

Evaluating Interface Design Choices on WAP Phones: Single-choice List Selection and Navigation among Cards

LUCA CHITTARO AND PAOLO DAL CIN

*Department of Mathematics and Computer Science
University of Udine
via delle Scienze 206, 33100 Udine, ITALY
+39 0432 558450
chittaro@dimi.uniud.it*

1 Introduction

WAP phones are a growing relevant part of the mobile market, and the number of offered WAP services is increasing at a very rapid rate. Usability is a crucial requirement for this kind of services, which must be easily operated by users on small screens and keyboards. Unfortunately, very little has been published on the evaluation of WAP devices and services on users, and the main focus of the few available papers (e.g., [2,5]) is not on the rigorous experimental evaluations demanded by HCI research. For these reasons, we chose to make user evaluation the main focus of our research on WAP interfaces. In particular, this paper presents a user study that evaluates two of the possible user interface design choices which have not been thoroughly investigated neither in the literature nor in design practice. Besides the specific results obtained, one of the contributions of this paper is also to present a case study of a carefully thought experimental procedure for evaluating different design choices for a WAP service.

2 The Considered Design Alternatives

Current guidelines (e.g., [4]) on WAP service design tend to provide general principles that build on previous HCI knowledge (e.g., “always define a card for confirmation of potentially dangerous actions”) or known technical limitations of WAP devices (e.g., “limit the length of labels to a maximum of 5 characters, otherwise some telephones will truncate them”). This kind of guidelines can be formulated without the need of experimental backing, but leave many questions unanswered, e.g. about making design choices among the specific interaction styles supported by WML (the language for content formatting in the WAP standard). Answering these questions requires a thorough, complex, and time consuming experimental activity. In this paper, we consider two of these questions. The first concerns *navigation among cards* (NAC, hereinafter), i.e. the

possible ways users can follow to move from one card to another in a WAP service. WML offers a choice between two implementations: (i) *Links* (exploiting the `<anchor>` tag): the user sees the possible destinations as underlined links in the current card, and clicks on them to move to the chosen card (see Fig. 1 for an example), and (ii) *Action screen* (`<do>` tag): the user clicks on the “Options” key on the keyboard and sees the possible actions that can be taken in the current card listed on the full screen (see Fig. 2). The second question concerns *single-choice lists* (SCSs, hereinafter), i.e. lists displaying a number of alternatives of which only one can be chosen. WML offers two possible implementations: (i) *List of links* (exploiting the `<anchor>` tag): the user sees the list of alternatives displayed as underlined links and moves up/down to reach the desired value and click on it (see example in Fig. 3), and (ii) *Selection screen* (`<select>` tag): the user sees the currently selected value between brackets, (s)he can accept it or click on it to see the possible alternatives listed on the full screen and select another value (see Fig. 4).

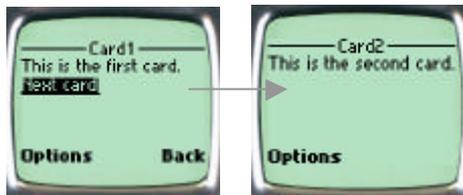


Fig. 1 NAC implemented as a *Link*.

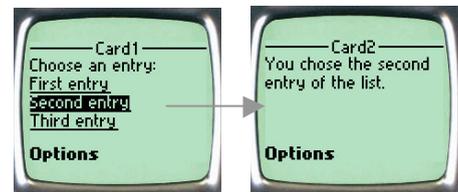


Fig. 3 SCS implemented as a *List of links*.



Fig. 2 NAC implemented as an *Action screen*.

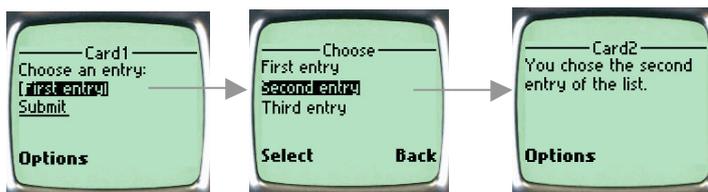


Fig. 4 SCS implemented as a *Selection screen*.

