Exploring Smartphone-based Web User Interfaces for Appliances

Katie Derthick*[#] **James Scott*** **Nicolas Villar*** **Christian Winkler*[†]**

*Microsoft Research
United Kingdom
{jws, nvillar}@microsoft.com

[#]University of Washington
USA
derthick@uw.edu

[†]University of Ulm
Germany
christian.winkler@uni-ulm.de

ABSTRACT

We describe the SAWUI architecture by which smartphones can easily show user interfaces for nearby appliances, with no modification or pre-installation of software on the phone, no reliance on cloud services or networking infrastructure, and modest additional hardware in the appliance. In contrast to appliances' physical user interfaces, which are often as simple as buttons, icons and LEDs, SAWUIs leverage smartphones' powerful UI hardware to provide personalized, self-explanatory, adaptive, and localized UIs.

To explore the opportunities created by SAWUIs, we conducted a study asking designers to redesign two appliances to include SAWUIs. Task characteristics including frequency, proximity, and complexity were used in deciding whether to place functionality on the physical UI, the SAWUI, or both. Furthermore, results illustrate how, in addition to support for accomplishing tasks, SAWUIs have the potential to enrich human experiences around appliances by increasing user autonomy and supporting better integration of appliances into users' social and personal lives.

Author Keywords

Smartphones, appliances, user interfaces.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms

Human Factors

INTRODUCTION

On appliances such as coffee machines, washing machines, etc., user interfaces (UIs) are constrained by various important factors [2], particularly the need to minimize cost of the physical UI hardware. It is typical to find appliance UIs that have physical buttons or knobs for input, and LEDs or segment-based LCDs for output. This limits the UIs of such appliances, often obscuring or even leaving out all but basic functionality.

As an illustrative example, the home cappuccino machine illustrated in Figure 1 (c) is a capable machine retailing for 500 USD, but has just 7 buttons on its UI. One button, for "latte," serves up a particular amount of coffee and steamed milk when pressed. However, by manipulating the same button in an obscure procedure that is non-obvious without reading the manual, these default amounts can be changed. In fact, 20 distinct functions can be accessed using those 7 buttons (e.g., set water hardness, cleaning functions, etc.), so all but 7 functions involve using the buttons in a way that goes beyond just "press," and are therefore somewhat obscure. In addition, there are other settings that the machine internally exercises control over which are not exposed on the UI at all, such as the temperatures and pressures used in the coffee-making process. Nor can the machine detect which user is in front of it and offer a customized experience, e.g., providing their preferred latte proportions or even using their preferred language. In fact, this machine has no writing on it, using icons to represent drink types, which is common for many appliances to avoid having to localize them for each country they are sold in.

Yet, many users of such coffee machines have smartphones

	Appliance UI	Smartphone UI
Cost of UI hardware	$tens	$hundreds
Processor	8 or 16-bit, kHz or low MHz	32-bit, GHz, with GPU
User input	Physical controls (e.g., buttons)	Multitouch, sensors (accelerometer, camera, etc.)
User output	LEDs, perhaps a low-res screen	High-res color screen, audio, vibration, etc.
Help	Separate user manual	Self-explanatory UIs possible with images/video/audio
Localization	Often icons-only to minimize cost	User's preferred language
Adaptiveness	Static	Highly customized, personalized, context-sensitive

Table 1. Comparison of appliance and smartphone UIs

MobileHCI '13, August 27 - 30 2013, Munich, Germany

http://dx.doi.org/10.1145/2493190.2493239

with them or nearby [5]. These have much more powerful and general purpose UI hardware, centered on a high-resolution touchscreen with a processor and GPU. If only the appliances could present a UI directly through a user's smartphone, they could provide a more helpful, adaptive and customized UI. Table 1 highlights the differences between typical UIs found in appliances and smartphones. In this paper, we refer to smartphones due to their high user penetration, but actually all the discussion is relevant to tablets, notebooks, or PCs, if those are "to hand" as well.

By augmenting appliances with modest additional hardware, we present a mechanism by which appliances easily "take over" nearby smartphones in order to present more powerful UIs than their physical controls provide, with modest additional cost. By incorporating a WiFi chipset supporting "access point" (AP) mode, an appliance can be easily found by a nearby smartphone user by simply choosing that AP on the WiFi settings page, which causes the appliance's web UI to be displayed automatically, allowing the appliance to take advantage of the powerful smartphone hardware. We refer to such Smartphone-based Appliance Web User Interfaces as SAWUIs.

We envision that SAWUIs go beyond simply supporting task-oriented user interfaces (e.g., substituting for a "latte" button), but also open the door to new opportunities for appliances to integrate into and support everyday life. To explore these opportunities, we conducted a study with 13 user experience and industrial designers, who were asked to revisit the interface design of two appliances in the light of SAWUI's potential for new interaction experiences.

This paper contributes:

1) An architecture for smartphone-based appliance web UIs (SAWUIs) by which users can connect a smartphone immediately to a nearby appliance and allow the appliance to present its UI using the phone, with no modification to the phone, no reliance on cloud services or networking infrastructure, and only modest additional hardware in the appliance.
2) A prototype implementation of SAWUIs and example appliances: a coffee machine and a mood light, showing how SAWUIs enable features such as personalization and the use of language.
3) A study of the opportunities created by SAWUIs by having 13 user experience and industrial designers redesign appliance interactions using this paradigm.

RELATED WORK

The idea of using mobile devices to access and control physical appliances with embedded web servers has been around for some time [1, 10].

