Quality of Adaptation: User Cognitive Models in Adaptation Quality Assessment

Víctor López-Jaquero, Francisco Montero and Pascual González

Laboratory on User Interaction & Software Engineering (LoUISE)
Instituto de Investigación en Informática (I3A)
University of Castilla-La Mancha, 02071 Albacete, Spain
{ victor | fmontero | pgonzalez }@dsi.uclm.es

Abstract Adaptation capabilities are becoming more and more popular in applications due to several facts, including heterogeneity of users, devices and physical contexts of use where applications are currently used. Adaptation is not per se good, and poor adaptations usually lead to disappointed users that reject or disable adaptation mechanisms. Therefore, in order for adaptation to reach the mainstream mechanisms to guarantee the *quality of adaptation* (QoA) are required. In this paper one of those mechanisms is proposed, which is based on results coming from cognitive models developed in the literature as a mean to asses quality of adaptation.

Keywords: quality of adaptation (QoA), adaptive user interfaces, quality assessment, GOMS, cognitive models, ISATINE framework.

Introduction

In the global village where we are immersed, a growing interest for interaction with computers can be observed in our societies. Nevertheless, as more and more people get attracted to interaction with computers new needs arise. The range of user typologies is widening; expert users are no longer the critical mass of software applications, and people from many different social and cultural backgrounds need to use the same applications. Besides, the number of available platforms to interact with is drastically increasing, providing many different platforms with a great range of variability in their capabilities for graphics, networking, computing power or interaction techniques. Furthermore, the uses that users give to software applications and the ubiquitousness of computing are introducing new exciting possibilities regarding the physical contexts where interaction can take place; users can use their applications at home, in a bar, in the streets or even in the countryside.

This situation has introduced an awesome challenge to software engineers: creating software applications able to work under all those situations illustrated above. Obviously, it is impossible to create a version of each application for each situation; therefore applications need to be able to be adapted or to adapt themselves to the situations described above (or at least to a set of those situations). When it is the user the actor in charge of adapting the application to the new situation, the application is called adaptable. On the other hand, when it is the application itself the actor that makes the adaptation automatically, the application is said to be adaptive [1].

Nevertheless, adaptation per se is not necessary good. There are numerous examples of adaptive systems that failed and were rejected by users.

Thus, not every adaptation is valid, but just those adaptations which are good enough for the user interacting with the application being adapted. Therefore, it is necessary to devise approaches supporting the assessment of the quality of an adaptation with respect to the user.

In this paper, an assessment method for the evaluation of the *Quality of Adaptation* (QoA) is presented, that takes into consideration the cognitive model of users.

The Adaptation Process

To achieve the goal of adapting a user interface (UI) a series of steps are required. The most accepted sequence of steps for adaptation is the one proposed in by Dieterich in [5]. The four steps proposed in this adaptation framework are:

1. Initiative: one of the entities involved in the interaction suggests its intention to perform an adaptation. The main entities are usually the user and the system.
2. Proposal: if a need for adaptation arises, it is necessary to make proposals of adaptations that could be applied successfully in the current context of use for that need for adaptation detected.
3. Decision: as we may have different proposals from the previous stage, which adaptation proposal best fit the need for adaptation detected should be decided, and whether it is worth applying each proposal.

4. Execution: finally, the adaptation proposal chosen will be executed. One important factor when making any changes in the UI is how the transition from the original UI to the adapted one is performed. Before the execution stage, a prologue can be executed to prepare the UI for the adaptation. For instance, if the adaptation includes switching from one code to another code, the prologue function should store the current state of the application, so it can be resumed after the adaptation takes places. On the other hand, an epilogue function can be provided to restore the system after adaptation takes place. This epilogue will take care of restoring application state and resuming the execution of the application.

However, these four steps suffer from several shortcomings, being the most relevant to this paper the inability of this adaptation process to consider the evaluation of the adaptation. This drawback, among some others, led us to propose a new framework for adaptation called ISATINE [7], which specializes Norman's mental model [9] for adaptation.

