Consistency: a Factor which Links the Usability of Individual Interaction Components Together

Willem-Paul Brinkman
Brunel University
Uxbridge, Middlesex
UB8 3PH
United Kingdom
willem.brinkman@brunel.ac.uk

Reinder Haakma
Philips Research Laboratories Eindhoven
Prof. Holstlaan 4
5656 AA Eindhoven
The Netherlands
reinder.haakma@philips.com

Don G. Bouwhuis
Technische Universiteit Eindhoven
P.O. Box 513
5600 MB Eindhoven
The Netherlands
d.g.bouwhuis@tue.nl

ABSTRACT

An underlying assumption of component-based software engineering for interactive systems is that the overall usability of a new assembled device mainly depends on the usability of its individual components. This paper challenges this assumption by presenting findings of a series of lab experiments in which 48 subjects operated several consumer devices. The experiments focussed on the effect inconsistency may have on the usability of individual components. The results indicate that inconsistency could cause components, in the same or in higher layers, to activate an inappropriate mental model for other components. Furthermore, the application domain also seems to have an effect on the subjects' understanding of the functionality a component provides.

Keywords

Consistency, usability, component-based software engineering, usability testing, usability evaluation.

INTRODUCTION

Component-Based Software Engineering (CBSE) advocates the development of independent components, which can be used to create a new device. To do this, components should be autonomous units, free of the context in which they are deployed. This idea is one of the major success factors behind object-oriented development; it reduces the complexity of large software projects and improves the maintenance and reliability of a system (Cox, 1990). This approach is also used for the development of interactive systems. Interaction components such as input devices, or output devices are developed and tested in isolation to optimise usability. HCI theories such as the Layered Protocol Theory (LPT) (Farrell, Hollands, Taylor & Gamble, 1999) support CBSE. LPT describes how interactive systems can be broken down into individual components and claims that these components can be replaced by other components without affecting the remaining part of the system as long as components provide the same services. The underlying assumption is that using highly usable components will result in highly usable systems. However, others (Hertzum, 2000) suggest that software re-use can cause conceptual mismatches. The same concept may be used in several components, but it may not mean the exact same thing. We argue here that inconsistency can also cause components to affect each other's usability negatively, making an overall usability prediction of a system based on the usability of the individual components unreliable. This means that although a component can be developed and tested in isolation, a usability evaluation of the entire device is still required.

CONSISTENCY AND COMPONENTS

Consistency has no meaning on its own; it is inherently a relational concept (Kellogg, 1989) and can be described as doing similar things in similar ways with agreement between agents about which things are similar (Reisner, 1993). This means that a component is regarded as consistent when both designers and users partition the interaction with the component in the same way. Furthermore, designers and users have to apply the same criteria, or dimensions, to consider the interaction with components to be similar. Likewise, inconsistency involves disagreement between designers and users about which things are similar, since what designers may find consistent may not be consistent for users at all (Grudin, 1989).

In this paper we regard interaction components as elementary units of interactive systems, on which behaviour-based evaluation is possible (Brinkman, Haakma & Bouwhuis, 2004). An interaction component is a unit within an application that directly or indirectly receives signals from the user. These signals enable the user to change the physical state of the interaction component. Furthermore, the user must be able to perceive or to infer the state of the interaction component. Therefore, an interaction component should provide feedback. Without the possibility of perceiving the state, users cannot control it, making their behaviour aimless. The points where input and output of different interaction components are connected demarcate the border between layers. An interaction component operates on a higher-level layer than another interaction component, when the higher-level interaction component receives its user's messages from the other interaction component.

Consistency in a component-based environment is related to the feedback a component provides, and especially the feedback that guides users in their action selection. When this kind of feedback fits into the users' mental model, users can derive the consequence of an action from this mental model. The system feedback is also responsible for the users' activation of a mental model. However, if something else besides the component's feedback were to determine what mental model users apply, the usability of a component would be partially outside the control of its designer, which would undermine the component's autonomy.