Much of the previous work in remote appliance interfaces has focussed on the concept of *universality*. Nichols, et al., for example, have looked at formal descriptions of appliance/application UIs (e.g., menu hierarchies) so that this universal description can be transferred to a remote device such as a PDA, where the interface can be automatically generated [14]. Nichols, et al., also explored how to improve automatic UI design using a set of design requirements [15], and how to maintain UI consistency between multiple devices [16]. Others have looked at enabling universal connectivity between appliances and control devices [12].

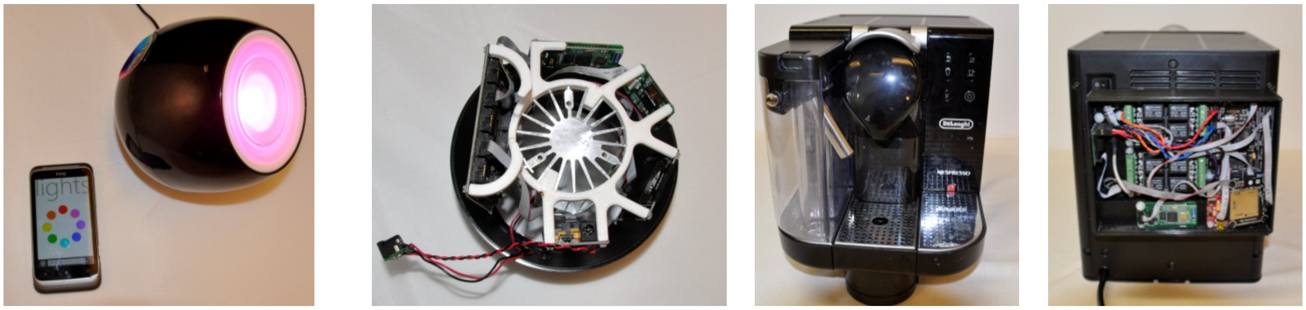
Our work does not address enabling universality; we make the assumptions that nowadays WiFi, HTML, and AJAX are prevalent. These standard technologies allow appliances to directly send interactive UIs to any smartphone in an ad-hoc fashion, without requiring an "app" installation or going through a hub device. We focus instead on how the affordances of physical UIs and SAWUIs complement one another, looking at how the design of individual appliances' UIs can take advantage of smartphones.

Perhaps the closest previous work, Roduner, et al., [19] used Java MIDlet-based phone apps to control tasks on a dishwasher, coffee maker, printer, and radio, and conducted a user study with 23 participants who compared the appliance interface and phone-based interface. Participants generally liked the phone-based interface and completed "exceptional" tasks quicker, but the phone was less useful for "everyday" tasks. The MIDlet approach, however, does not handle association issues ("How do I control *this* machine?") and requires a specific app to be installed for each device. Nichols, et al., [17] compared completion times of users using automatically-generated UIs versus standard appliance UIs to accomplish a set of tasks, finding that it took users around half as long to complete tasks with the auto-generated UI. While task-focused studies are important, we believe the opportunities presented by SAWUIs extend beyond accomplishing tasks quicker, to support of users' everyday life practices. We therefore conducted a more exploratory, design-focused study of the user experience opportunities created by SAWUIs.

Kranz, Holleis, and Schmidt [11] discuss a set of related design implications for devices in the Internet of Things, including: provide information when and where it's useful, support information provision without explicit interaction, overprovision, consider specialized components, consider visibility of controls, and more. We return to [11] in the Discussion.

Establishing associations between devices has been widely studied, including by Kindberg, et al., [10] using beacons (e.g. infrared) and tags (e.g. RFID, visual), and by Holmquist, et al., with accelerometers [8]. In our work, we use an access point in appliances as this enables existing WiFi-capable devices to easily associate with appliances.

An interesting analogy can be found in "cyberforaging" [20] in which mobile devices make use of situated resources (e.g., screens). In the extreme case, a user may only need to carry a "Personal Server" [23] to access any device. In our work we turn this around: appliances are



(a) Augmented mood light and a phone showing its UI

(b) Inside of mood light showing AP module (top) and other prototyping electronics

(c) Augmented coffee machine

(d) Rear of coffee machine showing prototyping electronics

Figure 1. Prototype appliances supporting phone web UIs (SAWUIs). Much of the extra electronics could be eliminated in production devices. SAWUIs employ HTML, CSS & JavaScript served by a webserver and WiFi access point in the appliance.

effectively foraging for more powerful smartphones to augment their limited UI hardware.

There are a number of existing commercial systems that allow appliances to be interacted with through smartphones. Home automation standards like X10, Insteon and Z-Wave can be used with phone apps; however, the level of control available is typically fairly coarse-gained, limited to functions such as turning devices on or off through remotely-controllable AC power strips. The SAWUI concept is designed to be built into appliances, thus allowing fine-grained control of all appliance features. There is also a rich body of research into smart homes and how their occupants relate to them, e.g. [7, 3]. While SAWUIs may eventually be integrated into smart homes, in this paper we focus on the design space from an appliance-centric viewpoint that includes public appliances as well.

Some existing appliances have associated phone apps, including the Sky+ digital TV recorder (my.sky.com/mysky/makethemost/sky-plus), and Philips Hue light bulbs (www.meethue.com). However, these devices require pre-installation of apps to work, and/or connectivity to cloud services. SAWUIs supports ad-hoc interaction with any smartphone by using a WiFi AP and web-based UI.

