Context-Aware and Adaptive Web Interfaces: A Crowdsourcing Approach

Michael Nebeling and Moira C. Norrie

Institute of Information Systems, ETH Zurich
CH-8092 Zurich, Switzerland
{nebeling,norrie}@inf.ethz.ch

Abstract. Web site providers currently have to deal with the growing range and increased diversity of devices used for web browsing. It is not only technically challenging to provide flexible interfaces able to adapt to the large variety of viewing situations, but also costly. We discuss the idea and challenges of adopting a crowdsourcing model in which end-users can participate in the adaptation process with the goal of enabling a much wider range of use contexts to which applications can adapt.

1 Introduction

Nowadays, web-based services are accessed from a wide range of devices with very different characteristics, not only in terms of screen size and resolution, but also supported input and output modalities. It is becoming increasingly difficult and also cost intensive for web site providers to cater for the large variety of client devices used today. We believe that the only feasible way to address this challenge is to adopt a crowdsourcing model in which end-users can become involved in the adaptation of web interfaces for different devices and preferences. To achieve this, we have started to address the technical challenges of designing a model, architecture, language and runtime system capable of supporting the dynamic definition and deployment of web site adaptations in a safe and efficient manner [1]. In this paper, we focus on the crowdsourcing principles underlying the approach and discuss how they relate to previous research on context-aware and adaptive interfaces and the many new forms of human computation [2].

We start by discussing different approaches to adaptation in Sect. 2. We then characterise our crowdsourcing approach in Sect. 3 and discuss how we aim to improve user participation, quality control and aggregation of contributions. Finally, Sect. 4 outlines remaining challenges and our ongoing research efforts.

2 Adaptation Approaches

In a recent study, we have shown that the majority of web sites today still use a fixed layout designed for a resolution of 1024x768 pixels despite the fact that average screen settings are much higher than that [3]. Rather than thinking of more flexible layout solutions, a number of web sites nowadays come in several

A. Harth and N. Koch (Eds.): ICWE 2011 Workshops, LNCS 7059, pp. 167–170, 2011.

special versions. However, the employed adaptations are typically designed for only a certain class of devices, e.g. for touchphones or tablets. The same is true for most of the advanced algorithms that promote automatic adaptation techniques, as these typically require substantial server-side processing [4], or only work for very specific scenarios, e.g. desktop-to-mobile adaptation [5]. The new approach for devices such as the iPhone or iPad is to provide special browser support, e.g. by enabling gestures for zooming and automatic scaling of the viewport, but this means that the problem is treated as a matter of input technique rather than an interface design issue and also requires user intervention.

Research in web engineering has focused on languages and model-driven approaches to support context-awareness and adaptivity in applications [6,7]. A second stream of research has looked at different abstraction levels of the user interface to be able to adapt to multiple user, platform and environment contexts [8]. The authoring of adaptive and multi-modal user interfaces has also been the subject of extensive research [9]. The suggested development processes typically start from some kind of domain or task model and logical descriptions of the interface. This is then followed by subsequent transformation steps to generate the final interface for a particular context of use. However, all these approaches rely on developers to specify the required forms of adaptation, which is almost impossible with the diverse and rapidly evolving range of settings.

Our new idea was to adopt a crowdsourcing model in which system developers provide an initial interface and adaptive features of the system can evolve at runtime with the help of users [1]. A number of interesting systems have been built to demonstrate the potential benefits of using crowdsourcing techniques. For example, Soylent [10] is an extension of Microsoft Word that allows users to create tasks related to the document, such as shortening of paragraphs or proof-reading, to be carried out by other users. Another useful system is HelpMeOut [11] which recommends potential solutions for compiler errors and runtime exceptions that other programmers have also encountered. While many crowdsourcing solutions tend to build on external services such as Amazon's Mechanical Turk[1], our goal was to *embed* crowdsourcing mechanisms into applications in order to provide end-users with the tools to collectively solve problems such as the lack of adaptivity. To achieve this, we first had to develop visual tools for end-users to be able to design new adaptations directly in the browser and then extend the common web application architecture with several components so that adaptations can also be shared between users and even across sites (Fig. 1). Popular examples of crowdsourcing platforms with somewhat similar goals are programmableweb.com and userscripts.org, where already large communities of active users maintain shared collections of web mashups and augmentations.

3 Crowdsourcing Approach

For a characterisation of our approach, we will use the taxonomy proposed by Quinn and Bederson [2].

