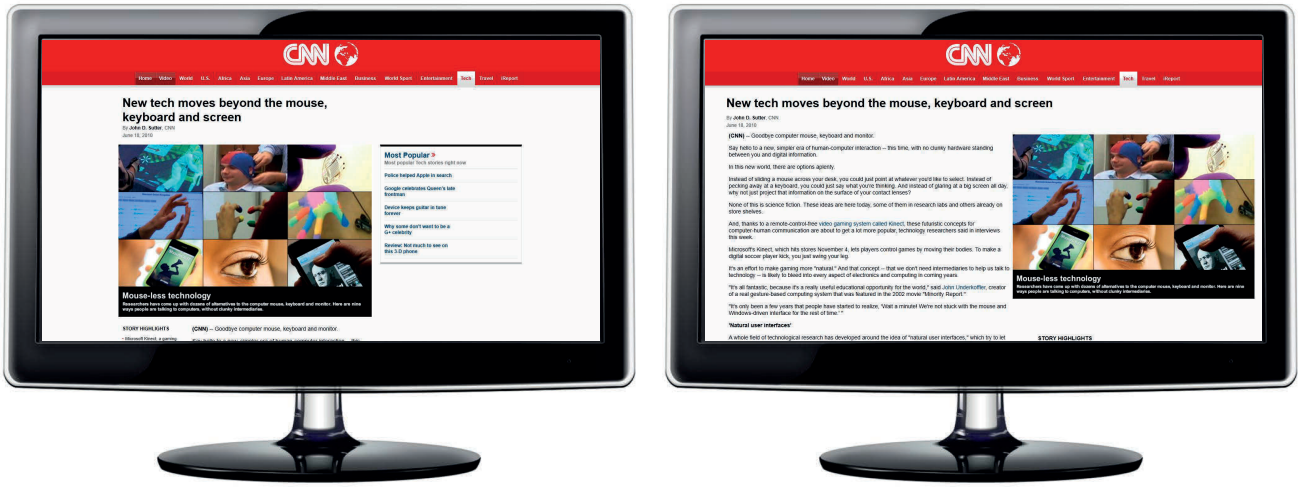
CrowdAdapt: Enabling Crowdsourced Web Page Adaptation for Individual Viewing Conditions and Preferences

Michael Nebeling, Maximilian Speicher and Moira C. Norrie
Institute of Information Systems, ETH Zurich
CH-8092 Zurich, Switzerland
{nebeling,norrie}@inf.ethz.ch, maximilianspeicher@gmx.de



(a) Original design of the CNN web site

(b) Crowdsourced adaptation

Figure 1: Juxtaposing the original CNN web site viewed on a medium-size screen at 1680x1050 pixels and the best-matching, highest-ranked crowdsourced layout that uses more screen real estate and increases the amount of text visible without scrolling.

ABSTRACT

The range and growing diversity of new devices makes it increasingly difficult to design suitable web interfaces for every browsing client. We present *CrowdAdapt*—a context-aware web design tool that supports developers in the creation of adaptive layout solutions for a wide variety of use contexts by crowdsourcing web site adaptations designed for individual viewing conditions and preferences. We focus on one experiment we conducted for an existing news web site using CrowdAdapt (i) to explore the design space in terms of layout alternatives created by the crowd, (ii) to identify adaptation preferences with respect to different viewing situations, and (iii) to assess the perceived quality of crowd-generated layouts in terms of reading comfort and efficiency. The results suggest that crowdsourced adaptation could lead to very flexible web interfaces informed by individual end-user requirements. In particular, scenarios such as the adaptation to large-screen contexts that the majority of web sites fail to address could be supported with relatively little effort.

Author Keywords

Adaptive layout; responsive web design; crowdsourcing.

ACM Classification Keywords

H.5.2 User Interfaces: Screen design

INTRODUCTION

The growing range and increased diversity of new devices that vary widely, not only in terms of screen size and resolution, but also supported input and output modalities, makes it difficult to design web interfaces that adapt well to every browsing client. Many of the existing methods for automatic adaptation provide little control for developers and are typically tailored to very specific scenarios such as desktop-to-mobile adaptation [8] and therefore not easily extended for other use cases. On the other hand, more comprehensive solutions to accommodate a much wider range of use contexts require advanced layout generation techniques [18] that are, however, often not feasible in a web context.

Given the need for more flexible web interfaces, but the high design effort and costs required to create them, our goal is

EICS'13, June 24–27, 2013, London, United Kingdom.

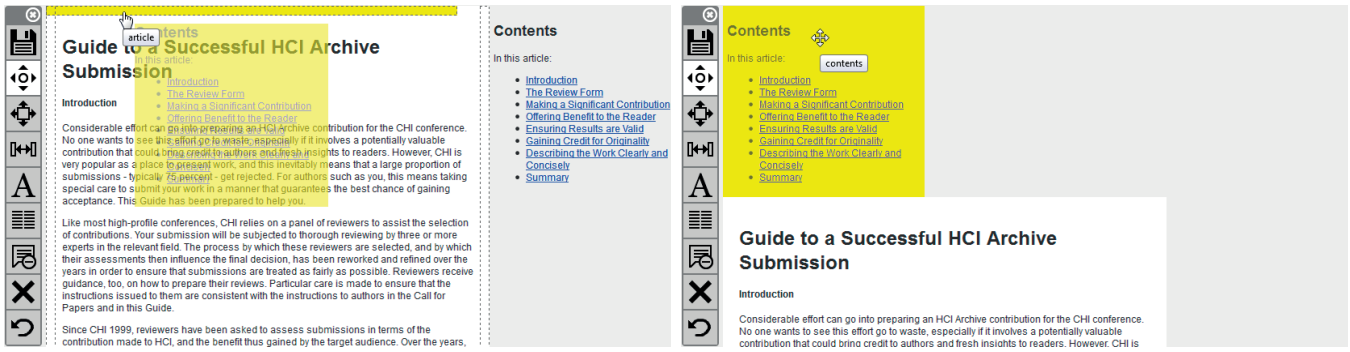


Figure 2: Demonstrating CrowdAdapt's move operation (before and after) for re-anchoring the sidebar element as an example.

to enable crowdsourced adaptation of existing web sites. Recently, we have started to address the technical challenges of designing a model, architecture and runtime environment capable of supporting the dynamic definition and deployment of web site adaptations in a safe and efficient manner [14]. In this paper, we present *CrowdAdapt*, a context-aware web design tool that we developed based on these building blocks to enable and test our concept of crowdsourced adaptation. Our specific combination of end-user development and crowdsourcing for context-aware adaptation is novel and marks an important step forward in research on mixed-initiative interfaces [4]. Specifically, CrowdAdapt is able to aggregate crowd-generated layouts designed for different window and screen sizes to provide an adaptive layout solution catering for a large variety of viewing conditions. We present one experiment in detail in which CrowdAdapt was used in a real-world setting for the CNN web site (Fig. 1), one of the world's most popular news web sites according to the Alexa ranking[1].

