compatible with the data object's configuration, evaluating each of them according to cognitive criteria. In our previous example, the list of interactive objects that may be used to get information about the favorite color of the user includes at least a popup menu and a text field (among many others). Any of these interactive objects may accomplish the task. However, DEVA would select a text field object for visually disabled users and pop-up menus for users who prefer a mouse-based interaction (see Figure 2 and Figure 3). The scoring method for the search algorithm is based on the 'Keystroke Model' developed by [10] and redeveloped by [11].

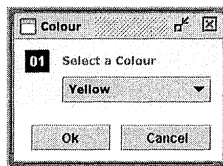


Fig. 2. Interactive dialog created by DEVA when the user prefers a mouse-based interaction or has good pointing skills.

The second step is to design the dialog's layout. We make use of orientation metaphors ([12] cited by [13]) mapping the vertical attribute of any location of the visual space to the 'up is more' and 'down is less' metaphors [13].

The last step defines the size, shape, color, etc. of every interactive object selected. This task is performed using a fuzzy logic engine powered by more than

fifty different rules based on state of the art Human-Computer Interaction guidelines as well as on the data stored in the user model.

Once the user has finished the interaction process with the dynamically generated dialog, the dialog is converted into a data container again (including the information provided by the user during the interaction process) which is sent again to CodeX for a further processing by the application.

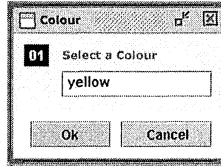


Fig. 3. . Interactive dialog created by DEVA when the user prefers a text-based interaction or has good typing skills.

4.1 ANTS (Automatic Navigability Testing System).

At the same time that the user interacts with the generated dialogs, his actions are continuously recorded and analyzed on the fly by the third module of GADEA: ANTS. The result of this analysis is included in the user model, keeping it updated and therefore, increasing the adaptation accuracy of the system.

Part of the information required to perform the adaptation proposed is provided explicitly by the user when registering in the system. This information includes the user's age, list of user's disabilities (short-sighted user, deaf user, etc.), visual precision (obtained using a simple test), whether the user is lefty or right handed, preferences for an specific kind of interaction (mouse or keyboard) and so on.

Other information required is obtained by a small army of data-gathering agents, which keeps track of the user behavior in every available interactive dialog displayed at execution time. The design metaphor of ANTS is based on the life of a community of ants. The ants (agents) depart from their anthill (a server), looking for food (the information) in every picnic available (interactive dialogs). Once the ant has got the food, it comes back to the anthill and carefully stores the food in one of the available warehouses (the user model).

The ants keep track of every action performed by the user in every user process, creating an abstract navigation model similar to those employed by the Object Oriented Hypermedia Design Model [14]. This navigation model is used to determine the landmarks, routes and mental models [15] employed by the users when they execute the user processes; obtaining the user's expertise degree when working with the application [16].

Some agents are able to analyze the data collected, determining the user's precision in the performance of certain tasks. For example, this system has agents specialized in analyzing the user typing skills (based on the Salthouse's regularities [17]) or the visual form recognition skills (based on Sternberg's regularities 1969, [18]).

5 Testing the System with Real Users

In order to test the precision of this adaptation model, we performed several tests for each feature of the system. As a paradigmatic example of such kind of tests, we are going to describe the tests applied to the chromatic adaptation algorithm of GADEA.

Five basic interactive dialogs were created to test this feature (labeled from v1 up to v5.). Each dialog displayed a different text message. Next, volunteers participating in the tests were grouped depending on their visual precision into five groups labeled from A to E. Finally, GADEA generated an adapted version of each dialog for each group.

The results obtained were successful for users of the groups B and C (adults with normal-low and normal-high visual precision) since the average of the selections performed by the volunteers always favored the versions designed by GADEA for their groups. In the case of the group B an average of 84% of the selections favored the version designed by our system for users of that group. This proportion was a little bit higher for the group C (91.42%).

Results for groups D (teenagers with excellent visual precision) and E (adults with high visual precision) reached 70% and 76% respectively. The worst results were obtained by the group A (elderly people with low visual precision) since only 53.33% of the versions developed by GADEA were selected as the most suitable.