ISATINE: A Framework for Adaptation

The specialization of Norman's model for adaptation results into the ISATINE framework, whose seventh stage, Evaluation of adaptation, includes the assessment of the quality of the adaptation produced. The work described in this paper can be framed into this seventh stage of ISATINE framework. Next, a description of all seven stages of adaptation according to ISATINE framework is presented:

1. *Goals for user interface adaptation*: any entity (U, S, or T) may be responsible for establishing and maintaining up-to-date a series of goals to ensure user interface adaptation. Although this adaptation is always for the final benefit of the user, it could be achieved with respect to any aspect of the context of use (with respect to the user herself, the computing platform used by the user, or the complete physical and organizational environment in which the user is carrying out her task). The goals are said to be *self-expressed*, *machine-expressed*, *locally* or *remotely*, depending on their location: in the user's head (U), in the local system (S), or in a remote system (T).
2. ***I**nitiative for adaptation*: this stage is further refined into formulation for an adaptation request, detection of an adaptation need, and notification for an adaptation request, depending on their location: respectively, U, S, or T. For example, T could be responsible for initiating an adaptation when an update of the UI is made available or there is a change of

context that cannot be detected by the system itself (e.g., an external event).

3. ***S**pecification of adaptation*: this stage is further refined in specification by demonstration, by computation, or by definition, depending on their origin: respectively, U, S, or T. When the user wants to adapt the UI, she should be able to specify the actions required to make this adaptation, such as by programming by demonstration or by designating the adaptation operations required. When the system is responsible for this stage, it should be able to compute one or several adaptation proposals depending on the context information available. When the third party specifies the adaptation, a simple definition of these operations could be sent to the interactive system so as to execute them.
4. ***A**pplication of adaptation*: this stage specifies which entity will apply the adaptation specified in the previous stage. Since this adaptation is always applied on the UI, this UI should always provide some mechanism to support it. If U applies the adaptation (e.g., through UI options, customization, personalization), it should be still possible to do it through some UI mechanisms.
5. ***T**ransition with adaptation*: this stage specifies which entity will ensure a smooth transition between the UI before and after adaptation. For instance, if S is responsible for this stage, it could provide some visualization techniques, which will visualize the steps, executed for the transition, e.g., through animation, morphing, progressive rendering [15].
6. ***IN**terpretation of adaptation*: this stage specifies which entity will produce meaningful information in order to facilitate the understanding of the adaptation by other entities. Typically, when S performs some adaptation without explanation, U does not necessarily understand why this type of adaptation has been performed. Conversely, when U performs some adaptation, she should tell the system how to interpret this evaluation.
7. ***E**valuation of adaptation*: this stage specifies the entity responsible for evaluating the quality of the adaptation performed so that it will be possible to check whether or not the goals initially specified are met. For instance, if S maintained some internal plan of goals, it should be able to update this plan according to the adaptations applied so far. If the goals are in the users' mind, they could be also evaluated with respect to what has been conducted in the previous stages. In this case, the explanation of the adaptation conducted also contributes to the goals update. Collaboration between S and U could be also imagined for this purpose.

Adaptation Quality Assessment

Quality of Adaptation (QoA) is the extent to which a set of adaptations produce a user interface to achieve specified goals with usability (ISO 9241-11; ISO 9126-1) in a specified context of use.

In this work the context of use is conformed by the characteristics of the user, the platform (both hardware and software), the physical environment where interaction takes place and the current task the user is doing.

The adaptation engine should guarantee a certain degree of QoA, similarly to the way QoS[1] (Quality of Service) does for networks data flows. No adaptation should be applied that would produce a QoA value bellow the guaranteed threshold.