Dey, et al., [5] found that people in their study are within arm's reach of their phones 53% of the time, and in the same room 88% of the time. This is good news for the feasibility of SAWUI to be usefully deployed.

SMARTPHONE-BASED APPLIANCE WEB USER INTERFACE (SAWUI)

In the previous section, we discussed a number of existing ways in which appliances can support smartphone-based UIs. Many of these rely on users pre-installing "apps" on phones, precluding walk-up-and-use scenarios and restricting the range of phones supported.

In contrast, our proposed architecture makes use of an HTTP server in the appliance itself, serving a web UI (HTML5, JavaScript, CSS) on demand. This has the advantage of working with any smartphone with a web browser (e.g. any Windows Phone, Android or iPhone), thus allowing any smartphone-carrying user to walk up and easily use the SAWUI.

To network between the phone and the appliance, we use standard 802.11 WiFi networking, with the appliance either providing an access point (AP), or using a pre-existing shared AP, or using both modes simultaneously

Appliance provides a WiFi access point

We augment appliances with WiFi chipset configured as an access point (AP), and a lightweight web server and DNS server. By responding to DNS and HTTP requests similarly to a "captive portal" (i.e., airport or hotel WiFi hotspots for which phones automatically bring up a login page when a WiFi connection is made), we cause phones joining the AP to visit the webpage with the appliance's UI. After choosing the AP, e.g. "Coffee Machine," on a phone's WiFi settings page (see Figure 2), no further user interaction is necessary before the SAWUI appears in just a few seconds in our prototype implementation. Presenting appliances as WiFi APs means that no pre-existing shared network relationship is required between appliance and phone. For example, guests at a home or visitors at a public place might not have any network credentials for that place, but they can nonetheless control appliances, just as they could by using the appliance's physical UI. By using WiFi instead of custom networks such as 802.15.4, Insteon, or Z-Wave, we ensure compatibility with commodity smartphones. By using the captive portal to automatically bring up the UI, we avoid requiring the user to input any further identifier for the appliance, as choosing the AP selects the appliance. Since phones also typically order the WiFi access points found by signal strength, even when many AP-enabled appliances are present, the nearby devices are easy to find on the AP list.

Security Issues

The use of an AP has advantages in providing a generic and simple way of accessing SAWUIs. This is particularly true if using “open” APs without security, since after the AP is selected on a phone, the SAWUI comes up without any further inputs. However, this raises security issues: the appliance may be accessed by anyone within WiFi range, and the system may be susceptible to man-in-the-middle attacks whereby appliances can be impersonated to gather inputs from the user. If the SAWUI includes functionality like “Post on Facebook,” then these inputs may include personal information such as Facebook IDs or credentials.

One way around this is to use a secured network with a secret key, and display that key visibly on the appliance. If this was done using a QR code to encode the WiFi AP details and credentials [24], the key would not have to be manually typed in and the overhead to the user would be minimized. Geofencing techniques [21] can also be used to ensure that only nearby devices have access to the AP network.

Using an existing WiFi access point

An appliance can also use WiFi in client mode, connected to an existing network. In this way, the user’s phone does not have to switch WiFi networks to use the appliance, and can connect to remote devices rather than just those that are within direct signal range. To find the appliance UI, users can simply type in a name to their browser such as http://coffee/ which the appliance has registered using DNS, or use other discovery mechanisms such as uPnP. This URL could be bookmarked for future direct access.

Another advantage of using an existing AP is that the appliance can get both local network and internet access through it, and use this to provide services such as remote control, integration with other internet-connected services like social networks, and integration with house-wide applications, through an architecture like HomeOS [6].

There are two main disadvantages to using an existing WiFi network rather than an AP in the appliance. One is that “guest” use is not enabled since the phone must be on an existing network. Another is that getting an appliance onto a shared network in the first place is a problem, since credentials need to be provided to the appliance somehow.

Appliance APs, Existing APs, or Both?

Appliances with an AP can more easily be connected to by anyone. However, one downside is that phones connecting to the appliance cannot get WiFi internet access (though they can still use cellular Internet access). Appliances with a client WiFi interface do not have that issue, however they require an existing infrastructure AP that users have credentials for. To provide the best of both worlds, one compelling way forward is for appliances to use both a WiFi AP and client at the same time. This could be done using a single network interface, as shown by VirtualWiFi [4, 18], or simply by incorporating two WiFi devices

In an example use case, appliances in a home might be accessed by family members through the home’s AP, while visitors to the home could use the individual appliances’ Aps for “guest mode” access.

Hardware cost

The required additions to appliance hardware (a better microcontroller, more memory/storage, a WiFi interface) are modest. For the microcontroller and memory, most appliances already incorporate such elements and Moore’s Law means that their cost is dropping all the time. Somewhat counterintuitively, the server (the appliance) can actually be less capable than the client (the phone browser) in terms of processing power, memory, etc., since from the server’s point of view the HTML5, Javascript, CSS, etc. files are just text strings to be sent verbatim; the interpretation of those files as code or markup only happens on the client. Already, inexpensive 8-bit microcontrollers such as the PIC can support TCP/IP, DNS and HTTP [13].

With respect to the WiFi and the use of AP mode, WiFi chipsets capable of operating as APs as well as clients are available for as little as $7.80 [9].

