Int. J. Human-Computer Studies 77 (2015) 1–9

Contents lists available at ScienceDirect

Int. J. Human-Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

Adaptive support for user interface customization: a study in radiology

Wiard Jorritsma [a,*], Fokie Cnossen [b], Peter M.A. van Ooijen [c]

[a] University of Groningen, University Medical Center Groningen, Dept. Radiology, Hanzeplein 1, 9713 GZ, Groningen, The Netherlands
[b] University of Groningen, Institute of Artificial Intelligence and Cognitive Engineering, Nijenborgh 9, 9747 AG, Groningen, The Netherlands
[c] University of Groningen, University Medical Center Groningen, CMI-NEN, Dept. Radiology, Hanzeplein 1, 9713 GZ, Groningen, The Netherlands

ARTICLE INFO

Available online 9 January 2015

Keywords:
Human-computer interaction
Interface customization
Adaptable
Adaptive
Mixed-initiative
Picture archiving and communication system

ABSTRACT

Objectives: This study aimed to evaluate the usefulness of adaptive customization support in a natural work environment: the Picture Archiving and Communication System (PACS) in radiology.
Methods: Adaptive support was given in the form of customization suggestions, generated based on behavioral user data, which participants could choose to accept or ignore. Twelve radiologists worked with the standard adaptable version of the PACS for six weeks, during which their actions on the PACS interface were logged. Based on these logging data, customization support was generated for each specific participant. Half of the participants received support and half did not. After the support was given, logging continued for another six weeks. Participants' customization behavior and performance, measured as the average time needed to review a radiography study, were compared between the groups. Subjective responses to the customization support were also measured.
Results: Participants who received support used the PACS's customization facilities more effectively than participants who did not receive support. Participants receiving support accepted most of the customization suggestions and all participants indicated that the support was useful. We did not find an increase in efficiency due to the support, possibly because the performance measure we used was not sensitive enough. Subjective responses did show that participants perceived the support as increasing their efficiency.
Conclusion: Adaptive customization support would be a useful addition to the standard adaptable PACS interface, because it allows radiologists to customize their interface more effectively.

1. Introduction

Over the past decades, digital filmless radiology has largely replaced its traditional film-based counterpart. In the modern hospital, the distribution of patient images is controlled by a Picture Archiving and Communication System (PACS). The PACS also provides the user interface through which the radiologists retrieve, view and manipulate images. As in many other modern software packages, the number and complexity of functions in the PACS is very high and continues to increase. This poses the challenge of creating a user interface that presents these functions to radiologists in an appropriate way and allows them to interact with the software efficiently. Because different radiologists use the software in different ways, depending on their goals and interaction preferences, creating an interface that suits each radiologist is a difficult task.

As a solution to this problem, most PACSs have an *adaptable* interface, which allows radiologists to customize several aspects of the PACS (e.g. the items in the toolbar, the functions of the mouse, and the display protocols[1]) according to their personal needs and preferences. Research in the field of human-computer interaction has shown the potential benefits of adaptable interfaces compared to static ones (e.g. Findlater and McGrenere, 2004; McGrenere et al., 2002), but there are also problems with this approach. Users do not always customize effectively (Bunt et al., 2004), or they do not customize at all (Mackay, 1991). This means that they will never interact with the system in a maximally efficient way.

An alternative approach to interface personalization is the *adaptive* approach, where the interface changes automatically based on the user's behavior. This approach ensures that the interface is customized to the user, without requiring any effort on the user's part. However, automatically changing the interface can reduce the system's predictability and transparency, and can

* Corresponding author. Tel.: +31 50 3615720.
E-mail address: w.jorritsma01@umcg.nl (W. Jorritsma).

[1] A display protocol automatically arranges images on the screen in a way that is appropriate for the types of images that need to be displayed.

http://dx.doi.org/10.1016/j.ijhcs.2014.12.008
1071-5819/

undermine the user's sense of control over the system (Höök, 2000).

The *mixed-initiative* interface combines elements from both approaches, creating an adaptable interface with adaptive components that help the user customize effectively. The adaptive support is usually presented in the form of user-specific customization suggestions, which the user can choose to accept or dismiss. For example, the system could suggest to add or remove functions from a toolbar based on their frequency of use. Compared to a purely adaptive interface, a mixed initiative interface increases the system's predictability and transparency (the system does not change itself without the user's knowledge and consent), and increases the user's control over the system (users can dismiss the system's customization suggestions and can initiate their own customizations). Compared to a purely adaptable interface, a mixed-initiative interface increases the likelihood of customization (users may be more likely to customize because the adaptive support reduces the effort to customize, or because it simply reminds them of the possibility to customize). It can also increase the quality of customization, because the set of customization suggestions may contain useful customizations that users would not have considered on their own.