To address the assessment of the quality of each adaptation applied, a description of which parameters characterize the quality of adaptation is required. In this work the quality of adaptation is parameterized in terms of two concepts: migration cost and adaptation benefit. Migration cost [3] represents the physical, cognitive and conative effort the user requires to migrate from one context to another. Adaptation benefit represents how good an adaptation will be for the user in the new context.

For the evaluation of the migration cost an association of the components in the definition with the criteria used in their assessment has been made. In the current version physical effort has been left out, because it is relevant just in some interaction environments, such as virtual environments where the effort required to manipulate the equipment (Head-Mounted Displays, Data Globes, etc) is really relevant in the evaluation of the interaction.

The cognitive effort is assessed by means of two criteria: discontinuity and cognitive load, while conative effort is assessed according to user preferences.

Discontinuity appears when the user is forced to divide his attention between two entities. For instance, continuity would be preserved if an adaptation would replace a *widget* with another one which occupies the same screen area than the original, because the user does not need to draw his attention from the same screen area. However, if a small *widget*, for instance a *comboBox*, is replaced with a group of radio buttons with a medium amount of items, the user will need to regain the focus on the right place before continuing using normally the UI, to be able to cope with the noti-

---

[1] http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.htm

ceable change in the size of the screen area where the user focus needs to be.

Cognitive load in a UI adaptation is represented as the increase/decrease of the amount of information the user is asked to process in order to perform his current tasks. Therefore, those adaptations directly affecting the amount of information shown (i.e. when applying techniques such as *stretch text* o *accordion* [2]) have an impact in the cognitive load.

Finally, user preferences will be applied as a corrective measure to the metrics for migration cost previously discussed. Thus, for example, if the user has some kind of visual impairment and prefers a textual representation of information, despite it implies a higher cognitive load, the migration cost will be weighted to reflect the aforementioned user preference.

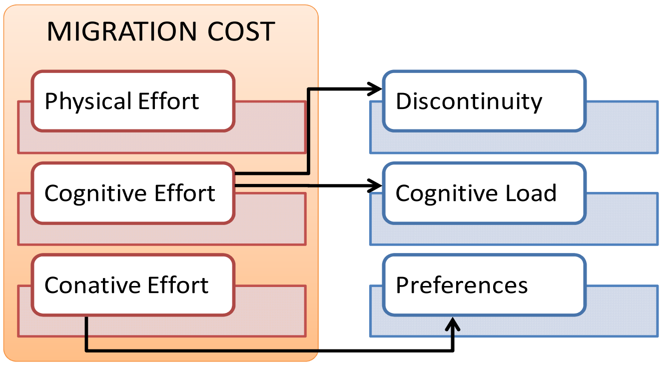


Fig. 1. Migration cost components and their relationship with the assessment criteria.

Discontinuity is assessed according to the mental effort required to resume the activity interrupted by the adaptation process. To support the evaluation of this mental effort an estimation based on interaction cognitive modeling, an especially in the empirical and theoretical results obtained from human-computer interaction is used. More exactly, the evaluation is based on results obtained by means of GOMS-inspired techniques [4].

GOMS-Based Interaction Time Estimation

Cognitive models are mainly used to make predictions about how a person will behave given a specific situation. In human-computer interaction the most extended one is GOMS (Goals, Operators, Methods and Selection

Rules) [4]. GOMS is based on the Human Processor model [4]. This model provides a hierarchical specification describing how to reach a set of goals in terms of operators, methods and selection rules. The operators can be perceptual, motorial or cognitive acts. Methods are sequences of subgoals and operators used to structure the specification of how to reach a goal. Finally, the selection rules provide a means to decide which method should be used in a given situation to reach a goal (if several methods are available to reach that goal).

GOMS supports [8], among other things, estimating the time required to perform a task and finding out what interaction steps take longer or are more error-prone. By using some GOMS specifications and the final applications for them, an estimation of each basic interaction task was achieved.