On the other hand, by adding this modest hardware, the appliance gets to make use of the much more powerful hardware capabilities of smartphones (see Table 1). Furthermore, by using SAWUIs, some of the elements of the physical UI might be removed, which can reduce the cost of augmenting an appliance, or even make it cheaper! As a case in point, the Living Colors mood light in Fig. 1(a) has physical controls on the light itself but also comes with a dedicated remote control, a separate unit which requires a radio link to the main unit, all of which could be replaced by a SAWUI. Furthermore, the SAWUI may provide additional features, such as timer control, which is not provided by the current remote control.

EXAMPLE APPLIANCES

To build proofs of concept appliances with SAWUIs, we took two existing off-the-shelf appliances and modified them to add SAWUIs: a DeLonghi Lattissima coffee machine and a Philips Living Colors mood light. We used mostly off-the-shelf Microsoft .NET Gadgeteer [22] hardware components, with the exceptions of the WiFi AP interface, for which we designed and built a new .NET Gadgeteer-compatible WiFi AP module based on the ConnectOne Nano WiReach WiFi module, and another custom module to control the Living Colors light.

Both prototypes used the FEZ Hydra mainboard, SD card module and USB Client DP module from GHI Electronics, and our custom AP module. For the coffee machine, we added an extra box on the back; for the mood light, we used a 3D printed rig to hold the components (Figure 1).

The coffee machine was controlled by soldering leads onto the internal circuit board where the buttons were attached, and using two Relay modules (from Seeed Studio) to

Figure 2. To access a SAWUI, a user chooses the appliance from the WiFi settings screen (left) of their phone. The SAWUI automatically appears with no further user input. We prototyped a SAWUI-controlled coffee machine (middle) and mood light (right). The phone's language setting is used automatically. Cookies store personal settings persistently.

programmatically press the buttons. We used Seeed Studio current sensors on the coffee pump and the milk valve so we could determine when the coffee and milk were pouring.

For the mood light, we used a custom module which mimicked the protocol used by the Living Colors remote receiver to communicate with the light controller itself, using an ARM Cortex M0 processor.

NEXT STEP: DESIGN EXERCISE

Having prototyped SAWUIs and built example appliances, we wished to explore the potential design and user experience opportunities created by SAWUIs. As the first step, we chose to conduct a series of design exercises with user experience and industrial designers, to see how they made use of SAWUIs in the redesign of existing appliances.

We chose to work with user experience and industrial designers because they are trained to view technology as flexible and imagine how technology can affect peoples' everyday lives, and as well as to communicate their ideas verbally and visually.

We wish to emphasize that our design exercise is but one element of a series of evaluations necessary to fully understand SAWUIs; we do not claim that this exploratory study gives us the same perspective on real use as "in the wild" studies. However, it makes sense as the first step to do a design-led study because it provides valuable insights on the kind of broader experiences that this technology allows us to offer users—over and above potential optimizations for the time taken to perform tasks.

To ground the technological capabilities of SAWUIs for our participants, we identified four key features of SAWUIs to highlight based on our bottom-up understanding of the technology's capabilities. We used the exact wording below to communicate these features to our participants. We will revisit these features in the Discussion section.

Product Customization. The function of the appliance can be customized to the person using it. For example, the coffee machine remembers your previous setting using a cookie, and brings that up as default. In contrast, physical UIs give the same result no matter who pushes the buttons.

Localization. The appliance can use the language of the person using it, both for the initial UI presentation, and, importantly, for feedback. For example, when the coffee machine is out of water, it can say, "Out of water" in Spanish if your phone language preference is set accordingly. In contrast, physical UIs often use abstract icons to avoid the cost of making a different faceplate for each geography. Feedback is also abstract (e.g., short blinks or long blinks of an LED).

Instructions built in. The phone-based interface can be more self-explanatory, using pop-outs and rich media. For example, the coffee machine UI can include a video or checklist of the cleaning procedure and remind you to undertake it at the right times. In contrast, physical UIs often come with instruction manuals—despite the designer trying to make the appliance UIs self-explanatory.

Adaptive UI. The phone-based interface UI can change depending on whether the user is (e.g.) novice or expert. For example, the coffee machine UI could have a "novice" mode and an "expert" mode which includes more settings for steam and boiler pressure settings, etc. In contrast, physical UIs have a set number of buttons/functions. Of course, some can be included behind a flap or around the side, but that makes for a clumsier experience.

DESIGN EXERCISE METHODOLOGY

The study included 13 participants, 6 female and 7 male. 10 participants were user experience designers at a large software company and 3 were in the process of earning a PhD in industrial or interaction design. All participants were working or had worked in a job using product design skills. The study was piloted with 2 designers, whose results are not included.

The study occurred in three phases: (1) a demo of SAWUIs; (2) two design tasks; and (3) a follow-up interview.

In the first phase, we demonstrated the capabilities of SAWUIs over a video chat service. Each participant was shown a smartphone connecting to a coffee machine via WiFi, adjusting the quantity of milk and coffee via the SAWUI, and brewing a customized cup of coffee via the SAWUI. Participants received a document outlining each feature described in the previous section—customization, localization, instructions built in, and adaptive UI; each feature was described in detail to the participant. The demo served to ground the second phase of the study.

In the second phase, participants were asked to redesign two appliances—a coffee machine and a washing machine—incorporating SAWUIs where useful. We explained to participants that we were interested in the interaction paradigm, not coffee machines and washing machines per se. We chose to include a washing machine rather than the light fixture described earlier since the light fixture was too simple; we felt we would learn more from redesigns of appliances with both simple and complex functions. Participants had access to PDF versions of instruction manuals for each appliance.

To guide their designs, participants were given worksheets listing tasks supported by each appliance. Participants were asked to indicate whether they would locate the UI features required to complete the task on the appliance, the smartphone, or both, as shown in Table 2.