An important scenario that we wanted to enable using our approach is the adaptation to large, high-resolution displays that the majority of web sites still fail to support [13]. This is surprising given the fact that average screen sizes and resolutions have dramatically increased over the past few years. For web design, however, wide-screen contexts present a new setting in need of interface design guidelines and flexible layout solutions to make appropriate use of the greater screen real estate and increasingly horizontal screen distances. Typical design issues with current web layouts at larger viewing sizes are that background rather than content often fills the screen in the case of static layout, or that the text spreads across the entire screen width in fluid layout. This means that screen real estate is potentially wasted, or that readability is impacted due to excessively long lines of text. Our vision is that crowdsourced adaptation will alleviate the problems in such cases. For example, one of our study participants used CrowdAdapt to reduce the need for scrolling for the static, fixed-width layout of CNN at larger window sizes by adjusting the spatial layout and size of the headline, image and story areas (Fig. 1). We do not want to make a case for this particular layout, but instead use it as an example of a crowd-generated layout that illustrates the potential of crowdsourced adaptation.

[1]http://www.alexa.com/siteinfo/cnn.com

In the following, we present the key components of CrowdAdapt. For each component, we will discuss both the implemented functionality and the key design challenges that had to be addressed. This is followed by a description of our experiment based on the CNN web site and a discussion of the results obtained from user evaluations.

CROWDADAPT

The key idea behind CrowdAdapt is to allow end-users to adapt the interface to their specific use context if it is insufficiently supported by the current web page design. Drawing from individual user contributions allows the system to build an adaptive layout solution that caters for a wide variety of device characteristics and user preferences. At the same time, the design task can be kept rather simple since each user only has to think in terms of their own setting. CrowdAdapt provides a visual design environment that augments web pages loaded in the browser to allow users to customise the layout. The changes are then deployed on a server in the form of a new layout template which will be automatically downloaded and applied in subsequent visits of the same user. Moreover, the layouts are automatically shared with other users so that new visitors using the same, or a similar, device can directly benefit from the adaptations. CrowdAdapt supports two deployment modes so that it can either be bundled directly with a web site or be installed separately as a browser plugin. The first deployment scenario does not require separate infrastructure and allows contributors to share their customisations with other users of the same site. Also other users require no additional software for viewing the site in a shared layout, which may be preferred by those primarily in the role of the consumer. The second scenario follows the popular example of userscripts.org, where a large collection of scripts for augmented browsing based on plugins such as Greasemonkey is self-maintained by an active user community. The CrowdAdapt plugin allows users, not only to create new adaptations for any web site they would like to customise even if the web site provider does not directly support this, but also to obtain adaptations contributed by plugin users for other sites.

Direct Manipulation Toolkit

The core of CrowdAdapt is the direct manipulation toolkit running on the client to provide the user with a set of visual tools for customising the interface directly in the browser.

There were three major challenges in developing this component. First, we had to decide on the concrete set of adaptation operations that would be required to adjust interfaces to different conditions. Second, all operations had to be designed with non-technical end-users in mind. Finally, each adaptation must yield a valid manipulation of the interface in order to keep the underlying implementation intact.

The current set of 7 adaptation operations were derived from an analysis of the differences between web page layouts at different viewing sizes and the changes required to make effective use of both small and large-screen settings [13]. The example web sites that we considered for the analysis ranged from news web sites, blogs, wikis and forums to other types of applications such as web mail and social media sites such as Facebook and Twitter that are typically used by active user communities for both the consumption and sharing of content. The defined operations, especially when used in combination, cater for a wide range of adaptations, but also reflect what is technically feasible without imposing particular web design conventions. Note that we refer to web page "elements" in terms of the rendered interface. At the hypertext level, elements refers to the HTML DOM elements part of the `body` which includes both simple elements, such as headings `h1` to `h6`, or container elements such `div` and `span` which may nest other elements.

The two main operations are "move" and "resize". Move repositions web page elements in the document via drag-n-drop. Elements can be freely positioned in the page or re-anchored and snapped to other elements by dropping on either edge of the target (Figure 2). Resize scales elements in horizontal and/or vertical direction by dragging the edges or corners as known from common window managers. While the actions can be controlled via additional handles that are dynamically displayed for in-place element manipulation, the remaining operations trigger a context-sensitive popup menu on the selected element. The "spacer" operation increases or decreases the space around an element via "+" or "-" menu options. The "hide" function toggles the visibility of an element or restores previously hidden ones via the menu. Alternatively, "collapse" replaces an element with a placeholder link to allow users to later unfold the content. Both of these operations are especially useful for adaptation to smaller screen sizes, but also as intermediate design steps when bigger changes are performed to a more complex layout. The "font size" operation increases or decreases the text height. Moreover, the "multi-column" operation controls the number of columns used for content layout, which is particularly useful for large-screen settings. While only one tool can be active at a time, operations can be subsequently combined with each other as well as reverted via an undo command.

Each of the operations updates the CSS and/or the HTML DOM as required, which is straightforward in most cases. For example, for freely moving or resizing elements, only the CSS position and dimension properties are adjusted. Similarly, the spacer operation controls the CSS margin and multi-column layout is based on the new CSS3 properties[2]. On the other hand, for re-anchoring moved elements and in order to support element nesting and maintain the z-index, the HTML DOM is also updated by internally moving the dragged element node either before or after the drop target. Hiding an element again just toggles the CSS display property, which is sufficient to prevent that the inner content is loaded by the client in future visits. While these design decisions generally help to maintain the functionality of the web interface, it is still possible to break JavaScript that refers to page elements via the DOM element path rather than the ID if the element was moved to a new position.