Several studies have shown that consistency can affect the overall usability of a device (e.g. Payne & Green, 1989; Polson, 1988). However, little has been said about whether consistency can cause components to affect each other's usability. This study looks at three situations where this may occur: between components in the same layer; between components in different layers; and between a component and an application domain. All situations concern users' misinterpretation of the feedback because of the mental model they apply. The reason why users apply a particular mental model may depend on factors outside the component, such as feedback of other components or the application domain.

Before describing the three experiments that studied these three situations, the general experimental set-up of the experiments is presented. After the presentation of the experiments, the findings are discussed in general.

GENERAL EXPERIMENTAL SET-UP

All three experiments were conducted simultaneously under the control of one PC application written in Delphi 5. All 48 subjects, students of Eindhoven University of Technology, participated in all three experiments and received NLG 15 (roughly € 7) for their effort. The experimental design was counterbalanced for possible two-way interaction effects between the experimental conditions of the three experiments and the order in which they were asked to use the devices. Throughout the task performance, the message exchange between the interaction components of the devices was recorded. This made it possible to count the number of messages a component received. This has been shown to be a powerful objective component-specific measure, as it presents the amount of effort a user has made to control a component (Brinkman, Haakma, & Bouwhuis, 2001).

INCONSISTENCY WITHIN THE SAME LAYER

The experiment to study the effect of inconsistency between interaction components within the same layer was conducted with four simulations of a room thermostat.

Room Thermostat

The room thermostat had two very similar interaction components —daytime and night-time temperature— which users presumably expected to be more or less similar things and therefore could be operated in a similar manner. Two similar versions of both components were designed, which resulted in four simulations. In one version the temperature had a display with a moving pointer (Figure 1, left image, upper display) and a fixed scale, in the other version the display had a fixed marker and a moving scale (Figure 1 left image, lower display). The Left and the Right button had an opposite effect in the two versions.

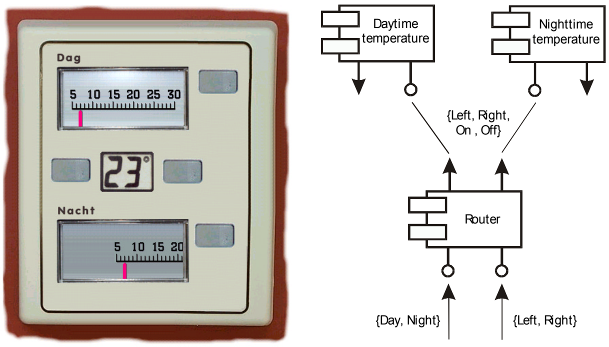

Figure 1: (left) an inconsistent room thermostat, and (right) part of the compositional structure of the device.

Results

An ANOVA was conducted on the number of messages received by the Nighttime interaction component. The analysis took the versions of the Daytime Temperature interaction component (2) and the version of the Nighttime Temperature interaction component (2) as between-subjects variables. The results revealed a significant main effect ($F(1,44) = 9.22$; $p. = 0.004$) for the version of the Nighttime component. More messages were received when the component was implemented with the moving scale version. In addition, the analysis found a significant two-way interaction effect ($F(1,44) = 7.06$; $p. = 0.011$) between the Daytime and Nightime versions. More messages were received in the prototype that had the moving pointer version for the daytime temperature and the moving scale for the night-time temperature (Figure 1) than in the other three prototypes. The explanation for this interaction effect is that when subjects started with setting the daytime temperature, implemented with a moving pointer, they activated a more familiar mental model than that associated with the moving scale version. The Nighttime Temperature component was interpreted in the light of this powerful mental model, which did not fit with a moving scale implementation and made subjects click on the wrong buttons.

INCONSISTENCY BETWEEN LAYERS

The experiment to study the effect of inconsistency between interaction components in different layers was conducted with four simulations of a web-enabled TV set. A mistake that novice Lynx users probably easily make, served as a model for a possible inconsistency problem between two layers. Lynx is a text-based web browser that allows users to access the web in non-graphical environments without the use of a mouse.