Task	Appliance	Smartphone	Both	Why?
Brewing	***(tickbox)***	***(tickbox)***	***(tickbox)***	***(text)***

Table 2. Excerpt of a tasks worksheet for the design exercise.

The "Why?" column was included to help participants remember the reasons for their design decisions during the follow-up interview, rather than as data to be analysed specifically.

Since participants were not directly interacting with either appliance, the worksheets served to help participants comprehend the range of functionality supported by the appliances. However, to accomplish the study goal of exploring the larger opportunities created by this interaction paradigm, participants were also asked to sketch designs that included new appliance UIs (if they wished) and SAWUIs. We emphasized to participants the importance of designing the holistic system, rather than simply adding a SAWUI; in other words, participants were asked to consider how the whole system of use—appliance and smartphone—is affected by the introduction of this interaction paradigm. Including the table as well as the sketching tasks helped participants ground their designs in real functionality, while at the same time, explore possibilities for the holistic system.

Participants were asked to design under cost constraints similar to the original appliance designs: participants were told how the addition of the SAWUI adds sophisticated technology without much cost, and that they could make changes to the appliance UI using hardware comparable in price to the original hardware UI buttons, so that the appliances would remain within a similar price range.

In the third phase, participants were interviewed via voice chat and asked to walk through their worksheets, task by task, and sketches of their holistic system redesigns. Participants were asked to explain why they made the decisions they made on both the worksheets (tasks) and in their sketches (holistic system). Participants were asked for general reflections on the interaction paradigm.

Each interview was audio recorded, transcribed and summarized in a memo. Two researchers open coded the memos for themes, then over a series of meetings discussed and agreed upon the main themes. These main themes were used to guide closed coding of transcripts, which resulted in memos on each theme. The themes are presented in the next section, and include representative quotations and sketches.

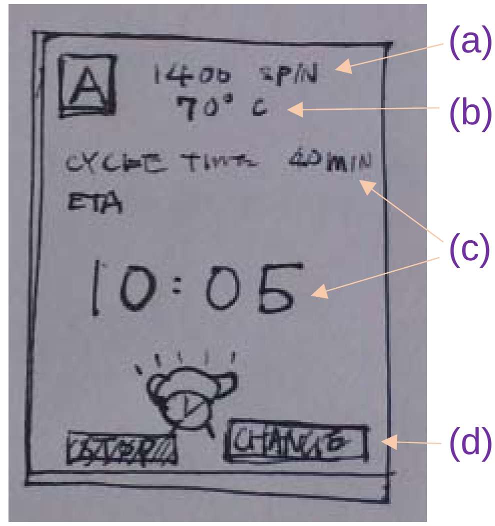


Figure 3. A SAWUI screen from P13's design that includes real-time feedback (c), informative feedback (a), information about the consequences of choosing a setting (b), and the ability to change a setting within a sequence (d).

DESIGN EXERCISE RESULTS

The study design produced a count of UI locations—appliance, smartphone, or both—chosen for each task, sketches and interview data. The count of UI locations and participants' explanations why they chose that UI location provide insight into criteria guiding design decisions for specific types of tasks, while the sketches and participants' explanations of them provide insight into possibilities created by the SAWUI paradigm for the holistic system, for UIs, and for people's broader life experiences.

Worksheet Results: UI Locations, Design Criteria, and Simplifying UIs

The results of the worksheet for the coffee machine design are shown in Figure 3, which includes the percentage of participants (x-axis) who located each task (y-axis) on the appliance UI, SAWUI, or both (bars). These quantitative results are not reported for their statistical significance or generalizability, but to indicate general patterns in participants' design choices. We include the results of the coffee machine worksheet only, partially because of constraints on space, but also because the results are representative of both worksheets. Furthermore, it is not the specific appliance or tasks we are interested in, but participants' decisions and criteria.

As shown in Figure 3, participants chose more often to offer tasks on the SAWUI alone (50%) than the physical UI alone (14%), and often used both (37%). There was little consistency between functions/within participants, but more consistency within functions/between participants. For example, 100% of participants located the UI features to accomplish the function of "Changing water hardness" on the SAWUI, and locating "Setting quantity for

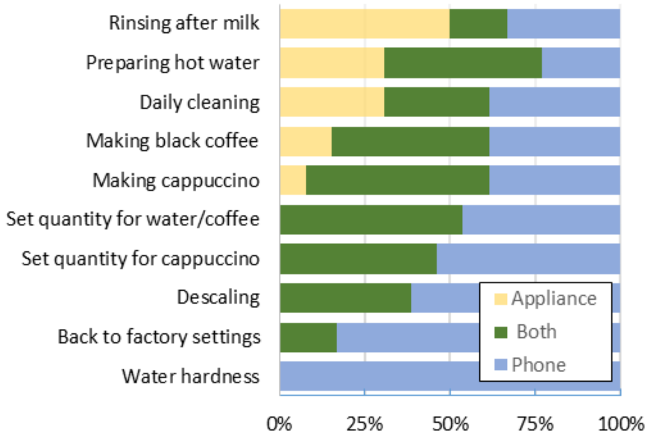


Figure 4. Percentage of participants locating UI features to accomplish tasks on the appliance UI, SAWUI, or both.

water/coffee" was consistently split between solely on the SAWUI or on both UIs.

