User Driven Adaptation: An HCI approach to Adaptation

Jim Cai and Shizhong Li
University of Toronto
{jcai,sli}@cs.toronto.edu

Abstract

Mobile devices have been widely used these days to provide web access any time anywhere. However, its limited bandwidth, screen size and power resources all put constraints on its usability. To solve this problem, content are typically adapted to work with these constraints. However, existing automatic adaptation systems often fail to adapt according to user semantics. In this paper, we present the first stage research of our user-driven adaptation system, which we believe will best adapt content to serve specific user interests. Our controlled experiment evaluates two interaction interfaces that can be used in such adaptation system. The result of the experiment shows that if the interface is designed appropriately, our approach can achieve maximum flexibility of adaptation with little user interaction overhead.

1. Introduction

Mobile devices, such as cell phones and network-enabled personal digital assistant (PDAs) are fast becoming the platform of choice for accessing web content, and are expected to play a significant role in ubiquitous computing [1]. However, traditional web content are designed for powerful desktop computers and needs to be adapted to these mobile devices in order to meet their bandwidth, power and screen size constraints. Most of the current research these days focuses on adapting content on-the-fly [2] [3] where a set of rules and policies are specified to adapt content of specific sizes to various devices. For instance, a simple bandwidth adaptation rule could be to always adapt object content to a cell phone device at 30% of its original file size if the file is more than 500KB. We usually refer to the class of adaptation schemes that rely on such rules or constraints as automatic adaptations [4].

Automatic adaptation seems promising in transferring web experiences from desktop computer to mobile devices. It is also a scalable solution as content is trans-coded on-the-fly without many human interventions. However, this approach does not always produce the optimal adaptation results that users would want to see. In fact, a common critique to all automatic adaptation techniques these days is the fact that they all lack of considerations for user semantics. Please consider the following scenarios:

User A, a handheld computer user, tries to read news off the CNN website. He is interested in only the key story of the day with its high resolution images. User B is on a business trip, he would like to view a video clip that is very critical to his business at the highest possible quality level, regardless of the fact that he is using a cell phone which charges him at a ridiculous downloading rate.

There are two key characteristics here. The first one is known as content semantic. Typically, content that are more relevant to the task that the user is trying to accomplish will likely to receive more attention from the user. Consequently, users will likely to request for higher quality levels for such content. For example, user A requires high resolution images for the top story, but may only require very low quality or even thumbnail versions of other images which he is not particularly interested in.

The second characteristic is irrationality. In some occasions, the contents are so critical that users demand more details than usually anticipated. For example, despite of the fact that user B is facing a big bill, he still wants high resolution videos. It is obviously impossible for rule based automatic adaptation schemes to capture either of these user characteristics.

1.1 User-driven adaptation

To address content semantic and user irrationality, we present user-driven adaptation, an approach that allows complete user control over the adaptation process. Specifically, the system always serves the content object at the lowest possible quality as default. Individual users are then allowed to interact with the system to adjust the quality of the adaptation. User-driven adaptation has considerable advantages over the traditional automatic adaptation schemes. First of all, this adaptation scheme is more flexible and user friendly. Since the users have total control over the adaptation, they can adjust the adaptation to meet their requirement for any particular task. Secondly, since the users are in control of the adaptation, their adaptation decisions can accurately reflect resource limitation of the devices and therefore maximizes the use of bandwidth, battery and screen size. None of the sophisticated rules and policies implemented in automatic adaptation systems can achieve such accurate judgments. Thirdly, if automatic adaptation is in use, the user driven adaptation can still be used as a platform for altering the bad adaptation decisions made by automatic schemes.

The only drawback of such approach is that it introduces more frequent interactions between users and the systems. Therefore, our research goal is to seek the best possible interface and interaction tools for such human-computer interaction, such that it minimizes the overhead introduced by adjusting content quality manually. As we can see, the system problem of adapting content now shifts to an HCI problem of how we can improve the usability of the adaptation interface.

There are two ways we can tackle this design problem, namely from the input side of the interface and the output side. At the input side, we have to decide what should be a suitable interaction technique to provide user refinement to the adaptation. We also have to decide the variations of such technique when applied to different devices and object types. Finally, we have to gauge how users interact with multiple objects simultaneously in an efficient way. At the output side, the system needs to help and guide the users to adapt content to the degree that they prefer. This includes informing the user of possible refinements available, informing the user the cost of doing each refinement, and recommending to the user an optimal adaptation predicted based on automatic adaptation policies, such as Community Driven adaptation [1].