The values empirically estimated by Card, Moran, Newell et al. [4][8][10] used within the assessment of adaptation quality in this work are summarized in Table 1.

Parameter	**Estimated time**
Enter a keystroke	230ms
Point with a Mouse	150ms
Move hands from keyboard to pointing device	360ms
Move hands from mouse to keyboard	360ms
Perceive a change	100ms
Make a saccade[2]	230ms
Recognize a 6-letter word	340ms

Table 1. Summary of estimated times for some basic interaction tasks.

Discontinuity Evaluation

Evaluating the discontinuity produced due to a UI adaptation includes the evaluation of the different influences that an adaptation can produce on the UI resulting in some kind of discontinuity. It is also necessary to consider that as a result of an action different collateral effects can arise.

In Table 2 the adaptation effects considered in the context of this work are listed. Notice several effects can appear at the same time. For instance, the size of a *widget* could change at the same time it is moved. Collateral ef-

[2] Time to position and get the information in each jump.

fects can appear also. For instance, when a *widget* is removed, the rest of the *widgets* can move to occupy the screen space freed.

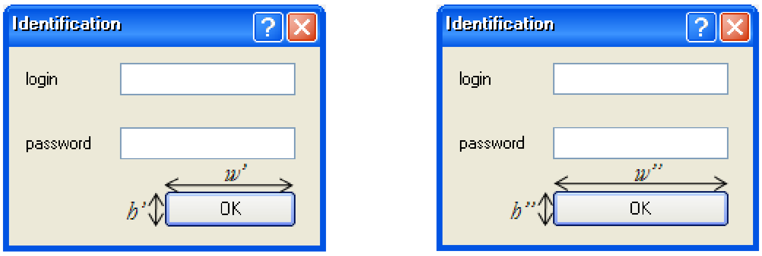
Adaptation Effect
Enlarge/reduce a *widget*
Move a *widget*
Delete a *widget*
Add a new *widget*
Replace a *widget*
Add a new container
Change the layout of the *widgets*/containers of a container
Enlarge/Reduce a container
Delete a container

Table 2. Adaptation effects considered for discontinuity.

Next, how the discontinuity in an adaptation is computed will be illustrated for one the items listed in Table 2.

Discontinuity when enlarging/reducing a widget

When an adaptation enlarges/reduces a *widget* discontinuity can appear because of changes in the *widget* size. Besides, discontinuity can also appear as a collateral effect produced by the elements included in the same container than the enlarged/reduced *widget*. These elements can be pushed by the enlarging/shrinking widget to accommodate themselves to the new available screen space.



a) original user interface b) adapted user interface

Fig. 2. Example of a simple adaptation where the size of a *widget* is changed.

In Figure 2 a simple example for an adaptation where a widget size changes is shown. In the left part of the figure the original presentation for the UI is illustrated. In the right part the adapted presentation is shown.

In equation 1 a formula to compute the discontinuity when the *widget* e_i changes its size is shown. Where *h'* is the height of e_i in the original UI, and *w'* is the width of e_i in the original UI. Similarly, *h''* and *w''* are the height and width of e_i in the adapted user interface, respectively. Finally, *h* and *w* are the current horizontal and vertical screen resolution of the device, respectively. The denominator in the formula (*h* x *w*) weights the discontinuity according to the screen resolution. This is important, since a *widget* size change of 25 pixels for a desktop PC can be nearly noticeable, while in a PDA with a much more reduced screen resolution it can be really important. Discontinuity is expressed in percentage. The right-most part of the formula represents the discontinuity produced as a collateral effect by the rest of the widgets on the same screen.

$$Discontinuity(e_i) = \left(\frac{h'' \times w'' - h' \times w'}{h \times w}\right) \times 100 + \sum_{j=1,\ j\neq i}^{n} Discontinuity(e_j) \qquad \textbf{(1)}$$

Cognitive Load Evaluation

When an adaptation is applied, a variation in the cognitive load derived from the amount of information shown can appear. Thus, to compute the benefit or damage in terms of cognitive load that an adaptation produces it is necessary the computation of cognitive load differential between the original user interface and the adapted one.