The decision varied task by task, but all participants referenced at least two of the following criteria when explaining their decisions to locate a particular task on the appliance UI, SAWUI, or both:

- Frequency: Is this task completed frequently or infrequently? Is this a primary (frequent) function?
- Proximity: Does this task require the user to be at the appliance to accomplish? Is it possible for the user to accomplish this task remotely?
- Complexity: Is this task very complex or very simple to accomplish?

Not all participants applied these principles in the same ways, but all participants used these as criteria when making design decisions. For example, some participants referenced *frequency* as the reason for locating a task on the SAWUI, while others used it as the reason for locating a task on the appliance. All participants considered *proximity*, though they had different ideas about whether tasks could or could not be accomplished remotely. For example, P11 located "Rinsing after milk" on the SAWUI because she felt rinsing was a button-press-only task, and so could be accomplished remotely via the SAWUI, whereas P02 located it only on the appliance UI because the milk container has to be removed before rinsing, therefore requiring the user to be at the appliance. In general, if the participant felt the task was very *complex* to accomplish, he or she would locate it on the SAWUI.

As discussed, participants most often located UI features to accomplish tasks on the SAWUI only or on both the appliance UI and SAWUI (Figure 3). Participants frequently described the decision to remove certain interactions from the appliance UI in terms of "simplifying" the interaction for users. Simplifying was described by participants in two ways: the "hiding" of infrequently-used functions from the appliance UI (and locating them on the SAWUI), and the "discovery" of complex or difficult-to-accomplish tasks on the SAWUI. As P02 explained, *Yeah, so hide the things away but at the same time, you take it out and suddenly as a user, you get easier access to functionality which was hidden before.*

Sketches and Interview Results: System Transparency, Beyond Transparency, and Sharing outside the System

More than accomplishing tasks, participants saw the potential of SAWUIs to increase system transparency, move beyond system transparency, and support sharing outside the system. These themes are not mutually exclusive; aspects of each can be seen to support the others.

While we instructed our participants to design within a similar cost range of the original appliance, participants nevertheless saw possibilities in this interaction paradigm that would require additions to the technology described in this paper. We return to this point in the Discussion section.

Appliance Transparency: Real-time Feedback, Informative Feedback, and Changes in a Sequence

By using two-way communication between appliance and SAWUI, participants increased the transparency of the appliance by providing users real-time feedback, informative feedback, and the ability to change settings within a sequence.

Real-time feedback is common on appliances, such as a slowly-turning knob that indicates what phase a washing machine is in. Not all appliances, however—the coffee machine used in our study included—provide this type of feedback. All participants included real-time feedback in their SAWUI designs, such as P13's "Cycle time 40 min," shown in Figure 4 (c). P13 enriched real-time feedback further by including the exact time of day the cycle will finish.

Participants also included an addition to real-time feedback that is more difficult for typical appliances to provide: feedback on specific information regarding the current system state. P06 explained:

Having different types of blinking lights I think is pretty poor feedback to what it is exactly that you're doing. […]The washing machine gives "Error 20." Okay, whatever. It's kind of nice if you have feedback like, "This [error] is actually the water pipe and it's not connected right and this is how you change it," because apparently it knows, so it can give you much more information.

Blinking lights and obscure codes are uninformative feedback for acknowledging user input and for errors. As P06 explained, the SAWUI allows the system to inform the user of its state using specific, informative language, rather than LEDs or codes. Information on system state doesn't have to be negative, of course, and could include information such as when a fresh pot of coffee is ready, as P12 suggested, or the current phase of a wash cycle, as shown in Figure 4 (a).

Finally, participants designed ways for users to make changes at any given stage of a sequence. Figure 4 (d) shows how users can change settings without having to stop the current cycle, whereas the current washing machine design requires users to stop the cycle, walk through the sequence stages until they get to the desired phase, and then make a change. Furthermore, this change can be made remotely.

Beyond Appliance Transparency: Non-procedural Information and Appliance Personality

Moving beyond appliance transparency, participants included non-procedural information and appliance personality. Non-procedural information took two forms: information on the consequences of choosing a setting, and educational information.

Information on the consequences of choosing a setting can be illustrated by an example from P09, whose design was strongly influenced by her passion for environmental sustainability. Her design included information on how much water a certain load size would require and how much electricity a certain spin speed consumes. Likewise, the sketch in Figure 4 (b) includes the water temperature of the current setting.

P12 provided educational non-procedural information, for example, about differences between types of coffee drinks—not all users know the difference between a latte or a cappuccino, she explained, herself included. By using the smartphone to scan labels, P12 also included in her design the option for users to learn more about the origins of the coffee, and other participants, like P02, included the option to acquire information about the origins of fabric while doing laundry. P12 explained that providing this non-procedural information to users helps them *feel a little bit more educated but in a low-cost, kind of serendipitous way.* Participants included ways to help users acquire more information about what they are consuming, moving beyond appliance transparency, to increased transparency of the entire system of which consumption is a part.

Likewise, participants saw the potential for SAWUIs to add a bit of personality to the appliance. P12 said it clearly:

I think with the UI on the mobile device like it could really be much more playful; it could almost be like having—almost characterizing or creating a caricature of the device, giving it personality so that the person like looks forward to engaging with it; […] it's really just procedural, right, but now it becomes more emotional, it becomes more fun.

Participants explained how the more informative feedback discussed in the previous section could potentially come in the form of a playful message—*"Go drink your coffee" or "Yay, you're done!"* (P04)—or a more pleasant medium, such as P08's decision to replace a washing machine's buzzer with a ringtone. Figure 5 shows a sketch from P03, who capitalized on the smartphone's ability to remember

Figure 5. A sketch from P03 showing the SAWUI remembering a user and welcoming her back—with personality!

settings, greeting a user with, "Welcome back, Jane!" Beyond appliance transparency, personality adds dimension to users' experiences with an appliance.