[1] http://mturk.com

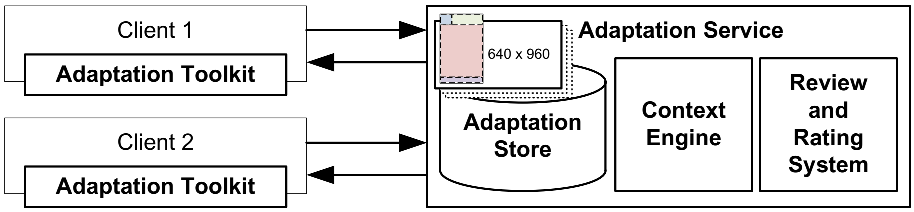

Fig. 1. Architecture showing two clients that contribute and consume web site adaptations for small-screen devices using the adaptation operations provided by the toolkit and server-side components for sharing and deployment of crowdsourced adaptations

Motivation. User motivation and participation can depend on many factors. Our intention was to provide end-users with simple, visual tools for customising directly the final interface according to the viewing situation without the need to understand the underlying languages or models. This makes it easy also for non-technical users. We expect a major effect on motivation by letting users see that the interface can improve through their actions and, similar to [12], by letting them know that other users can so benefit as well. Also, what to them may seem like a *personalisation* of the interface will in fact describe an *adaptation* for a particular use context when it is shared with the crowd. Even if only a few users share the adaptations created on their devices, then already many others using the same devices can directly benefit. Along the lines of Quinn and Bederson, this would mean that motivation in our approach is primarily guided by *implicit work* and a very light form of *altruism*.

Quality. In our crowdsourcing model where the viewing quality of a web site is primarily regulated by end-users, quality control may involve a number of schemes where we have given the highest priority to *defensive task design*. The main idea was to provide adaptation operations concerning only those aspects of the design that are directly related to the viewing context, such as size and position of web site elements, rather than content or functionality [1]. To increase the quality of adaptation scenarios, we capture precisely the context in which the adaptation process takes place by collecting all kinds of context information related to the device and the user. Finally, the recommender system used by our platform to determine the best-matching adaptations for a given context is complemented by a review and rating system. This enables *multilevel review* and a *reputation system* corresponding to [2], where system administrators, or promoted users, have a bigger say so that their approval or rejection have significant impact on the ranking and therefore the deployment of adaptations.

Aggregation. Also important for our crowdsourcing approach is the the idea of *iterative improvement*. Depending on the individual viewing situation and the current quality of adaptations, participating users may come up with whole new designs or only provide minor improvements over the original layout or other user-adapted versions. We support this by building on an adaptation technique that uses cascading stylesheet definitions and server-side components that are capable of managing different versions of adaptations for the same context [1].

4 Conclusion

Given the large variety of devices used for web browsing, we have started to explore a crowdsourcing approach to support web site providers in the design of flexible interfaces and to enable a much wider range of use contexts to which applications can adapt. As a first step, we developed a platform and visual toolkit for crowdsourced adaptations of web interfaces, which we discussed in this paper. We are currently carrying out extensive technical evaluations of this approach when it is applied to a number of existing web sites and used by a larger group of users. In particular, we are experimenting with different sharing and ranking modes for the controlled definition and deployment of new adaptations. This is important for cases when really a crowd of users contribute with web site adaptations and several different sets of adaptations have been defined for the same or similar contexts. The data we are collecting can provide interesting insight into adaptation requirements, help us improve both the underlying methods and the overall crowdsourcing approach and potentially lead to new web design patterns and guidelines for the wide variety of devices and platforms available today.

References

1. Nebeling, M., Norrie, M.C.: Tools and Architectural Support for Crowdsourced Adaptation of Web Interfaces. In: Auer, S., Díaz, O., Papadopoulos, G.A. (eds.) ICWE 2011. LNCS, vol. 6757, pp. 243–257. Springer, Heidelberg (2011)
2. Quinn, A.J., Bederson, B.B.: Human Computation: A Survey and Taxonomy of a Growing Field. In: Proc. CHI (2011)
3. Nebeling, M., Matulic, F., Norrie, M.C.: Metrics for the Evaluation of News Site Content Layout in Large-Screen Contexts. In: Proc. CHI (2011)
4. Schrier, E., Dontcheva, M., Jacobs, C., Wade, G., Salesin, D.: Adaptive Layout for Dynamically Aggregated Documents. In: Proc. IUI (2008)
5. Hattori, G., Hoashi, K., Matsumoto, K., Sugaya, F.: Robust Web Page Segmentation for Mobile Terminal Using Content-Distances and Page Layout Information. In: Proc. WWW (2007)
6. Ceri, S., Daniel, F., Matera, M., Facca, F.M.: Model-driven Development of Context-Aware Web Applications. TOIT 7(1) (2007)
7. Frăsincar, F., Houben, G.J., Barna, P.: Hypermedia presentation generation in Hera. IS 35(1) (2010)
8. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A Unifying Reference Framework for Multi- Target User Interfaces. IWC 15 (2003)
9. Paternò, F., Santoro, C., Mäntyjärvi, J., Mori, G., Sansone, S.: Authoring pervasive multimodal user interfaces. IJWET 4(2) (2008)
10. Bernstein, M.S., Little, G., Miller, R.C., Hartmann, B., Ackerman, M.S., Karger, D.R., Crowell, D., Panovich, K.: Soylent: A Word Processor with a Crowd Inside. In: Proc. UIST (2010)
11. Hartmann, B., MacDougall, D., Brandt, J., Klemmer, S.R.: What Would Other Programmers Do? Suggesting Solutions to Error Messages. In: Proc. CHI (2010)
12. Rashid, A.M., Ling, K.S., Tassone, R.D., Resnick, P., Kraut, R.E., Riedl, J.: Motivating Participation by Displaying the Value of Contribution. In: Proc. CHI (2006)