Implementations in Fig. 1 (*Link*) and 3 (*List of links*) look much simpler with the displayed trivial example, but this is not necessarily the case with more complex cards of real WAP sites. As an example, when a card contains more than one SCS and there are many values to choose from in each list, *Selection screen* choice looks simpler than *List of links* because it allows the user to see the choices to be made and the selected values in a form-like screen that can be examined without scrolling too much, while *List of links* expands every list, and can require much more effort to reach the desired values. Our focus on the two above described WAP interface elements is motivated by a number of reasons: (i) they are crucial in the development of any WAP site, and can deeply affect

its usability, (ii) existing guidelines for WAP interface design (such as [4]) do not give criteria to choose among the described possibilities, (iii) frequently cited WAP references (such as [1]) stress the fact that research is needed on providing guidelines for them, (iv) choosing the best implementation for these elements has been and is a topic of debate in mailing lists of WAP developers such as [6,7], witnessing the lack of design guidance and consensus about these aspects and the practical importance of studying the issue.

3 Experimental Evaluation

In the literature, the few user studies that concern WAP sites are performed using WAP device emulators (such as the Nokia or the Phone.com emulators) on a normal PC. We do not believe such a kind of testing can provide reliable indications of user performance for several reasons: (i) the user experience is sensorially too different from the real use of the device (both from a visual and haptic point of view), (ii) the employed peripherals are too different both for the input and the output, (iii) the interaction styles are too different (pointer-mediated interaction with virtual controls vs. direct interaction with physical controls). Therefore, our experimental activity was carried out using a real WAP phone. As explicitly indicated by research performed by mobile phones manufacturers [3], user efficiency and overall usability are critical design attributes in the mobile field. We thus devoted particular attention to their evaluation.

3.1 Subjects, Experimental Setting and Task

A total of 40 subjects (equally split between males and females) was involved in the experiment. The subject population was chosen to be representative of a general public, and to be diverse (in age, occupation, and ability with computers). With respect to WAP phone experience, none of the subjects had used a WAP phone before.

The experiment was carried out using a Nokia 7110, connected wirelessly to a full WAP service. This allowed us to take the experimental activity out of our usual research lab, and move it into a common home environment that made subjects more at ease. The WAP service allowed the user to get information about the movies played in a city's cinemas and make reservations. We developed the system by following current general usability guidelines for WAP services (e.g., for the avoidance of browser-specific or phone-specific features). The system is particularly suited for testing different versions of the considered elements, because it contains a large number of SCSs and a deep navigation structure. The experimental task had to: (i) be representative of frequently occurring tasks in real WAP sites, and (ii) heavily involve the interface elements under evaluation. Therefore, we chose a search and selection task where the user had to find a pair of movies satisfying some specific requirements and reserve seats for the proper

show time. Fig. 5 shows the employed mobile phone, and the purpose of its different controls in the considered task. Before carrying out the task, subjects were individually instructed about the functionalities of the WAP phone and the WAP service.