The rest of the paper is structured as follows. In section 2 we will briefly review several research works that is highly related to user driven adaptation. In section 3 we will discuss our first stage research, which concentrates on evaluating interaction technique for our system. In section 4, we present a controlled experiment that illustrates why user interface design matters for our system and what are the tradeoffs between two particular interaction interfaces: mouse click and mouse wheel scroll. Finally, we conclude our work in section 5 with the discussions of our planned future work.

2. Related Work

To the best of our knowledge, this is the first work that aims at building a comprehensive user interface for adaptation systems. However, there is significant past research that contributes to content adaptation systems.

The most typical content adaptations are ruled-based [5][6] and constraints based [7]. In rule-based system, a set of mathematical formulas are used to guide the adaptation process. For instance, an adaptation rule for image objects could be to adapt content to 30% of its original size if the original size is between 200 and 500k. Constraints based system also captures the resource availability and tries to maximize the usability, given available resources. However, it is impossible to set up separate rules or constraints for every data objects. As a result, only a small set of rules can be established and applied to all sets of content, regardless of the content type, who the user is, and how relevant the content is to the current task. Hence rules and constraints based systems are far from ideal for content adaptation.

Perhaps the closet work to the idea of user driven adaptation is the community driven adaptation (CDA) system [1]. In fact, user driven approach is initiated and based on CDA. CDA assumes that users who share the same characteristics (e.g., device type, preference) can be grouped into the same community, such that knowing how one user in the community prefers the content to be adapted will likely to be transferable to all other users in the same community. For this reason, CDA allows the first several users to refine the adaptation results, and tries to use these feedbacks to infer future accesses to the same content. The biggest challenge for CDA is how to classify communities and identify individual users into such communities. The predictions based on wrong community groupings can results in worse adaptation results than simple ruled based systems. Besides, user preferences are sometimes confidential information that is subject to security and privacy issues. On the other hand, the user driven approach never attempt to predict how users will adapt, which means the adaptation process is completely controlled by individual users, and always accurately reflect the requirements of individuals. In addition, no personal preference will be collected as no communities are required to be built.

3. First stage progress

In this first stage of our work, we primarily concentrate on evaluating the impact of interactions techniques to user driven adaptation. In particular, we select two basic interaction techniques, the mouse click, and the mouse wheel scrolling, to evaluate and compare.

Clicking

The clicking interaction is perhaps the simplest and most natural interaction technique to everyday internet users. In our adaptation system, one mouse (left) click over a particular image will request on behalf of the user one quality level improvement from the proxy for that image.

Scrolling

Mouse wheel scrolling is slightly more complicated than clicking. However, we frequently use this operation when browsing web content or fill HTML forms. In our adaptation system, we allow the users to use mouse wheel scrolling to select an intended fidelity level before the request is actually sent to the proxy. To do this, user first points his mouse at the image. As he/she scrolls, the number that indicates fidelity level on the screen will change accordingly. Once the user decides on the intended fidelity level, he has to hold down the mouse wheel to send the request.

To keep our evaluation model simple, we test our interface with one fixed object type and one layout mode. The object type we use is progressive JPEG, and the web layout mode we use is single image web page. To best capture the characteristics of both interaction techniques, we performed a controlled experiment which is described in detail in section 4.

4. Experiment

In this section, we will first present our physical experimental setup. It will be followed by the experiment methodology and procedure. At the end, we will present our experiment results and summarize our findings.

4.1 Setup

Our system consists of three components: two customized web browser, a web server and a transcoding proxy. We modified the web browsers by extending the Internet Explore 6 with Jscript to support fidelity improvements by means of mouse left click and wheel scrolling respectively. The proxy converts images embedded in HTML documents obtained from the web server into a progressive JPEG image. It then divides it up according to scans based on JPEG standard compression [9] proposed by the ISO JPEG standards committee. In this paper, we refer the numbers of scans being downloaded as fidelity levels. For example, we say that a user is at fidelity level 9 if 9 scans of this image have been downloaded. To evaluate and compare our interaction techniques, the proxy also logs all user requests and interactions with the system.

A small private network was set up in our lab to conduct the experiment. The web server and transcoding proxy runs in a Pentium 2.4GHZ desktop computer with 512 MB memory. The modified IE6 browser is running on the IBM Thinkpad R51 laptop computer with 14 inches screen.

4.2 Methodology

4.2.1 Participants

16 volunteers participated in this experiment, in addition to three pilot participants. All of the participants were graduate student in computer science department. All of them were told that the experiment was half an hour in length and they could withdraw the experiment without penalty at any time.