Sharing outside the Appliance: Settings, Social Activity, and Other Applications

As well as designing features directly related to the appliance, participants capitalized on the smartphone's ability to share information, both between users, such as sharing settings or information about social activity, and with other applications.

Participants included the sharing of settings in their designs. Shared settings typically took the form of "recipes," whether for a type of coffee drink or a program for the washing machine. Participants imagined sharing settings supporting social interactions, such as being able to take a friend a drink brewed using his individual recipe, or being able to save others' settings in your system. Participants described how sharing settings could also potentially make accomplishing household tasks easier, such as in this personal example from P04:

My partner does the laundry most of time and every now and then I'd go and have to do it and can never really remember the right buttons 'cause I just don't do the laundry very often. But if we have a smartphone app customized to the laundry where all you have to do is push the button and it does all the settings, that's cool. [...] So we're sharing a little bit that responsibility, [...] being able to sort of take tasks that one person in the household is usually more responsible for and condense it into a couple of clicks, "Wow that's cool."

Participants also designed ways for users to share social information. For example, multiple participants described users receiving discreet notifications, such as a small icon appearing on the phone screen or a social network site, when their friends at work get coffee. Participants also provided ways for users to, for example, invite others to join them for coffee by sending invitations using in-phone applications such as their contacts list.

Finally, participants frequently included interaction with applications outside the appliance-plus-smartphone system in their designs. While the social interactions discussed previously might require information shared via the internet, participants also explicitly included interaction with other applications in their designs, such as having the number of times a user brews a cup of coffee sync with health or other tracking apps he or she uses. P05, for example, suggested the possibility of a user's game character energizing each time the user brews a cup of coffee. Participants' decisions to sync appliance-use information across applications points to opportunities for SAWUIs, and users, outside the appliance.

By creating designs that support system transparency, features that go beyond system transparency, and sharing outside the system, our participants revealed how SAWUIs allow much more than simply accomplishing tasks quicker.

DISCUSSION

In this section, we discuss the implications of our findings for users and for the SAWUI architecture.

Implications for Users

Results of the study with designers revealed the potential for SAWUIs to affect far more than time-on-task. As user experience designers, our participants are trained and practiced not only in designing technology for ease of use, but in imaging how technology can affect people's everyday lives. Therefore, the themes that emerged in participants' designs serve to support what P12 eloquently described as *the human experience—or maybe another way of putting it is, not the experience you have with the device as much as the experiences that happen around the device*.

Improved system transparency in the form of real-time and informative feedback and the ability to make changes within a sequence point to increased *user autonomy*. If users are not constrained by the limited information provided by an appliance (or by the appliance's location), they gain more control over their direct manipulation of the appliance and of their own space and time.

Moving beyond system transparency by providing non-procedural information and appliance personality provide an *enriched* and *enriching* experience with the appliance. Increased visibility into the larger system of which a user's consumption is a part—via the availability of non-procedural information—points to a more informed user, who is better able to make educated decisions about usage and consumption in a broader context. An appliance with more personality also enriches the user's experience to potentially become a more *emotional* or *meaningful* one. The idea of "personality" also points to the potential for different "appliance profiles," such as the eco-friendly appliance. Considering home appliances, the ability of SAWUIs to support appliance profiles could support the "shared usage model" described by Brush and Inkpen [3].

Finally, the ability to support sharing with other users and applications indicates the potential for SAWUIs to support the human experience through better *integration* of appliance use into users' *social* and *personal lives*.

Our findings suggest implications for SAWUIs or other mobile devices that extend appliances that echo and extend the guidelines in [11]. We suggest designers: consider the frequency, proximity and complexity of a task when deciding where to locate UI features to accomplish it; utilize mobile devices' ability to provide additional information, including real-time and informative feedback and non-procedural information; consider including multiple profiles or user access levels; and support the sharing of settings, social activity, and information from other applications.

Implications for the SAWUI Architecture

Recall that the technical side of this paper's contributions is a new way for users to quickly obtain a web-based user interface for an appliance using any smartphone. We now reflect on how the broad feedback of the designers in our study relates to this concrete technology, and to next steps for technology development for SAWUIs.

During the first phase of our study, we described four properties of SAWUIs compared to physical UIs: product customization, localization, instructions built in, and adaptive UIs. In the designs generated, all four of these features were present, supporting our preconceptions that these were useful aspects of SAWUIs for appliance design. For product customization, while we had envisioned that appliances may use cookies to store preferences on individual phones, this does not facilitate sharing between users. Thus, persistent preferences stored on the appliance be a better approach as it enables sharing, though this raises important privacy questions. Participants did not explicitly mention localization and adaptive UIs, but their use of language and their ideas for themed UIs (e.g., an eco-friendly UI) spoke to these features.

In addition, participants mentioned other ideas that our initial architecture does not directly support. A number of participants' designs required the SAWUI to interact with other features of the phone, e.g., the calendar, camera or alarm; apps on the phone, such as health tracking apps; and web-based apps, including ordering things online, social networks. Counterintuitively, the latter category is most easily feasible with our original architecture, since a JavaScript app on the smartphone can communicate with other websites. Other interactions could be achieved using apps on phones, which the SAWUI could trigger, e.g. through linking to "ics" files for calendar entries.