CrowdAdapt does not require manual configuration for every web site. By default, operations can be invoked on all page elements with ID so that a unique reference to the respective DOM node can be maintained. This is usually feasible because the main web site components are typically labelled for CSS and programmatic access via JavaScript anyway. However, CrowdAdapt can be configured by both web developers and users. For example, web developers may extend, or constrain, the scope of adaptations by including, or excluding, certain web page elements based on jQuery selectors[3]. Users can configure the toolkit to automatically deactivate text selection, links to other pages and embedded objects such as videos or Flash animations in order not to accidentally trigger associated actions while customising the web interface.

Adaptation Engine

While the toolkit operations allow users to fit the interface to their individual viewing condition, the actual context-awareness comes from the adaptation engine which is responsible for managing the overall adaptive interface solution based on crowd contributions. With each manipulation of the web interface, CrowdAdapt learns a new adaptation rule for the current context and user. For example, as the user increases the width of the main content container to 1600 pixels on a Full HD display, CrowdAdapt internally generates a new design instruction similar to the one below which describes the adaptation based on CSS3 media queries[4].

```
@media only screen
  and (min-width: 1870px)
  and (max-width: 1970px)
  and (device-width: 1920px)
  and (orientation: landscape) {
 #main { width: 1600px; }
}
```

Based on the user interaction and according to the respective adaptation operation, the adaptation rules are generated in three steps. First, the context information is automatically collected in terms of the window size, screen resolution and orientation of the client device. CrowdAdapt then approximates the window size using min/max values based on a configurable 100 pixel threshold to match similar viewing conditions. Finally, the actual adaptation rule is composed of the condition that triggers the adaptation with respect to the client context and the design rule that alters the original layout to better fit the new condition. All manipulations are recorded

[2]http://www.w3.org/TR/css3-multicol

[3]http://api.jquery.com/category/selectors

[4]http://www.w3.org/TR/css3-mediaqueries

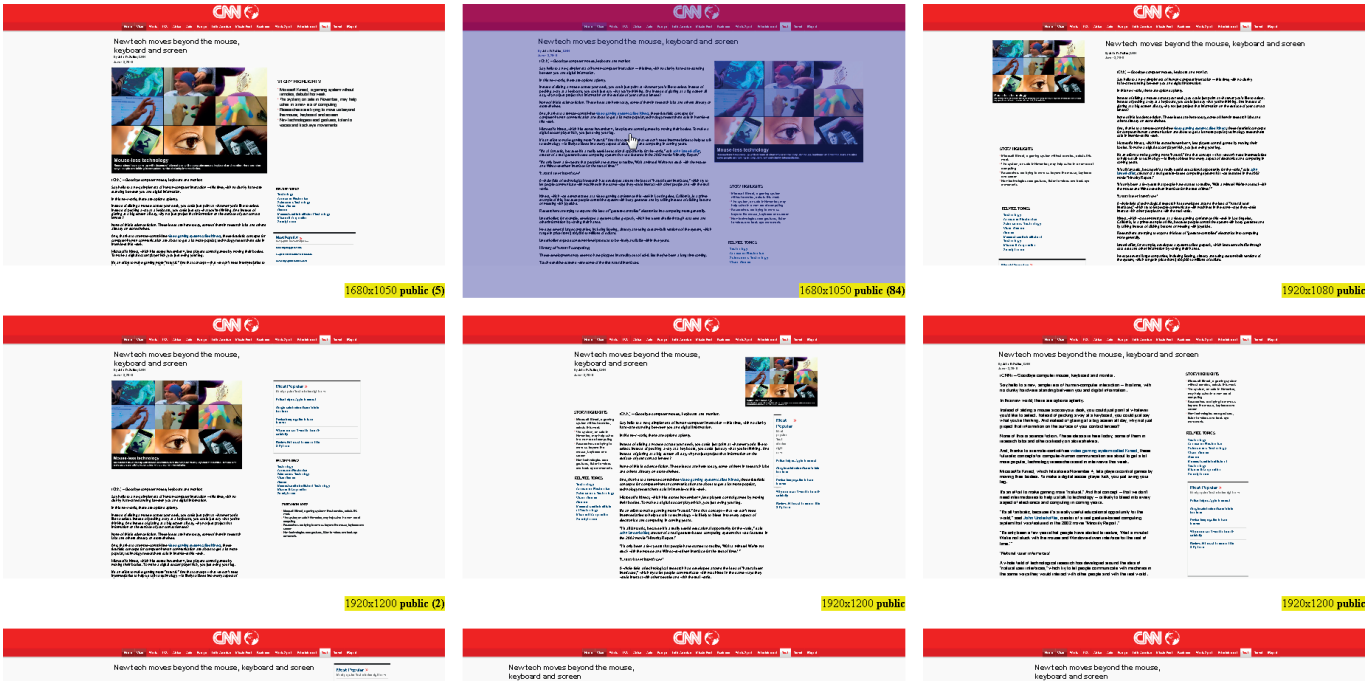


Figure 3: The in-built review and rating system allows end-users to preview adaptations and select the preferred layout.

on the client-side and sent to the server in suitable 30-second intervals as well as when the "save" function is used. In addition, the user is asked to save changes before navigating to a different site or closing the CrowdAdapt toolkit or browser window in order to confirm the changes. All generated design rules together then define the set of adaptations that describe the custom web interface similar to a new layout template.

By default, adaptations are strictly collected per URL path. However, many modern web sites dynamically generate pages with different content from a single URL based on different query strings. This enables reuse of adaptations between pages based on naming conventions. Alternatively, CrowdAdapt can be configured by the host site to aggregate page-specific adaptations based on URL pattern matching, which may be required for web sites that use URL rewriting. The same feature can also be exploited for web sites that consist of many individual HTML documents that essentially share the same elements due to similar structure and layout. While CrowdAdapt's support for generalising adaptation rules across web pages and sites is still rather limited, the basic mechanisms and the means for configuration already provide a relatively flexible solution that works well with many existing web applications due to the fact that the underlying content engines typically generate pages based on a fixed set of common templates. Nevertheless, generalisability is a hard problem due to the lack of standards in web interface implementations. The discussion later in this paper will look at possible ways in which CrowdAdapt could be extended to provide more advanced support for reusing adaptation rules.

Deployment and Review System

Since the main goal of CrowdAdapt is to collectively improve the adaptability of a web site for different viewing situations, we decided that the adaptations created by one user are automatically shared with other users of the web site. While many different schemes are possible, the deployment of new adaptations is therefore kept rather straightforward and is in fact mostly done automatically by CrowdAdapt, but still regulated by the end-users as follows.