4.2.2 Experiment design:

The participants were randomly assigned into one of the following 4 condition groups, where each group has 4 participants:

1. Mouse right-click interaction, with 19.2kpbs connection speed. We will abbreviate this condition as “click at low speed”.
2. Mouse right-click interaction, with 192kpbs connection speed. We will abbreviate this condition as “click at high speed”.
3. Mouse wheel scroll interaction, with 19.2kpbs connection speed. We will abbreviate this condition as “scroll at low speed”.
4. Mouse wheel scroll interaction, with 192kpbs connection speed. We will abbreviate this condition as “scroll at high speed”.

The tasks for this experiment were to recognize text content of different sizes. Each participant was presented with a total of 24 JPEG images, among which were 8 images with font size 12 (small), 8 with font size 24 (medium) and 8 with font size 48 (large). Therefore, participants were expected to complete 8 trials for each font size condition. Notice that the order of the images with different font sizes appeared to the participants in a randomized manner. Each trial contained 10 randomly generated alphabets, presented in one line. These images with random text were pre-generated and were split into 15 scans.

4.2.3 Experiment procedure

Participants were first presented with a warm-up experiment for the interaction technique assigned to them. They were asked to familiarize themselves with assigned interaction technique by performing similar clicking/scrolling tasks as in the experiment. We deliberately present the participants non-text content in warm-ups because we do not want them to have prior knowledge of how to best adapt text content. Once the experiment started, text content with particular font sizes was presented to the user through NISTNET [10] bandwidth emulator, starting at the lowest fidelity level. The participant can ask for subsequent refinements using their assigned interaction tools (either click or scroll). Once the participant recognized the text on the screen, they typed the alphabets on a textbox which also appeared on the screen. They were not able to proceed to the next trial until the alphabets they typed matched up with the correct sequence of alphabets on that image.

A calculation of the total number of trials is provided below:
4 groups
* 4 users
* 3 sizes
* 8 trials
= 384 trials

4.3 Factors and dependent variables

As we can see, this experiment was a 3*2*2 mixed factorial design. The with-in subject factor is the font size of the alphabets. As we have mentioned above, this factor has three levels, large, medium and small size. There are two between-subject factors, both has two levels, namely the connection speed (12.9k and 129k) and interaction techniques (click and scroll).

There are three dependent variables that we are interested in. One of them is the consumed bandwidth, which is a count of numbers of fidelities requested by a participant in one trial. Notice that fidelities are around the same size of each other. Therefore, fidelity count is a good measurement of bandwidth consumption. The second one was the task completion time, which measured the time it takes to finish one trial. The last dependent variable is number of interactions, which is simply a count on number of interactions made between the participants and the adaptation system. Notice that clicking only allows one fidelity increment per interaction, but scrolling allows multiple fidelity increments at one interaction.

4.4 Experiment hypothesis

H1: We anticipated a significant main effect on techniques, across all dependent variable. Namely, interaction interfaces make a difference to adaptation system.
H2: We anticipated that scrolling will reduce number of interactions, but increase the bandwidth consumptions to finish a trial.
H3: We anticipated an interaction effect on dependent variable bandwidth, between connection speeds and interaction techniques. High speed connection will result in more bandwidth consumption differences between two techniques than the low speed.

4.5 Experiment Results:

We divided our results into three parts according to the dependent variables we are trying to measure. In each part, we will compare the two techniques with respect to this dependent variable. Finally, we will briefly summarize our findings by looking at how these dependent variables relate to each other.

4.5.1 Bandwidth

Figure 1 illustrates how much bandwidth is needed to finish the task among different users. The horizontal axis indicates all 16 users grouped into 4 controlled groups: Low speed clicking, Low speed scrolling, high speed clicking and high speed scrolling. The Bars with three different colors represent three types of font sizes. The heights of the bars indicate the average bandwidth consumed by that user for a particular font size, over the 8 trials.

It is apparent from the graph that scrolling consumes more bandwidth. In fact, a one way ANOVA on bandwidth shows such significant effect on the interaction techniques ($F(1,12)=10.87$, $p=0.006$). Notice that at low speed, the bandwidth consumption differences are not significant between the two widgets. This is because the task completion time is tightly bounded by the low connection speed. A little extra usage of bandwidth would result in much latency. To avoid the unnecessary waiting time, the participants from both scrolling and clicking groups would try hard to minimize the bandwidth usage at low speed.