This experiment showed that, although relatively powerful, the chromatic adaptation algorithm requires finer tuning for certain kind of users (especially for elderly people). Such kind of tests also revealed themselves as a powerful tool to measure the accuracy of the adaptive algorithms of our system.

6 Conclusions and Future Work

We have presented a user interface management system which provides full support for the automatic generation of the lexical and syntactical level of the interface. Since these levels are designed dynamically at execution time (once the real user requirements are known) this strategy prevents the design of interactive dialogs based on misconceptions about the users.

Tasks assigned to the designers are limited to the semantic and conceptual levels of the interface, designing the interchange of information between their programs and the user instead of how that information is displayed. This strategy simplifies the amount of work to be done by the programmers.

Several improvements may be done in this basic open adaptation framework. Our tests showed that the usefulness of the system depends on the precision of adaptation algorithms as well as on the precision of the information stored in the user model. The inclusion of new adaptation features to the system is relatively easy. It requires the addition of new rules to the fuzzy inference engine and (if required) new data-gathering agents to obtain updated information about the input parameters used in the rule. However, each new feature added to the system requires a careful testing process to obtain valid results.

7 References

1. C. Reynolds, A Critical Examination of Separable User Interface Management Systems: Constructs for Individualisation. ACM SIGCHI. (1997) 29(3).
2. B. Schneiderman, Designing the user Interface. Reading, MA: Addison-Wesley. (1987).
3. G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, N. Souchon, L. Bouillon, M. Florins, J. Vanderdonckt, Plasticity of User Interfaces: A Revised Reference Framework, Proceedings of the First International Workshop on Task Models and Diagrams for User Interface Design. (2002) 127 – 134.
4. Kobsa, J. Koenemann, W. Pohl, Personalized Hypermedia presentation techniques for improving online customer relationships. The Knowledge Engineering Review. (2001) 111-155.
5. G. Weber, P. Brusilovsky, ELM-ART: An adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education* 12 (4), Special Issue on Adaptive and Intelligent Web-based Educational Systems. (2001) 351-384.
6. P. De Bra, L. Calvi, AHA: a Generic Adaptive Hypermedia System. Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia HYPERTEXT'98, Pittsburgh, USA. (1998).
7. J. Fink, A. Kobsa, A. Nill, Adaptable and Adaptive Information Provision for All Users, Including Disabled and Elderly People. New Review of Hypermedia and Multimedia 4. (1998) 163-188.
8. J. Foley, W. C. Kim, S. Kovacevic, K. Murray, UIDE, An Intelligent User Interface Environment. Intelligent User Interfaces. Frontier Series. ACM Press. (1991).
9. W. Wahlster; User and Dialog Models for Multimodal Communication. Intelligent User Interfaces. Frontier Series. ACM Press. (1991).
10. S. Card, K. Stuart, T. Moran, P. Thomas, A. Newell. The Psychology of Human-Computer Interaction. Lawrence Erlbaum Associates (1983).
11. Newell, A. Unified Theories of Cognition. Harvard University Press. Cambridge, Massachusetts. (1990).
12. G. Lakoff; M. Johnson; Metaphors We Live By. Chicago: University of Chicago Press (1980).
13. E. Shulz; M. Van Alphen; W. Rasnake, Discovering User-generated Metaphors through Usability Testing. Proceedings of the Second International Conference on Cognitive Technology. Aizu, Japan (1997).
14. D. Schwabe, G. Rossi, Abstraction, Composition and Lay-Out Definition Mechanism in OOHDM. Proceedings of the ACM Workshop on Effective Abstractions in Multimedia. San Francisco, California. (1995).
15. S. Jul, G. W. Furnas, Navigation in Electronic Worlds: a CHI'97 Workshop. SIGCHI Bulletin Vol. 29, No. 4, October 1997, ACM Press. (1997).
16. S. Karn, S. Keith; J. Perry, J. Thomas, M. J. Krolczyk, Testing for Power Usability: a CHI 97 Workshop. ACM SIGCHI Bulletin Vol. 29, No. 4, October 1997 (1997).
17. T. Salthouse, Perceptual, Cognitive and Motoric Aspects of Transcription Typing. Psychological Bulletin (1986).
18. S. Sternberg, Memory Scanning: New Findings and Current Controversies. Quarterly Journal of Experimental Psychology (1974).