Users can select the links with the Up and Down arrow buttons on the keyboard. To activate the selected link, users have to press the Right arrow. With the Left arrow, users can return to the previous page. The possibility of an error may increase when links in the web page are placed on the same line. The supposed error occurs because to the activation of an inappropriate mental model —horizontal positioning with the Left and Right arrows.

Figure 2: (left) linear-oriented remote control; (middle) plane-oriented remote control; (right upper corner) matrix layout; (right lower corner) list layout.

Web-Enabled TV Set

The tasks the subjects had to perform, using a web-enabled TV set (Figure 2), was to find the web page that gave the departure times of a bus based on the bus stop, the bus number, the city and the province, which were all given in the instructions. The experiment had a 2 (web pages) × 2 (browser) between-subjects design. Variations in the web page's layout led to two versions of the Web Pages component. One layout, the *matrix* layout, placed the web links in a web page both on the same line and one below the other. The other layout, the *list* layout, placed all links one below the other. Variations in the remote control led to two versions of the Browser component. For one remote control, the *linear*-oriented version, the Up and Down buttons were interpreted as, "select the previous link" or "select the next link in succession". The sequence went from left to right and continued on the left of the next line. The Left and Right button were interpreted as "jumping to the previous web page" and "activate the selected link". For the other remote control, the *plane*-oriented version, the Up and Down buttons were interpreted as "select the link above" and "select the link below". Consequently, the Left and Right buttons were interpreted as "select the link left" or "select the link right". The subject could jump to the previous page with the Back button and activate the selected link with the Middle button.

Results

The minimal number of messages received by the web page server required to perform the tasks were different in the four prototypes. Therefore, instead of analysing the absolute number, the number of messages received that were needed in addition to the minimal numbers were analysed by subtracting the minimal numbers from the observed ones. The ANOVA took the versions of the Browser (linear or plane oriented) and the Web Page (matrix or list layout) as between-subjects variables. The results showed a significant main effect ($F(1,44) = 24.22$; $p. < 0.001$) for the version of the Browser and a significant main effect ($F(1,44) = 15.62$; $p. < 0.001$) for the version of the Web Page. The web server received more messages when a prototype was equipped with the linear-oriented instead of the plan-oriented version of the browser, and when a prototype was equipped with the matrix instead of the list version of the web pages.

The analysis also revealed a significant interaction effect ($F(1,44) = 16.82$; $p. < 0.001$) between the Browser version and the Web Page version. The web server received more messages in the prototype that combined the linear-oriented Browser version and the matrix Web Pages version than in the other prototypes. This demonstrates that even though the Internet architecture is developed to make web pages independent from the browsers, users might run into trouble when on a higher-level layer the web-page server activates an inappropriate mental model for the interpretation of lower-level browser's feedback.

INCONSISTENCY AND APPLICATION DOMAINS

The last experiment studied the effect the application domain may have by activating a general mental model, which in turn may activate a component-specific mental model, which users apply to interact with a component. Note the difference with the previous experiments. In this experiment it is not the feedback of other components, but the users' idea of operating a particular device that determines what component-specific mental model they apply.

Radio Alarm Clock and Microwave

The experiment took a radio alarm clock and a microwave as applications in which two versions of a clock were implemented. In the radio alarm clock, the clock determined when the radio should be switched on, and in the microwave, the clock determined when the cooking should start. The fit or misfit between the application domain and the clock was in the clock's feedback that was presented along with the timer time (Figure 3). In one version, the mechanical alarm version, the symbol of a ringing mechanical alarm clock was shown, in the other version, the hot dish version, a symbol of a hot dish. The clock had four different modes: displaying the current time, displaying the timer time, setting the current time, and setting the timer time. The current time was presented along with a symbol of a clock (Figure 3, right symbol). The timer time was presented along with the ringing mechanical alarm clock or the hot dish.

Figure 3: (left) ringing mechanical alarm clock, (middle) a hot dish, and (right) normal clock symbol.

When the subjects performed a task with the radio clock, the expectation was that the task of setting the alarm of an alarm clock would activate a general mental model on alarm clocks, which subsequently activates a component-specific mental model of setting the alarm of alarm clocks. In light of this activated component-specific mental model, subjects could more easily understand the feedback "the time the timer will go off" presented by the mechanical alarm clock and by a hot dish. The opposite was expected for the microwave, where the feedback indicates "the time cooking begins" which is probably better presented by the hot dish than by a mechanical alarm clock.