At high speed, the differences between these two techniques are much more significant in respect of bandwidth consumptions. We consequently accept hypothesis H3, which says that high speed connection will result in more bandwidth consumption differences between clicking and scrolling. At this condition, the connection speed is 10 times faster and the participants no longer experience long waiting time. Instead, they tried to request higher fidelity levels to make sure that they can easily read off the text from the screen. Participants who used clicking interface received feedback after each fidelity improvement. They would obviously stopped fidelity requests once they can read the text. On the other hand, scrolling participants did not know which fidelity would be sufficient for the task, they would normally overshoot to minimize their interactions with the system.

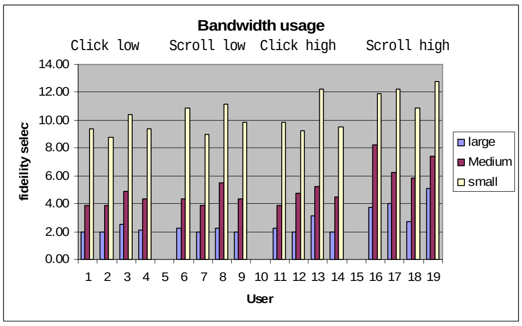

Figure 1

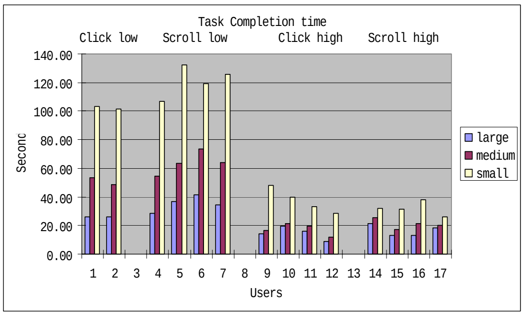

Figure 2

4.5.2 Task Completion Time

In the real world, the incentive of choosing one interface over another is either time or cost. The cost is usually associated with bandwidth usage, which we have discussed above. In this experiment, we also measured the time it took for the user to complete one trial. Notice that the time that users spent on typing down the answer was also included. Hence we explicitly made sure that every participant typed at approximately same speed.

Figure 2 illustrates the time it takes to finish different tasks by each of the 16 users. Notice that it is in exactly the same format as the bandwidth figure. From this figure, we learned that scrolling takes more time than clicking. A one way ANOVA on this dependent variable confirms this evidence (F(1,12)=5.551, p=0.036). This significant difference mostly came from low speed condition. Our initial guess was that at low speed, scrolling consumes more bandwidth than clicking. However, we have leant from the section 4.5.1 and figure 1 that this was not the case. Therefore, we believe that this time difference was not introduced by the network latency, but rather by psychological factors. Because the scrolling participants could request multiple fidelity increment at one time, this would result a longer waiting time for the downloading. As a result, the participants were likely to be less patient and ended up trying to guess instead of requesting further fidelity improvement, which further slow down the task completion. Therefore, a potential improvement to scrolling interface that we could implement is to make the adaptation process more interactive, by allow the user to stop the current downloading and display immediately the image of the fidelity that has been downloaded.

4.5.3 Interactions

Figure 3 shows how individual participants interact with the browser during the experiment. We define one interaction as one single request for fidelity improvement. Notice that we treat the initial downloading of the lowest fidelity image as one interaction. Hence by default, all trials will receive one interaction before any clicking and scrolling. As expected, scrolling clearly resulted in a fewer number of interactions (F(1,12)=130.65, $p<0.001$), which accepted hypothesis H2. In fact, scrolling only resulted in average 2.13 interactions for one task. This has already included the one default interaction made when downloading the lowest fidelity image. Hence we conclude that if we want to minimize the number of interactions, then scrolling is an ideal option for interactions.