Adaptation Effect
Add *widgets*
Delete *widgets*
Add text
Delete text
Replace some *widgets* with other *widgets*

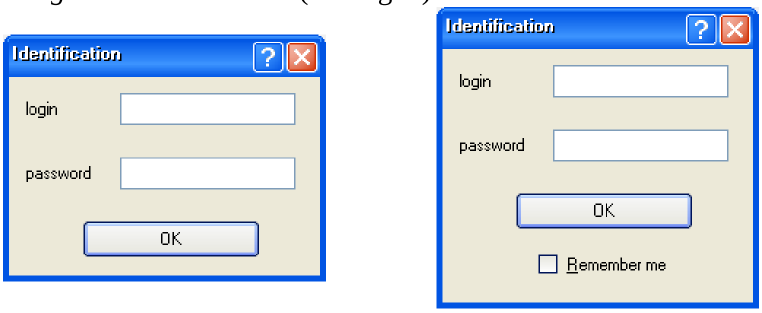
Table 3. Adaptation effects considered for cognitive load evaluation.

This differential is computed in terms of the increment or decrement of the information the user is asked to process in order to perform his tasks. Table 3 summarizes the adaptation effects considered when computing cognitive load differential in an adaptation.

Next, to illustrate how this computation is made, how cognitive load is computed for one of the effects shown in Table 3 will be described.

Cognitive Load Variation When Adding Widgets

When some *widgets* are added to the UI as a result of applying an adaptation, cognitive load increases, because the user needs to understand these new *widgets* that were added (see Fig. 3).



a) original user interface b) adapted user interface

Fig. 3. Example of an adaptation where a *widget* is added.

In equation 2 a formula to calculate the cognitive load differential when adding some *widgets* in the adapted user interface is shown. In this formula *perceive* is the time required to perceive an information unit (see Table 1), *Visibility* is a function that computes the visibility ratio of a *widget* (computed in terms of the screen space it requires to be shown) and *UnderstandWidget* is the time required to understand how a *widget* works. Therefore, to compute the cognitive load of a widget we take into account the time to perceive the *widget* elements, as long as the time to understand how the *widget* works. Thus, CC_k computes the cognitive load resulting from the inclusion of the *widget* CC_k in the adapted user interface. To compute the increment of cognitive load in the adapted user interface we need to calculate first the cognitive load for each new *widget* added (where *m* is the number of *widgets* added), because an adaptation can result in the addition of several *widgets*. When the computation of the cognitive load for *m widgets* is done, it is divided by the total cognitive load of *widgets* in the original user interface (*n*), to actually compute the increment of cognitive load.

$$CC_k = \frac{perceive \times visibility(w_k) + understandWidget}{2}$$
$$\Delta CC = \frac{\sum_{i=1}^{m} CC_i}{\sum_{j=1}^{n} CC_j} \qquad (2)$$

Adaptation Benefit Evaluation

Discontinuity and cognitive load increments represent the negative part of applying an adaptation for the user. Nevertheless, adaptation has also a positive facet that needs to be assessed. This positive facet comprises preferences, context frequency and user feedback.

Preferences represent the extent to which the adaptation meets the user preferences. Thus, this criterion is mostly assessed in terms of the data collected by data-mining the feedback provided by the user to discover hidden preferences, because the preferences in the description of the user profile stored in the user model are taking into account previously to determine if an adaptation should be fired or not.

When executing an adaptation it is necessary taking into account the feedback the user provided the last time the adaptation was applied. Thus, it is possible to consider how many times the adaptation was successfully applied (the user accepted it), and how many times the adaptation was rejected by the user. The inference of conclusions from this data can be either individual or collaborative. If it is individual, in the evaluation just the feedback from the current user will be considered, while if it is collaborative the feedback provided by other users will be also taken into account in order to adjust the migration cost for the given adaptation.