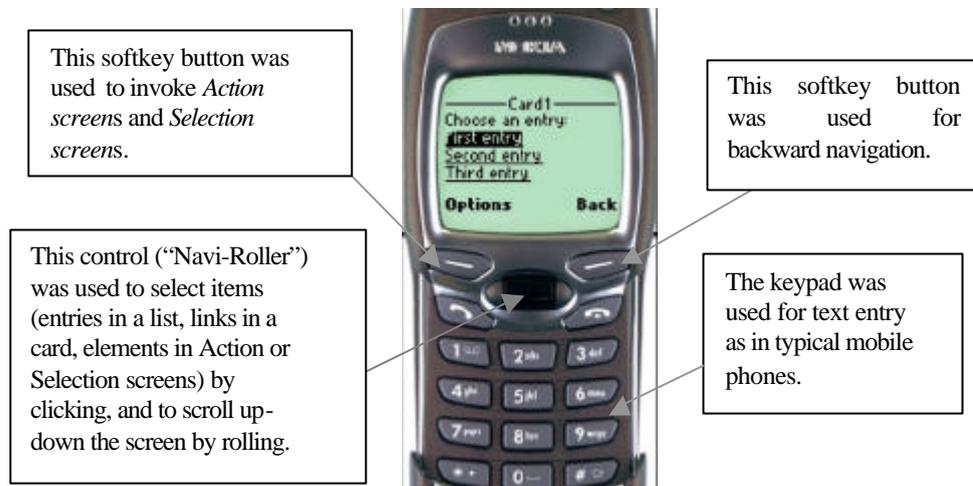


Fig. 5. Purpose of the different controls in the employed mobile phone.

3.2 Variables, Experiment Design, and Hypotheses

Two independent variables are involved in the study: (i) implementation of NAC (with two levels: *Links* or *Action screen*), and (ii) implementation of SCSs (with two levels: *List of links* or *Selection screen*). The following dependent variables have been measured to characterize user efficiency and usability.

Percentage of task completed in a given time was measured as the percentage of correctly visited cards (determined through software logging) among the set of cards necessary to complete the task. The time interval was chosen with a pilot study on a few subjects that determined an average time to complete the task equal to 9 minutes.

Time needed to complete the task was recorded (for subjects who were not able to complete the task, time needed coincides with the 9 minutes maximum allowed time). Since time spent can be influenced by variations in the connection speed, we used a video recording to precisely eliminate connection times (which anyway did not show large variations). The camera was placed in such a way that the subject clearly felt that his/her face was not recorded. To make subjects further at ease, they were told that the purpose of video recording was to evaluate technical performance of the phone display (so, implicitly, that it was not meant to evaluate them).

Number of clicks needed to complete the task was also measured.

To give an indication of how much the interface facilitated the user in trying to reach the required state, we recorded the *Number of deviations* users took with respect to the shortest path of cards from the initial state to the state required by the task. A deviation was detected when the user visited a card (and possibly a number of following

unnecessary cards) that was not prescribed by the shortest path of cards.

Total number of cards visited in the WAP site integrates the previous information, and was easily counted by software logging code.

Finally, users' *Subjective evaluation* of the interfaces was acquired with a questionnaire. A 2 x 2 factorial design was adopted. Subjects have been assigned to the four different conditions following a mixed design. First, according to a between-subjects design, subjects have been split into two homogeneous groups of 20 subjects each, and the first group was assigned only to the conditions employing *list of links* for SCSs, while the second only to the conditions employing a *selection screen* for SCSs. Then, in each group, every subject performed the task in the two possible conditions for NAC according to a within-subjects design. Therefore, two different versions of the task were needed and we designed them in such a way that they were of the same complexity and used the considered interface element in the same proportion. The assignment of subjects to version of the task and conditions in the within-subjects part of the design was carried out following a counterbalancing scheme where the order for the assigned version of the task and the order for the assigned condition were varied independently.

Our hypotheses were that: (i) *Links* provide better user performance than *Action screen* for NAC, and (ii) *List of links* provide better user performance than *Selection screen* for SCSs. Hypothesizing these effects for the *number of clicks* variable is trivial, because navigating with *Action screens* and making choices with *Selection screens* objectively requires more clicks (to invoke the corresponding screens). For the other dependent variables, we were instead motivated by the consideration that when an *Action screen* or *Selection screen* is displayed, the user temporarily loses sight and access to the contextual information provided in the card (s)he was examining. Therefore, we believe (s)he can become more easily disoriented (and require more mental effort for orientation) in navigating the WAP site with respect to *Links* and *List of links*.