- When a new user visits the web site, CrowdAdapt dynamically determines the client context and fetches all adaptations that potentially match the user's setting. If no adaptations are available for the current context, the original layout will be used, but the user will be asked to customise the interface. Otherwise CrowdAdapt automatically applies the currently best-matching adaptation.

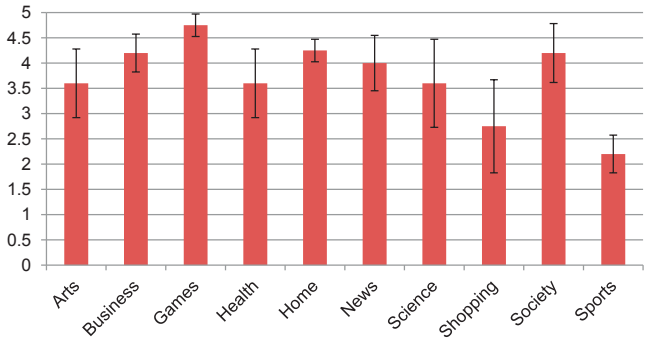
- The adaptations created or adopted by a user are automatically reused in subsequent visits of that user to seamlessly restore the custom interface without user invocation.

CrowdAdapt allows adaptations to be previewed and selected like different "themes" for the web site as illustrated in Figure 3. It can also be configured to always ask users before potentially matching crowdsourced adaptations are applied. Initially, custom layouts are ranked in terms of similar window and screen sizes, but the initial ranking is then influenced by user ratings so that matching adaptations will be ranked higher the more positive votes they receive.

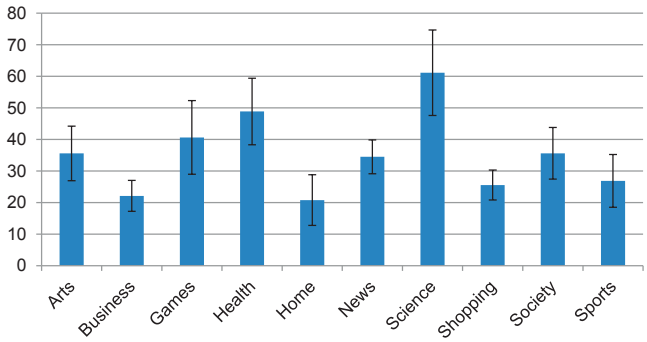
Implementation

The core client-side functionality is implemented based on the popular jQuery JavaScript framework[5] and its UI behaviours in combination with a PHP/MySQL server backend. To embed the tools in an existing web site, it is sufficient

[5]http://jquery.com



(a) Average ratings (1=poor, 5=excellent)



(b) Average DIV-ID ratios (higher is usually better)

Figure 4: Compatibility tests for 50 Alexa Top Sites

to link CrowdAdapt as an external JavaScript source and configure the adaptation engine with database settings. CrowdAdapt was developed and tested primarily with Firefox and requires version 3.6 or higher. However, in some versions, the multi-column function showed unpredictable behaviour so that elements could not be activated and modified after additional columns had been added. The CrowdAdapt plugin is an extended version of the script that additionally builds on Greasemonkey for Firefox to get access to any web page loaded in the browser. More configuration, e.g. for including or excluding certain page elements from the adaptation process and URL pattern matching, is usually optional, but may be required for some web sites as described earlier.

CrowdAdapt was tested on several popular sites. We carried out a study on 50 top sites from 10 different genres (Arts, Business, Games, Health, Home, News, Science, Shopping, Society, Sports) as ranked by Alexa[6]. Note that Alexa ranks top sites according to popularity and traffic. Demonstrating compatibility of CrowdAdapt for a selection of Alexa sites therefore means that many millions of users would benefit. CrowdAdapt was loaded into each site and the performance and compatibility assessed for two adaptation scenarios, namely small and large screens. A 5-point scale from $1 = poor$ for sites that showed major issues to $5 = excellent$ if all adaptations were successfully stored and reapplied was used for the rating. In addition, we computed the ratio of page elements that can be adapted without manual configuration.

[6]http://www.alexa.com/topsites

As shown in Fig. 4, most ratings were around 4=good with an average of 3.7 (sd=1.6, mode=5) across all categories. CrowdAdapt worked best with Business, Games, Home, News and Society sites (all above 4). By far the lowest ratings of 2.2 on average (sd=0.8, mode=3) showed in the Sports genre due to fixed-size, graphics-heavy designs of sites such as ESPN that are difficult to adapt without also editing background images etc. On average, 35% of DOM elements (sd=21.4) could be adapted using our ID-based approach with good results for the majority of sites. However, our tests revealed conflicts in some cases.

For example, due to how our proof-of-concept implementation prepares elements for move and resize, the move tool seemed very slow on Facebook and not always working correctly when trying to customise the position of elements. Facebook generates lots of IDs for many small components, such as posted comments which are, by default, all processed as potential drop targets. We were able to achieve a high performance when excluding unnecessary classes of DOM elements from customisation. Another example is Twitter where our tools generally worked well, but their Bootstrap UI toolkit[7] provides its own functionality to control layout based on a responsive grid, which might contradict CrowdAdapt rules. However, sites that employ responsive design are still relatively rare at this stage and also not our primary target, as they already make sure that the layout adapts to different screen sizes. Nevertheless, CrowdAdapt could be extended to better integrate with such frameworks. To enable also others to experiment with this, more information and the CrowdAdapt source code are available online[8].

EVALUATION

In a first experiment, we focused on how crowdsourced adaptation based on CrowdAdapt may be used to improve the reading experience in large-screen contexts for text-heavy sites such as CNN (Figure 1). CrowdAdapt was embedded in an example news article and configured so that the main article and sidebar content could be adapted. The site's header, top navigation bar and footer content were specifically excluded from customisation as this may be in the interest of the web site provider and also helped focus the user attention on the article layout. All adaptation operations were available except for the multi-column feature which we removed from the experiment due to browser compatibility issues. The adaptation engine was configured to partition contexts into widescreen and other screen formats. Similar to reports from DisplaySearch[9], widescreen here refers to HD resolutions of 1280x720 pixels and more, and 16:10 window size ratios. Our evaluation was guided by three questions:

- Are the proposed design tools usable and complete to support individual requirements and preferences?
- How would end-users make use of the system when designing for their viewing conditions?

[7]http://twitter.github.com/bootstrap
[8]http://dev.globis.ethz.ch/crowdadapt
[9]http://www.displaysearch.com

- Could crowdsourced evaluation based on CrowdAdapt overall enable a better user experience?