Results

An ANOVA was conducted on the number of mode change requests received by the clock. The analysis took the Clock version (2) and Application domain (2) as between-subjects variables. The analysis did not find a significant interaction effect ($F(1,44) = 0.02$; $p. = 0.887$) between two independent variables. Besides the straightforward interpretation that there is no general mental model that indirectly influences the interaction with a specific component, another interpretation is an unanticipated effect of the experimental set-up. Although the subjects may not have understood the inconsistent symbol presented with the timer, it was the only option.

The ANOVA did however reveal a significant main effect ($F(1,44) = 7.57$; $p. = 0.009$) for the application domain. Subjects less often changed the clock mode when they operated the radio alarm than when they operated the microwave. The same clock function was apparently easier to use in one application domain than in the other, which suggests that the usability of a components depends on the application domain.

DISCUSSION

The results of the first two experiments show that the control of a component can depend on other components. Feedback is interpreted with a component-specific mental model, which feedback of other components may have activated. The third experiment shows that the application domain may also have an impact on the usability of components. These findings may be limited to the phase where users learn to control a component, as was the case in all three experiments. Once users gain experience with controlling the components it might be that the dependence between them lessens because the correct components-specific model will be activated. Users might be more guided by the feedback initially, and later on more by their own experience.

The findings demonstrate that designers should not assume that selecting components that may be very easy to use in other applications would automatically result in a very easy to use new application. When designers are creating a new component they should try to predict what other components will be used in relation with their component. If this is not possible, the component should be designed according to a set of specific rules. Later on, when the component is used to build an application, developers should make sure that the components they apply follow the same rules, or at least that there are no conflicting rules. These rules can be laid down in a style guide. However, this does not guarantee an application without inconsistency, because users do not have to agree with what designers consider to be consistent. Only the involvement of users can solve this problem.

REFERENCES

Brinkman, W.-P., Haakma, R., & Bouwhuis, D.G. (2001). Usability evaluation of component-based user interfaces. *INTERACT'01*, Amsterdam: IOS Press, 767-768.

Brinkman, W.-P., Haakma, R., & Bouwhuis, D.G. (2004). Usability testing of interaction components: Taking the message exchange as a measure of usability, In R.J.K. Jacob, Q. Limbourg & J. Vanderdonckt (Eds.), *Pre-Proceedings of CADUI'2004* (p. 159-170). Dordrecht, The Netherlands: Kluwer Academics.

Cox, B.J. (1990). There is a silver bullet: A software industrial revolution based on reusable and interchangeable parts will alter the software universe. *Byte, 15*, 10, 209-218.

Farrell, P.S.E., Hollands, J.G., Taylor, M.M., and Gamble, H.D. (1999). Perceptual control and layered protocols in interface design: I. Fundamental concepts. *International Journal of Human-Computer Studies, 50*, 489-520.

Grudin, J. (1989). The case against user interface consistency. *Communications of the ACM, 32*, 1164-1173.

Hertzum, M. (2000). Component-based design may degrade system usability: Consequences of software reuse. *Proc. OZCHI 2000*, Sydney: Ergonomics Society of Australia, 88-94.

Kellogg, W.A. (1989). The dimensions of consistency. In J. Nielsen (Ed.), *Coordinating user interfaces for consistency* (p. 9-20). London: Academic Press.

Payne, S.J. & Green, T.R.G. (1989). The structure of command languages: An experiment on task-action grammar. *International Journal of Man-Machine Studies, 30*, 213-234.

Polson, P.G. (1988). The consequences of consistent and inconsistent user interfaces. In R. Guindon (Ed.), *Cognitive science and its applications for human-computer interaction* (p. 59-108). Hillsdale, NJ: Lawrence Erlbaum.

Reisner, P. (1993). APT: A description of user interface inconsistency. *International Journal of Man-Machine Studies, 39*, 215-236.