An example of a mixed initiative interface can be found in Debevc et al. (1996). This system provided adaptive customization support for Microsoft Word 6.0 by suggesting functions to be added to or removed from Word's toolbar based on their frequency of use. Both novice and expert Word users performed faster with this mixed-initiative interface than with the standard adaptable interface, although this effect was only found in a picture and text formatting task and not in a table editing task. The measure of task performance included the time users spent customizing the interface.

The adaptive support also reduced the average time needed to perform a customization. With support, users had to evoke the support mechanism and if they decided to accept a customization suggestion, it was performed automatically. This was faster than manually performing a customization, as done by users using the standard interface. Novices using the adaptable interface did not add any functions to the toolbar, while novices using the mixed-initiative interface did. Interestingly, experts in the mixed-initiative condition added fewer functions to the toolbar than experts in the adaptable condition. However, because a between-subject design was used, the possibility of a bias in customization behavior between the mixed-initiative and the control group cannot be excluded.

Bunt et al. (2007) implemented a similar mixed-initiative interface for Word 2003 and compared this interface with an adaptable one. They found that the mean task performance (both when excluding and including customization time) was faster in the mixed-initiative condition than in the adaptable condition, but this effect was not statistically significant (although the effect size was large), possibly due to the small number of participants used. Participants customized more effectively and efficiently when using the mixed-initiative interface. There was also a large subjective preference for the mixed-initiative interface.

Park and Han (2011) compared two mixed-initiative interfaces (adaptable with system support and adaptive with user control) for a PDA software prototype to an adaptable, adaptive and static alternative. Participants performed a controlled menu selection task (excluding customization time) slower and with more button presses on the static interface than on the others, but there were no differences between the other interfaces. The adaptable interface with system support did reduce the customization time compared to the standard adaptable interface. This system presented participants with their function usage frequencies, which allowed them to make faster decisions about which customization to perform. The perceived efficiency and overall preference was higher for the two mixed-initiative and the adaptive interfaces than for the adaptable and static ones.

These studies show that adding adaptive customization support to an adaptable interface has the potential to increase customization effectiveness and efficiency, interaction efficiency (although this is not strongly supported), and user satisfaction. However, all of these studies were conducted in laboratory conditions over short periods of time (the longest study lasted for 2.5 h). Users' customization behavior might be very different in real-world situations, where different (perceived) costs and benefits of customization apply and where users are not artificially manipulated to customize. Furthermore, little is known about the long-term effects of customization and how customization behavior changes over time.

The fact that our hospital had just purchased a new PACS presented us with the unique opportunity to study the effects of adaptive customization support on users working with a new interface in a real work environment. This study aimed to evaluate whether adaptive customization support would be a useful addition to an adaptable PACS interface.

1.1. Adaptable PACS interface

The graphical user interface of the PACS used in this study is shown in Fig. 1. At the top of the interface is a toolbar consisting of multiple tabs, each with a different set of functions (similar to the *Ribbon* in Microsoft Office 2007 and later). One of these tabs, called *My Tab*, is customizable; allowing users to add functions to it. A similar customizable region called *Right Click Menu* (*RCM*) is located at the top of the pop-up menu that appears when the right mouse button is clicked within the image. Users can add functions to these customizable regions by right-clicking on a function button and selecting *add to My Tab* or *add to RCM*. The default My Tab and RCM consist of eight pre-defined functions each. For most discussions in this article, the distinction between these two customizable regions is irrelevant. In these cases we will use the term *custom region* for both My Tab and RCM.

1.2. Adaptive customization support

The adaptive customization support was designed to help users customize the PACS's custom region effectively. It was based on users' function usage, which was logged by the PACS's built-in logging tool, and consisted of a table that gave insight into a user's function usage and a set of suggestions about which functions the user should add or remove to his or her custom region.

1.3. Research questions and hypotheses

In order to determine the usefulness of adaptive customization support for an adaptable PACS interface, the following questions were addressed:

(Q1) Does customization support increase the effectiveness with which users use the PACS's customization facilities?
(Q2) Does customization support increase the efficiency with which users interact with the PACS?
(Q3) To what extent do users accept the customization suggestions?
(Q4) Do users perceive customization support as being useful?

Effective customization (i.e. constructing a high-quality custom region) means trading off custom region coverage (the percentage of total function usage that the functions in the custom region cover) against custom region complexity (the number of functions