We aim to show the general potential of CrowdAdapt to be useful based on a qualitative analysis of the adaptations produced by the crowd. At the same time, its functionality and the feasibility of our design decisions concerning the adaptation tools and automatic deployment of new adaptations are assessed based on questionnaire feedback.

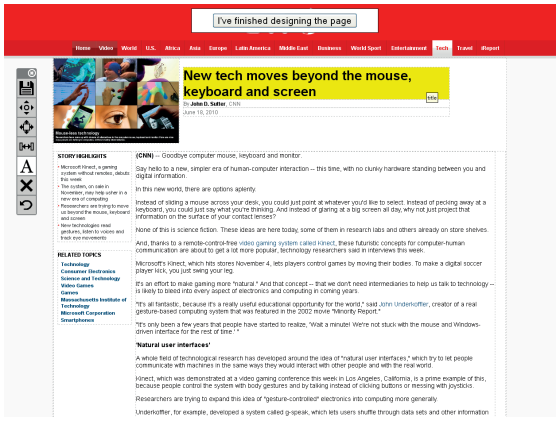
Method

To make for a controlled experiment while evaluating CrowdAdapt in its anticipated setting, the experiment was divided into two phases. Initially, the original layout was used as the basis for the experiment. In a second phase, this was changed to the best-ranked crowdsourced layout at this stage. With these two stages, we wanted to explore the design space based on different starting points and generally see whether users would appreciate layouts designed by other users and perceive them as an improvement over the original layout. Each new participant to the experiment was randomly given one of three tasks illustrated in Figure 5. Once they finished a task and submitted their responses, they were asked to work on additional ones as they liked. In the design task, participants were asked to customise the layout of the news article for their viewing situation and reading preferences and to provide feedback on the tools. Layouts were automatically shared if the design task was completed with questionnaire feedback. In the second task, participants compared and rated three layouts in random order—the original layout, a random matching adaptation not designed by them and the currently best-ranked matching adaptation. Ratings were collected in three steps where each step showed two previews for comparison, allowing users to focus on the differences between only two layouts at a time. The last task was specifically designed to test crowdsourced layouts when applied to the news article and collected user feedback in terms of reading comfort and efficiency. Participants were first asked to read the article and then to answer five questions on the text. Questions had to be answered by clicking on the respective text paragraph that contained the answer, rather than typing the answer directly since this required visual search and depends on layout quality as well as memory. In addition, the times for reading and answering were measured separately. The first task therefore assessed the direct manipulation tools, while the other tasks concerned the perceived quality and functionality of crowd-generated layouts and ensured that user feedback would not only be based on aesthetic considerations.

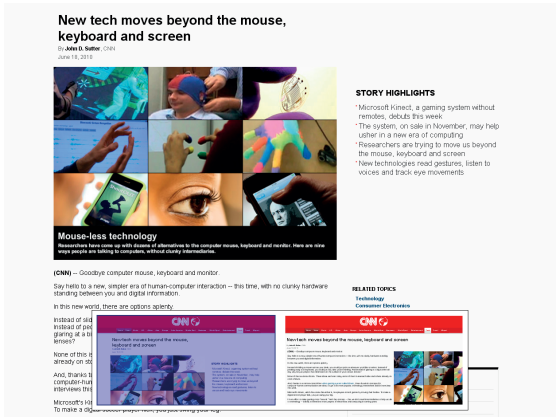
Results

Over a 10 day period, 93 participants who frequently consume online news on sites such as CNN were recruited via internal and public mailing lists, student forums, as well as social bookmarking sites Reddit and digg. The majority of participants compared layouts, 28 contributed a customised layout to the experiment and 42 provided reading feedback.

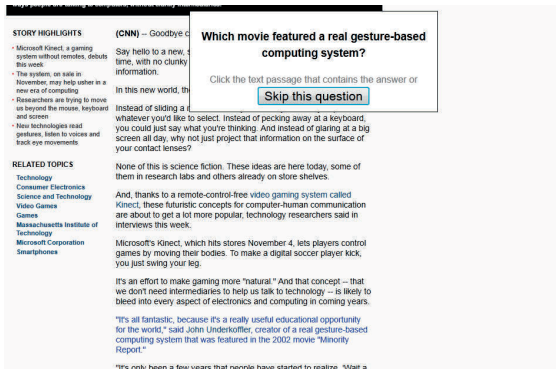
We registered 53 viewing contexts for the design and evaluate tasks combined. Screen sizes ranged from 1200 to 2560 pixels for the width and from 768 to 1920 pixels for the height.



(a) Design task



(b) Compare task



(c) Evaluate task

Figure 5: Showing the design task, the compare task for rating and the evaluate task for testing the layouts.

16 participants browsed in fullscreen and the rest in window mode (using less than 95% of the screen). From the 48 participants (91%) that had a high-resolution, wide-format display, 77% browsed in widescreen mode. Despite the small number of participants browsing in fullscreen, the median window size[10] was still 1417x912 pixels and the majority of users viewed the web site at a resolution of 1680x1050 pixels. This

[10]For the few users that resized the window during the task, we used the final window size as the basis for our analysis.

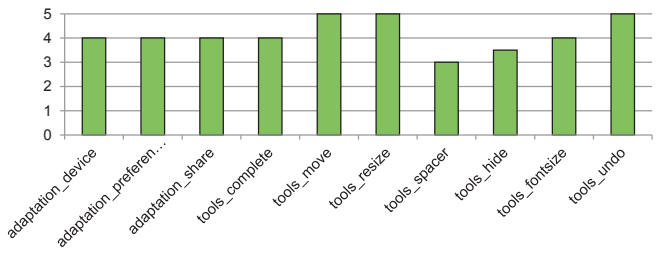
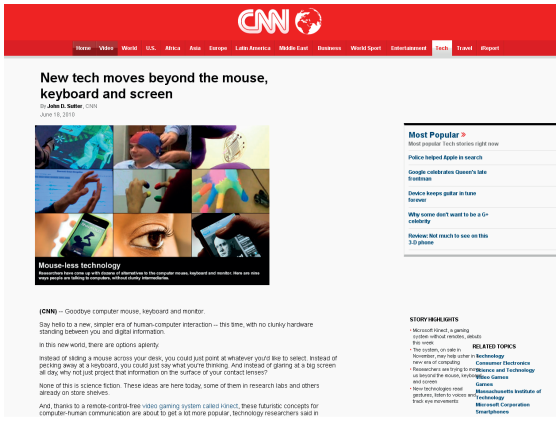