Context frequency also modifies migration cost. If an adaptation is appropriate for a context situation occurring often, the migration cost should be reduced, since the adaptation will be applied once but used many.

Finally, some adaptations are mandatory and they will be applied regardless of their migration cost. For instance, this is the case when the switching from a platform where there is audio playback support to a platform where there is not. Regardless of the migration cost all the audio elements of the user interface must be removed and replaced with equivalent elements for a non-auditory modality.

Conclusions and Future Work

In the path to provide meaningful adaptations to the user that actually improve the usability of the applications it is necessary to provide mechanisms to assess how good or bad an adaptation is, and to guarantee a certain *Quality of Adaptation* (QoA) to the user.

In this paper a mechanism is proposed that allows the evaluation of QoA in terms of migration cost and adaptation benefit concepts. The migration cost is computed by some expressions that take advantage of the basic interaction tasks times obtained empirically in GOMS-based studies.

Adaptation can be a powerful tool to improve user experience, but unless it is properly applied it can produce undesired effects that might lead users to reject or disable adaptation mechanisms. Therefore, adaptation quality mechanisms as the one proposed in this paper are required by adaptive applications in order to guarantee the quality of the adaptations applied.

As future work we plan to extend the concepts proposed for Virtual Environments where additional parameters in the migration cost should be considered. Further testing with real-world applications is also one of our inmediate goals.

Acknowledgments This work is partly supported by the PAI06-0093-8836 grant from the Junta de Comunidades de Castilla-La Mancha, Spain.

References

[1] Benyon D., Murray D.: Developing adaptive systems to fit individual aptitudes. Proceedings of the 1st international conference on Intelligent User Interfaces, pp. 115-121, Orlando, Florida, United States, ACM Press, 1993.
[2] Brusilovsky, P. Adaptive Web-based Systems: Technologies and Examples. The Twelfth International World Wide Web Conference, 20-24 May 2003, Budapest, Hungary.
[3] Calvary, G., Coutaz, J., Thevenin, D. A Unifying Reference Framework for the Development of Plastic User Interfaces. In Proceedings of IFIP WG2.7 (13.2) Working Conference EHCI'2001 (Toronto, May 2001), Springer Verlag ,LNCS 2254, pp.173-192.
[4] Card, S., Moran, T., Newell, A. The Psychology of Human Computer Interaction, Lawrence Erlbaum, Hillsdale, 1983.
[5] Dieterich, H., Malinowski, U., Kühme, T., Schneider-Hufschmidt, M.: State of the Art in Adaptive User Interfaces. Adaptive User Interfaces: Principle and Practice. Amsterdam (1993).
[6] López-Jaquero, V. Adaptive User Interfaces Based on Models and Software Agents, Ph.D. thesis, University of Castilla-La Mancha, Albacete, Spain, 2005. (in Spanish).
[7] López-Jaquero, V., Vanderdonckt, J., Montero, F., González, P.: Towards an Extended Model of User Interface Adaptation: the ISATINE framework, Proc. of Engineering Interactive Systems 2007. EHCI/DSVIS (Salamanca, 22-24 March 2007), Springer-Verlag, Berlin, 2007, to appear.
[8] Newell, A., Card, S.K. The prospects for psychological science in human-computer interaction. Human-Computer Interaction, 1, 209-242, 1985.
[9] Norman, D.A., Cognitive Engineering. In: Norman, D.A., Draper, S.W. (eds.): User Centered System Design. Lawrence Erlbaum Associates, Hillsdale (1986) 31–61.
[10] Olson, J.R., Olson, G. The Growth of Cognitive Modelling in HCI since GOMS. Human-Computer Interaction, 1990, Volume 5, pp. 221-265, Lawrence Erlbaum, 1990.