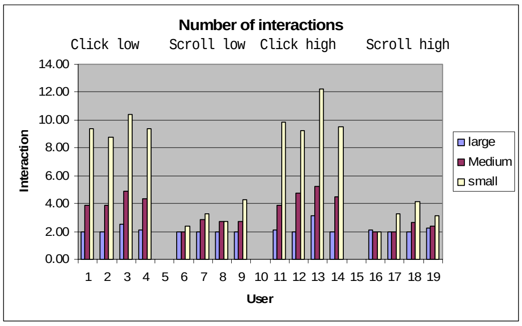

Figure 3

4.5.4 Locality

We refer locality in our experiment as the tendency of all participants to select one or two fidelity class for the text content of the same font size. Locality is important if we want to use our system as a refinement tool for automatic adaptation. Clearly, the more localities we obtain from user traces, the better predictions we can make. Table 1 illustrates the locality comparison between the two interaction techniques with a specific example. The table shows the distributions of fidelity requested for all small images, in a low connection speed environment. The distribution looks similar under other image size and connection speed conditions. “%cli” refers to percentage of trials in which the participant in the clicking group select this particular fidelity, and “%scr” refers to percentage of trials in which the participant in the scrolling group select that fidelity. The most dominating class of fidelity for both interaction techniques are highlighted in the table. As we can see, clicking results in a much higher locality as fidelity selections are much more centralized. The dominating fidelity class is fidelity 5, which made up 50% of all trials. If we assume that the participants who selected fidelity 4 was also satisfied when serving at fidelity 5, then we would have in total of 84.4% of selections covered by one fidelity class. On the other hand, scrolling does not have such high locality as compared to clicking. The fidelity selected ranges from fidelity 4 to fidelity 12, as opposed to the range from fidelity 2 to fidelity 6 in clicking. More importantly, the dominating fidelity class only represents 34% of all the selections. Obviously, if we want to use our system to help the user refinement part of automatic adaptation system such as CDA [1], we would use clicking as our basic interaction technique, since clicking results in more centralized user selection, which makes the predictions more accurate.

Fidelity	click	%cli	scroll	% scr
2	1	0.031	0	0
3	1	0.031	0	0
4	11	0.344	2	0.062
5	16	0.5	6	0.187
6	3	0.093	11	0.343
7			1	0.031
8			5	0.156
9			2	0.062
10			3	0.093
11			1	0.031
12			1	0.031

Table 1

4.5.5 Summary of findings

We would like to summarize our discoveries with respect to both interaction techniques. In this experiment, we discovered that scrolling indeed reduces the number of interactions that the users would have to make with the adaptation system. However, such advantage comes with the cost of wasting more bandwidth or time. Furthermore, it reduces the localities of user selections, which subsequently makes it hard to incorporate into prediction features. On the other hand, clicking conserves bandwidth and time for users, which consequently lower the cost for surfing the web. User selections tend to be more centralized to one or two fidelity classes which help improving the accuracy of predictions. However, clicking involves more interactions between the users and our adaptation system, which degrades the system usability.

5. Conclusion

In this paper, we have introduced User Driven Adaptation, a novel approach for content adaptation that recognizes user semantics and individual preferences, along with its various advantages over traditional automatic adaptation approaches. Meanwhile, we have shown that we can use HCI approaches to present the most suitable interface for this system, such that the burdens to the user are minimized. The controlled experiment presented above illustrates the huge differences made by different interaction interfaces. Therefore, we have reasons to believe that the user driven approach is promising if we adapt appropriate interface design.

So far, we have only outlined the framework of our adaptation system and evaluate the most basic elements in our interface. In the future, we plan to extent the current work and start to add more features to the interface. For instance, we plan to develop interaction techniques to deal with multiple images.

We also plan to add more system feedback features, such as estimated downloading time for a particular quality level, to help users make better adaptation decisions. Furthermore, we will gradually move our adaptation interface to smaller mobile devices such as PDAs and cell phones, where we can apply gestures or other interaction techniques to aid the adaptation.

References

[1] I.Mohomed, A.Chin, J.Cai and E. deLara. Community Driven Adaptation: Automatic content adaptation in pervasive environments. Proceedings of the 6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), English Lake District, UK, December 2004.

[2] E.deLara, D.S. Wallach, and W. Zwaenepoel. Puppeteer: Component-based adaptation for mobile computing. *Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems*, San Francisco, California, Mar. 2001.

[3] B. D. Noble, M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, and K. R. Walker. Agile pplication-aware adaptation for mobility. *Operating Systems Review (ACM),* 51(5):276–287, Dec. 1997.

[4] M. Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, 2001.

[5] B. N. Schilit, J. Trevor, D. M. Hilbert, and T. K. Koh. Web interaction using very small internet devices. *IEEE Computer*, 35(10):37–45, 2002.

[6] J. R. Smith, R. Mohan, and C.-S. Li. Transcoding internet content for heterogeneous client devices. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, Monterey, California, May 1998.

[7] Y. Dotsenko, E. de Lara, D. S.Wallach, and W. Zwaenepoel.Extensible adaptation via constraint solving. In *Proceedings of the 4th IEEEWorkshop on Mobile Computing Systems and Applications*, Callicoon, New York, June 2002.

[8] K. Britton, R.Case, A. Citron, R. Floyd, Y. Li, C. Seekamp, B. Topol, and K. Tracey. Transcoding: Extending e-business to new environments. *IBM Systems Journal*, 40(1):153–178, 2001.

[9] JPEG standards committee: http://www.jpeg.org

[10] NistNet: http://www.antd.nist.gov/**nistnet**