Figure 6: Median ratings of design tools (1=worst, 5=best)

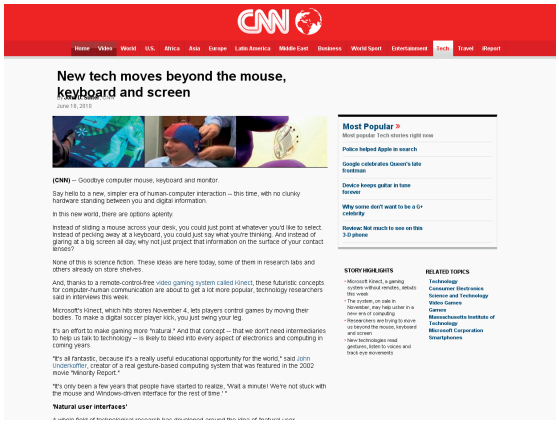
means that the "average" viewing condition was significantly larger than the "standard" resolution of 1024x768 pixels assumed by the original design of the web site.

Design task As shown in Figure 6, the design feedback showed a very positive assessment of CrowdAdapt's features when rating questionnaire statements concerning the design tools and support for sharing (consistent median of 4 on a 5-point Likert scale; 1 = *strongly disagree*, 5 = *strongly agree*). Participants therefore felt they were able to adapt the layout to fit their viewing conditions (*adaptation_device*) and preferences (*adaptation_preferences*). In addition, they felt that the new design should be the default layout for their viewing conditions and could also be better for other users (*adaptation_share*). Overall, the set of adaptation operations seemed well-suited to the adaptation scenario. Participants expressed that the tools allowed them to perform most of the changes they wanted (*tools_complete*) and the usefulness of each individual tool was rated positively. Participants most appreciated the move and resize tools (*tools_move* and *tools_resize*) and liked the tool for changing the font size (*tools_fontsize*). There was a neutral rating for the spacer operation (*tools_spacer*) which was also not used so much although essential for smaller optimisations concerning the spacing between nested DOM elements. The moderate rating for the hide tool (*tools_hide*) is not surprising given the experiment focus on large-screen adaptation. Finally, participants welcomed the undo tool (*tools_undo*). In particular, participants from the widescreen group, noted the absence of the multi-column operation we removed due to browser compatibility issues ("A function to create new text columns would be useful"). There were no notable differences otherwise.

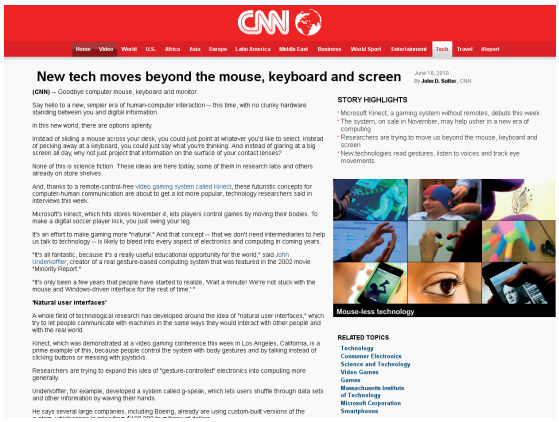
The set of 28 adaptations we received spanned 8 different screen sizes. The median window size was 1345x945 (min = 1107x579, max = 1903x1071) and the majority of participants (16) designed for widescreen contexts at a median resolution of 1920x1080 (min = 1280x800, max = 2560x1600), which was not anticipated by the original design. In the first parallel design phase, we could see a variety of adaptations of the original layout of which we show a selection in Figure 7. The vast majority of changes concerned the story image and article text. On average, the image was reduced to 84% of its original size of 640x436 pixels, while the original text width of 434 pixels was in all widescreen settings increased, on average by 168%. As a result, the image was often scaled down and in one case even cropped by resizing it to show only part of the original picture for which we present an example in Figure 7b. Four adaptations were completely text-oriented, in two cases even discarding the image. Moreover, the original



(a) Rough redesign of the layout with minor design problems



(b) Similar to Figure 7a, but the image was cropped and resized and sidebar elements better aligned



(c) Iteration of crowdsourced layout in Figure 1b showing minor changes to secondary elements to optimise space usage

Figure 7: Selection of crowdsourced layouts for widescreen (screenshots scaled at target window width)

font size of 14 pixels was increased in half of the adaptations designed for widescreen settings.

In terms of the position of elements, the sidebar content, originally aligned left of the article text, was moved to the right

of the screen in 74% of all contributed layouts. One participant also moved the image to the right which resulted in a lot more visible text on the first screen. By the end of our experiment, this contribution was the highest-ranked crowdsourced layout overall (Figure 1b). Also worth mentioning is the fact that 30% of crowdsourced layouts aligned the container with the entire content on the left-hand side of the screen, which is in contrast to the original centred layout.

Seven contributions showed minor design problems in terms of overlapping page elements. Figure 7a is a concrete example of this as it shows smaller issues in the bottom-right corner. However, some of the designs were also quite similar and so it happened that, similar to Figure 7b, other crowdsourced layouts provided an alternative with no design problems. As expected, participants designing based on a matching crowdsourced layout usually did not change the core layout aspects, but focused on finer details such as the position and size of secondary elements to win extra space for the article text. For example, Figure 7c shows an incremental adaptation that was created based on the best-ranked, best-matching crowdsourced layout from the first phase (Figure 1b). We can see that the participant who customised based on this layout could afford to pay more attention to detail as they repositioned the author and date of the article in the page and emphasised the story highlights by moving it on top of other elements in the sidebar and increasing the font size. Generally, we could observe the trend that designing based on a crowdsourced layout overall afforded less changes by participants.

Compare task In terms of ratings, we registered 143 direct comparisons until the end of our study period. Layouts with design problems received almost no votes and were therefore eliminated without administrative effort. The "best" crowdsourced layout won 41 times against all other layouts including the original design. The original design, on the other hand, was preferred 67 times to all user contributions. While not all crowdsourced adaptations were considered an improvement, crowd-generated layouts achieved best results for larger screen contexts that were poorly supported by the original design. There was little improvement over the original design for the smaller resolutions for which it was created, but this does not mean that crowdsourced versions were inferior. The overall trend in the widescreen group was positive and the ratings generally in favour of crowd-generated layouts, especially in the second stage where iterative design was allowed and achieved potentially higher quality of results.