All participants included remote use which is enabled by our architecture except in the configuration where the appliance only has an AP. However, asynchronous interfaces—where appliances notify users after an event occurs like laundry is finished—are not easily achievable using the SAWUI architecture presented, since webpages are not easily able to set up background or ongoing tasks To enable this case, the initial web-based connection could be used to kick off installation of an app.

CONCLUSIONS

This paper described the SAWUI architecture by which smartphones can easily show user interfaces for nearby appliances, with no modification or pre-installation of software on the phone, no reliance on cloud services or networking infrastructure, and modest additional hardware in the appliance. SAWUIs can provide personalized, self-explanatory, adaptive and localized UIs, in contrast to appliances' physical UIs which are constrained by many factors such as cost [2].

To explore the opportunities created by SAWUIs, we conducted a study asking designers to revisit the design of two appliances. We found that task characteristics including frequency, proximity and complexity were used in deciding whether to place functionality on the physical UI, the SAWUI, or both. Furthermore, the results of the study showed us how, in addition to support for accomplishing tasks, SAWUIs have the potential to enable richer human experiences around appliances by increasing user autonomy and supporting better integration of appliances into users' social and personal lives.

Further validation of the SAWUI concept could involve implementing redesigns of appliances drawing on these findings, and studying their usability and user experiences both in lab settings and "in the wild".

ACKNOWLEDGEMENTS

We are grateful to John Helmes and Steve Hodges for their input, and thank our participants for their time and ideas.

REFERENCES

1. Borriello, G. and Want, R. Embedded computation meets the World Wide Web. *Commun. ACM 43, 5, 2000*, ACM.
2. Brouwer-Janse, M.D., Bennett, R.W., Endo, T., van Nes, F.L., Strubbe, J.J., Gentner, D.R. Interfaces for Consumer Products: "How to Camouflage the Computer?" *CHI 1992*, ACM.
3. Brush, A.J.B, and Inkpen, K.M. Yours, mine, and ours: Sharing and use of technology in domestic environments. *UbiComp 2007*, ACM.
4. Chandra, R., Bahl, P. and Bahl, P. MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card. *Infocom 2004*, IEEE.
5. Dey, A.K., Wac, K., Ferreira, D., Tassini, K., Hong, J.-H., Ramos, J. Getting closer: an empirical investigation of the proximity of users to their smart phones. *UbiComp 2011*.
6. Dixon, C., Mahajan, R., Agarwal, S., Brush, A.J., Lee, B., Saroiu, S. and Bahl, V. An Operating System for the Home. *NSDI 2012*, USENIX.
7. Harper, R. (Ed.). Inside the Smart Home. 2003, Springer.
8. Holmquist, L.E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., Gellersen, H-W. Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts. *UbiComp 2001*, Springer.
9. iSuppli iPhone 4 Teardown, news article. Retrieved August 2012 from iSuppli at http://www.isuppli.com/Teardowns/News/Pages/iPhone-4-Carries-Bill-of-Materials-of-187-51-According-to-iSuppli.aspx
10. Kindberg, T., et al. People, Places, Things: Web Presence for the Real World. MONET 7,5 2002, Kluwer.
11. Kranz, M., Holleis, P, Schmidt, A. Embedded interaction: Interacting with the internet of things. *IEEE Internet Computing*, 14(2), 46-53.
12. LaPlant, B., Trewin, S., Zimmermann, G., Vanderheiden, G. The universal remote console: A universal access bus for pervasive computing. *Pervasive Computing*, 2004.
13. Microchip TCP/IP Stack. Application Note AN833, Microchip.
14. Nichols, J., Myers, B.A., Higgins, M., Hughes, J., Harris, T.K., Rosenfeld, R., Pignol, M. Generating remote control interfaces for complex appliances. *UIST '02*. ACM.
15. Nichols, J., Myers, B.A., Harris, T.K., Rosenfeld, R., Shriver, S., Higgins, M., Hughes, J. Requirements for automatically generating multi-modal interfaces for complex appliances. *IEEE Conference on Multimodal Interfaces*, 2002.
16. Nichols, J., Myers, B.A., Rothrock, B. UNIFORM: Automatically generating consistent remote control user interfaces. *CHI '06*, ACM.
17. Nichols, J., Myers, B.A. Studying the Use of Handhelds to Control Smart Appliances. *Proc. ICDCSW '03*. IEEE.
18. About the Wireless Hosted Network. Retrieved August 2012 from MSDN at http://msdn.microsoft.com/en-us/library/windows/desktop/dd815243.aspx
19. Roduner, C., Langheinrich, M., Floerkemeier, C., Schwarzentrub, B. Operating appliances with mobile phones—Strengths and limits of a universal interaction device. Pervasive 2007, Springer.
20. Satyanarayanan, M. Pervasive Computing: Vision and Challenges. *IEEE Personal Communications* 2001
21. Sheth, A., Seshan, S. and Wetherall, D. Geofencing: confining 802.11 coverage areas to physical boundaries. *Pervasive 2009*, Springer.
22. Villar, N., Scott, J., Hodges, S., Hammil, K., Miller, C. .NET Gadgeteer: A platform for custom devices. *Pervasive 2012*, Springer.
23. Want, R., Pering, T., Danneels, G., Kumar, M., Sundar, M., Light, J. The Personal Server: Changing the Way We Think about Ubiquitous Computing. *UbiComp 2002*, Springer.
24. WiFi Joiner App for Android. Retrieved August 2012 from http://vickypedia.me/wifijoiner