3.7 Statistical Analysis and Experimental Results

A two-factor mixed-design analysis of variance (ANOVA) has been performed with implementation of NAC as the within-subjects variable, and implementation of SCSs as the between-subjects variable. The results indicate that the effect is significant: for each dependent variable, a value of $p < 0.01$ has been obtained. Means are given in Tables 1 and 2. All values in the two tables show that user performance was better in the *Links* condition for NAC, and in the *List of links* condition for SCSs. The results in the percentage of completed task (both close to 100%) reflect the fact that the time interval we gave was sufficiently wide, and so most of the subjects were able to complete the task. The large difference in the number of clicks is no surprise, as previously discussed. The

findings we consider to be most interesting are those concerning the number of deviations and total number of visited cards. As shown by the two Tables, subjects in the *Action screen* and in the *Selection screen* conditions are more likely to deviate from the shortest path of cards that leads to task completion and to visit more cards than needed. This is consistent with our conjecture that having to invoke a separate screen (for navigation or selection purposes) while using a card is detrimental to user orientation capabilities. Subjective ratings of difficulty were mapped from the qualitative scale of the questionnaires into integers. Table 3 shows the mapping and the obtained means. Results concerning NAC were analyzed using the Wilcoxon test (dependent samples), while results concerning SCSs were analyzed using the Mann-Whitney test (independent samples). In both cases, a value of $p < 0.01$ has been obtained. Subjective ratings confirm the results provided by objective measurements, clearly showing that users perceive a greater degree of difficulty in the *Action screen* and *Selection screen* conditions.

Table 1. Navigation among cards (NAC): means.

	Links	Action screen
Time needed	469 sec.	513 sec.
Percentage of task	97.4%	96.1%
Number of clicks	61.30	129.50
Number of deviations	0.75	1.85
Total number of cards	34.10	38.00

Table 2. Single-choice lists (SCSs): means.

	List of links	Selection screen
Time needed	467 sec.	514 sec.
Percentage of task	97,7%	95,8%
Number of clicks	79.10	111.60
Number of deviations	0.77	1.90
Total number of cards	33.90	38.20

Table 3. Means of subjective ratings obtained with questionnaires.

Navigation Among Cards		Single-choice Lists	
Links	Action screen	List of links	Selection screen
4.07	2.70	3.65	2.75

(5=Very Easy, 4=Easy, 3=Normal, 2=Difficult, 1=Very Difficult)

4 Final Discussion

In closing, we would like to add a word of caution about the generalization of the obtained results to different situations. First, although our experiment was based on a representative example of current mobile phones, the market is likely to see the introduction of radically new designs with very different I/O modalities, that could deeply affect user interaction. From this point of view, the study we presented should be repeated with representative examples of other classes of WAP-enabled devices that present a structure and I/O peripherals which are very different from the considered one.

Second, we would like to stress the fact that we studied the behavior of *novice* users of WAP services, which could differ from that of *expert* users. The choice of focusing on novice users was motivated by the fact that WAP is a recently introduced technology for which attracting new users is a crucial priority, and experts users are currently very few. However, studying expert users will become an important topic, especially when WAP sites will feel the need of providing the two categories of users with different kinds of interfaces.

REFERENCES

1. Arehart C., Chidambaram N., Guruprasad S., et al.: Professional Wap, Wrox Press, Birmingham (2000)
2. Kaasinen E., Aaltonen M., Kolari J., Melakoski S., Laakko T.: Two Approaches to Bringing Internet Services to WAP Devices. Proc. 9th Internat. WWW Conf., Computer Networks Journal **33** (2000) 231-246
3. Jokela T., Pirkola J.: Using Quantitative Usability Goals: A Case Study about Development of a User Interface for a Cellular Phone. Proc. INTERACT 97. Chapman and Hall, London (1997)
4. Openwave Systems.: GSM Application Style Guide. <http://www.phone.com/pub/gsm900-1800.pdf> (2000)
5. Schmidt A., Schroder H., Frick O.: WAP – Designing for small user interfaces. Proc. CHI2000 Conf. Human Factors in Computing Systems, Abstracts Volume. ACM Press, New York (2000) 187-188
6. *wap-dev*, mailing list at wapwarp.com/wap-dev
7. *WML and WMLScript Programmers List*, mailing list at www.egroups.com