We will now focus on the highest-ranked crowdsourced layout for the rest of the discussion. The ratio between this custom layout and the original design was 3 to 4, indicating the promising result that crowdsourced adaptation was considered an improvement in 43% of direct comparisons.

Evaluate task The positive trend also showed in the reading times and feedback we collected on the original and crowdsourced layout. Overall, there was a 25% performance gain for the crowdsourced layout, where reading was on average 40 seconds faster compared to the original layout. The majority of participants found it easy to read the text using either layout (mode 4 on a 5-point Likert scale from 1 = *strongly disagree*, 5 = *strongly agree*) and the crowdsourced layout received higher ratings in terms of a comfortable reading experience (mode 4 as opposed to 2). The fact that users did not feel efficient with either layout (mode 2) is surprising given the improvement in task performance, but could still be explained by some of the design critiques both layouts received. For example, participants from the widescreen group commented on the original layout that "reading the text with big white margins to the left and right of it that are broader than the text itself is not so comfortable". On the other hand, the crowdsourced layout was generally received quite positively (e.g. "Matches the expected layout of a newspage"), but also received critiques such as "The whole text seemed very long to read, could have organized better to view everything in a single page instead of scrolling down".

In summary, the results of our experiment suggest that crowdsourced adaptation based on a system like CrowdAdapt has the potential to improve the browsing experience and could be appreciated. The overall feedback on the tool support and the idea of sharing customised layouts was very positive. In line with other experiments [12], the quality of crowd-generated layouts seemed generally better in the second, iterative rather than the parallel, design mode. Layouts with minor design problems and radically different contributions, such as the ones focusing on the text only, were generally outvoted, which is good as they may also not be in the interest of web developers. While an administrative component could be added to the approach to give more control to web developers, or trusted users, so that their approval or rejection have significant impact on the ranking of crowdsourced adaptations, a review phase based on the one implemented in the evaluate task might be sufficient for step-wise roll-out and a pilot-and-push workflow even without such a component.

DISCUSSION AND RELATED WORK

This paper has investigated the idea of crowdsourced adaptation with the focus on the technical tools. A major premise of the techniques developed as part of this work is that they integrate well with common web architecture and widespread technologies. In consequence, a significant amount of our work went into developing methods and tools that are compatible with state-of-the-art web development practices so as to increase compatibility with existing application code and reduce the manual work required of developers in order to build on our solution. CrowdAdapt's operations were carefully engineered to handle possible edge cases by building on web standards and training on different sites. However, our goal was not to aim for completely automatic solutions. Rather, we argue that the control should always remain with web developers. We make use of crowdsourcing primarily as a tool for developers to receive design feedback and elicit context-specific design requirements for many different viewing conditions and individual user preferences. Using crowdsourcing for these aspects is viable, as it keeps developers informed about new requirements as they emerge.

Unlike much of the research on crowdsourcing [9], our focus is *not* on paid crowd work via Amazon Mechanical Turk. Instead, we build on frequent visitors and the chance that es-

pecially power users could have a particular interest in customising the layout for their own use context. By making the new layout available to other users, one user can benefit directly from another user's customisations. Therefore, the amount of pages a new user needs to customise can be substantially reduced. As we promote sharing and reuse within the web site community, our approach is designed to directly benefit the same group of users that also contribute, which was shown to raise motivation to participate [17].

Compared to previous approaches, we see the following three advantages for crowdsourced adaptation as far as the underlying techniques are concerned.

First, in contrast to previous model-based approaches, our approach does not require or impose a certain web design method such as WebML [6] and also no specific models of the user interface [5]. Since our adaptation techniques build directly on the final web interface, it is not relevant how the interface was developed as long as it can be represented in HTML and rendered in web browsers. This also means that no interface generation or transformation processes are involved. Interface generation usually starts from a model, then using a rule-based approach or machine learning to build a representation from which the user interface can be generated. Typical examples of interface generation approaches are PUC [15] and Supple [7]. On the other hand, interface transformation refers to approaches where existing representations of a user interface are mapped to models from which a new interface can be generated. Interface transformation therefore usually consists of three steps: reverse engineering [1], model transformation [11] and interface generation. This is described in detail for the CAMELEON reference framework [5] which formed the basis of many model-based user interface approaches such as MARIA [16]. In contrast to interface generation and transformation, crowdsourced adaptation directly starts from an existing final interface rather than a model or more abstract representation, and also does not require any intermediate representations. As a consequence, there is no need for reverse engineering and other model transformation techniques.

Second, our approach is tailored to the adaptation of web interfaces, whereas the majority of approaches such as PUC, Supple and MARIA aim to be comprehensive and handle web interfaces just like another output channel. However, such general approaches to adaptive user interfaces usually require proprietary user interface descriptions at a more abstract level. As a consequence, different user interface description languages have been proposed that are, however, not native to the web and therefore increase the threshold for being picked up by web developers. We therefore argue that our focus on the adaptivity of web interfaces is not a limitation, but rather an advantage since it means that we can directly build on and complement, rather than replace, established web standards and technologies.

Third, while previous adaptive interface techniques required complex constraint solving algorithms and additional server-side infrastructure to support web environments [18], we leverage native browser support and, in particular, new features of HTML5 and CSS3 that make it possible that much of the adaptation process can now run directly on the client-side. This client-side adaptation approach therefore also requires less technical overhead compared to previous, commonly proxy-based, approaches.

From the initial experiments with CrowdAdapt, we learned that the sharing of customised layouts between users in similar contexts generally makes sense and could be appreciated. However, we need to point out several limitations to the experiment and our current implementation of the approach.

First, the question of whether crowd-generated layouts are "good" deserves special attention. A short, but fair, answer is: "it depends". Many customised layouts met the preferences of other users, but could still be criticised from a professional user interface design perspective. Important is that users must have felt that their customisations improved their viewing situation and that even our simple in-built voting system and the pilot testing among participants eliminated "bad" layouts in the majority of cases. The results also indicate that the iterative task design similar to [12] and task splitting using a variant of the well-established *find-fix-verify* pattern [2]—for our purposes with a combined *find-fix* phase in the design task and a two-step *verify* stage in the compare-evaluate cycle for minor and major testing of crowdsourced adaptations—helped improve the quality of results. Additional means might be required to handle cases in which the crowd agrees on a new, potentially better, layout, that was, however, not requested by a frequent visitor and might even contradict how some users are used to working with the site.

Moreover, there is an interesting interplay between personalisation and adaptation to device that involves several factors which were difficult to isolate in the experiment. Clearly, the preferred adaptation at a particular moment may no longer be preferred later on, as this depends on many factors including the task at hand. Our experiment was driven by the task of improving the CNN layout for readability. Personalisation was therefore not the main goal, although it is still a factor in end-user customisation. An important aspect of our online experiment was to see the variety of use contexts, especially in large-display settings, and how they impact the use of CrowdAdapt. We have considered a controlled lab study modelled on top of our experiment to reassess some of the findings. While this is likely to increase the validity of results, it will require significant resources to emulate the different viewing conditions and might only provide little new insight. We therefore accepted the necessary tradeoffs, but paid great attention to the experiment design. Note that, as CrowdAdapt targets spatial adaptations to better fit different screen sizes, it may be combined with other customisation tools that usually target personalisation of design and content (colours, language, etc.) rather than adaptation to device.

Finally, our use of CrowdAdapt explored how crowd-generated layouts could be used "as is". The focus was on looking at how both developers and end-users could potentially benefit from using basic forms of crowdsourced adaptation, but the features and workflow could be extended to further improve crowd work. Ideally, one would want to

allow end-users to apply several fixes that may come from multiple crowd-generated layouts. While this could also be the result of a longer iterative design process, we could develop additional tools for directly picking different aspects from layouts and merging the underlying adaptations into a new template. This is feasible since, in our approach, adaptations are managed for each interface component based on jQuery selectors and could be applied selectively rather than as a whole. In our experiments, the ID-based adaptation approach worked fairly well for reusing adaptations across different news pages, wikis and blogs. However, we are aware of more advanced DOM structure-based similarity measures used in systems such as PageTailor [3], which could also be added to the approach. It could also be interesting to use data mining in CrowdAdapt, e.g., for supporting users when they customise the interface similar to [4] by suggesting changes done by other users in matching contexts, and also to learn a model of common adaptations. This could be done similar to [10], but our particular goal would then be to produce design patterns for different web site genres and use contexts.

CONCLUSION

This paper explored a crowdsourcing approach to achieving a higher degree of adaptivity of existing web interfaces and supporting a variety of device settings and user preferences. Future work could be based on the technical tools presented in this paper and take the ideas further. For example, an interesting direction could be to specifically design for certain groups of users and to take cultural aspects as well as special needs into account. CrowdAdapt could be refined to support this by adjusting the operations and extending the context model with more user-related aspects. Another direction could expand on the ideas of community-based design and allow end-users, not only to adjust the layout, but to get more creative and also provide additional content and functionality. The fact that our tools allow for the integration with existing web interfaces as well as different deployment modes could enable also other researchers to experiment with such ideas.

Acknowledgements

We would like to thank Rob Miller and Fabio Paternò for their helpful comments on earlier versions of this paper. This work was supported by the Swiss NSF (grant nr. 200020_134983).

REFERENCES

1. Bellucci, F., Ghiani, G., Paternò, F., and Porta, C. Automatic Reverse Engineering of Interactive Dynamic Web Applications to Support Adaptation across Platforms. In *Proc. IUI* (2012).
2. Bernstein, M. S., Little, G., Miller, R. C., Hartmann, B., Ackerman, M. S., Karger, D. R., Crowell, D., and Panovich, K. Soylent: A Word Processor with a Crowd Inside. In *Proc. UIST* (2010).
3. Bila, N., Ronda, T., Mohomed, I., Truong, K. N., and de Lara, E. PageTailor: Reusable End-User Customization for the Mobile Web. In *Proc. MobiSys* (2007).
4. Bunt, A., Conati, C., and McGrenere, J. Supporting Interface Customization using a Mixed-Initiative Approach. In *Proc. IUI* (2007).
5. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., and Vanderdonckt, J. A Unifying Reference Framework for Multi-Target User Interfaces. *IWC 15* (2003).
6. Ceri, S., Daniel, F., Matera, M., and Facca, F. M. Model-driven Development of Context-Aware Web Applications. *TOIT 7*, 1 (2007).
7. Gajos, K. Z., Wobbrock, J. O., and Weld, D. S. Improving the Performance of Motor-Impaired Users with Automatically-Generated, Ability-Based Interfaces. In *Proc. CHI* (2008).
8. Hattori, G., Hoashi, K., Matsumoto, K., and Sugaya, F. Robust Web Page Segmentation for Mobile Terminal Using Content-Distances and Page Layout Information. In *Proc. WWW* (2007).
9. Kittur, A., Nickerson, J. V., Bernstein, M. S., Gerber, E., Shaw, A. D., Zimmerman, J., Lease, M., and Horton, J. The Future of Crowd Work. In *Proc. CSCW* (2013).
10. Kumar, R., Talton, J. O., Ahmad, S., and Klemmer, S. R. Bricolage: Example-Based Retargeting for Web Design. In *Proc. CHI* (2011).
11. Limbourg, Q., and Vanderdonckt, J. Multipath Transformational Development of User Interfaces with Graph Transformations. In *Proc. HCSE*. Springer, 2009.
12. Little, G., Chilton, L. B., Goldman, M., and Miller, R. C. Exploring Iterative and Parallel Human Computation Processes. In *Proc. CHI Extended Abstracts, HCOMP Workshop* (2010).
13. Nebeling, M., Matulic, F., and Norrie, M. C. Metrics for the Evaluation of News Site Content Layout in Large-Screen Contexts. In *Proc. CHI* (2011).
14. Nebeling, M., and Norrie, M. C. Tools and Architectural Support for Crowdsourced Adaptation of Web Interfaces. In *Proc. ICWE* (2011).
15. Nichols, J., Chau, D. H., and Myers, B. A. Demonstrating the Viability of Automatically Generated User Interfaces. In *Proc. CHI* (2007).
16. Paternò, F., Santoro, C., and Spano, L. MARIA: A Universal, Declarative, Multiple Abstraction-Level Language for Service-Oriented Applications in Ubiquitous Environments. *TOCHI 16*, 4 (2009).
17. Rashid, A. M., Ling, K. S., Tassone, R. D., Resnick, P., Kraut, R. E., and Riedl, J. Motivating Participation by Displaying the Value of Contribution. In *Proc. CHI* (2006).
18. Schrier, E., Dontcheva, M., Jacobs, C., Wade, G., and Salesin, D. Adaptive Layout for Dynamically Aggregated Documents. In *Proc. IUI* (2008).