Designing an Ambient Interaction Model for Mobile Computing

Jonas Elslander and Katsumi Tanaka

Department of Social Informatics
Graduate School of Informatics, Kyoto University, Japan
{jonas,tanaka}@dl.kuis.kyoto-u.ac.jp

Abstract. In this paper, we advocate and propose a new interaction model for mobile computing by positioning ambient notifications central to both the user experience and the operating system interface design process. We suggest a model that visually replaces applications as current first digital citizens in mobile operating systems by a modular stream based notification center. In order to do so, we define the general layers that make up the dynamics of the current as well as the proposed mobile computing experience. We conclude and demonstrate the benefits and areas of improvement of our newly proposed paradigm: an ambient mobile interaction model.

Keywords: Methodology, design, HCI, big data, application, interface, interaction, notification, ambient, push, pull, social, context, discovery.

1 Introduction

1.1 User needs

Present-day mobile operating systems make use of a centralized notification structure in an archaic way by heavily relying on push notifications for nearly all communication [1]. When analyzing the findings of previous research and related works supporting this claim, we realized the potential of significantly improving the whole mobile computing experience by using notifications as a central entity to the interaction model. This had led us to research, design and propose a multi-layered ambient interaction model based on five distinguishable user needs, explicated below. It is important to note that we define the term *notification* as any form of informative on-way communication initiated by the system and directed towards the user.

Context. The contextual environment of the user should allow the operating system's interface to more effectively focus on what's most relevant to the user at any given moment. Notifications imply relevance and are a naturally adequate vehicle therefore.

Social. The social graph of a user greatly influences the perception of consumed mobile information [1], but has yet to be implemented on a system-wide level. The distinction

N. Streitz and P. Markopoulos (Eds.): DAPI 2014, LNCS 8530, pp. 3–14, 2014.

of three social classes[1] and usage of sensorial data can be incorporated in order to achieve this goal.

Visual. The human mind is wired to effortlessly distinguish colors and details and to read emotions and situations. Subsequently, mobile operating systems' interfaces are suggested to use photographic pictures – rather than icons or text – whenever possible as their primary visualization technique for notifications.

Ambient. Currently push notifications are used by the system when information is considered important to the user. Widgets are being offered as an alternative to manually retrieving information from within applications for less urgent communication. As we learned, these opposite styles leave a lot of ground uncovered. An ambient approach can solve the gap between both extremes by presenting information in the peripheral attention of the user, creating a non-invasive way to stay informed.

Time. Introducing the concept of time as a system-wide visual dimension to structure information fosters an improved discovery compared to present-day rigid grid-based interfaces.

1.2 Trends

In addition to the above-mentioned user needs, there are several relevant trends changing the mobile computing landscape. We find ourselves at the advent of wearable computing, with smart watches and smart glasses being the first emerging device categories. These require a significantly different interface and interaction approach, as failed attempts to implement the current model have shown to hamper adoption break-trough. Screen independence is another long-term trend that has recently begun to change mobile computing by allowing data to be stored in 'the cloud'. However, screen independence can be taken much further than the ability to access he same data from various devices based on user identification. Our proposed model provides the opportunity to develop mobile operating systems founded on these broadened trends, as covered further in this publication.

1.3 Paper Structure

The remainder of this paper is structured as follows. Section 2 explicates our conclusive interaction model for current mobile operating systems. All layers and their joint relations are discussed. In section 3, we propose an ambient mobile interaction paradigm by highlighting the differences with the present-day interaction model and elucidating new concepts. In section 4, we conclude our findings and summarize the opportunities supported by the proposed ambient interaction model.

---

[1] 'Favorites', 'friends' and 'others' each account for a different user perception. [1].

2 Current Interaction Model for Mobile Computing

In order to design an interaction model both solving the user needs and facilitating the trend development discussed in section 1, we first synthesized the specifics of the current interaction model for mobile computing on smartphones and tablets. The results are visualized in Figure 1 and will be clarified in this section.

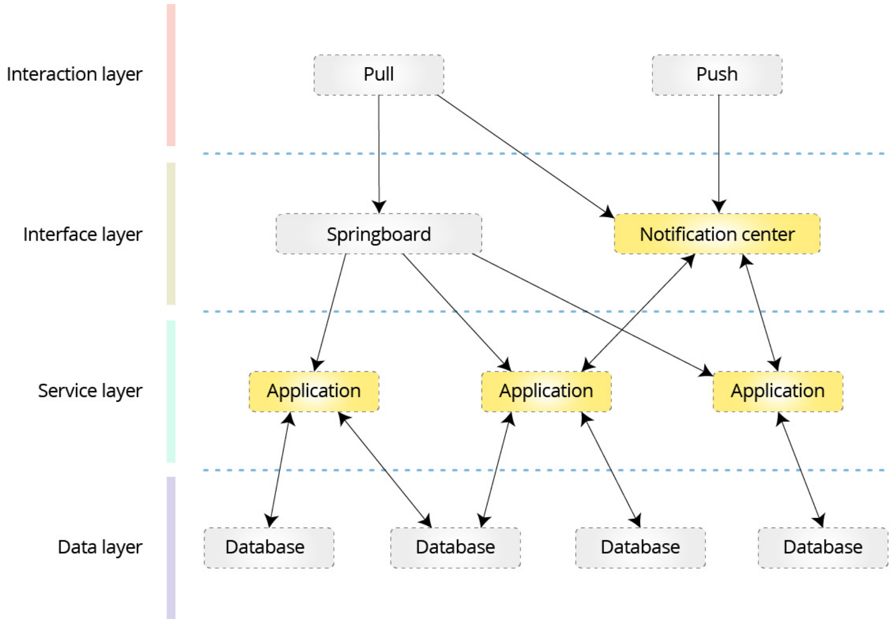

Fig. 1. Current four-layered interaction model

In the model we define four distinct layers: data, service, interface and interaction. The entities in the lower two layers are application-specific while the objects in the upper two layers are regarded as system-specific. This figure represents the information flow towards the user, with the visual endpoints being highlighted in yellow. Subsequently, we clarify the interpretation and scope of each individual layer, as well as their joint edges.

2.1 Data Layer

The data layer encompasses all means of data storage, both locally (i.e. on the mobile device) and externally as a connected database. The data layer bidirectionally communicates with the service layer (see next subsection) by offering and receiving information in the shape of structured data. It is the changing nature of the joint relationship between these two layers that has in great part shaped the interaction model evolution for desktop computing over the past decades.

2.2 Service Layer

We define the service layer as the tier concerned with selecting what information to present to the user as a consistent whole. Presently, an application-centric interaction model characterizes mobile computing. Data sources serve connected applications – or apps – that form the primary haven for users to retrieve information. The desktop computing environment, once recognized by a similar paradigm, has since evolved to a document-centric model because of technical limitations. Mobile apps – just like websites – serve a specific purpose and perform this task independent of other entities in the service layer: a weather application will update you on the forecast while a GPS app might give you the fastest route to a destination, but getting navigated to the closest-by sunny destination requires a lot of user interaction and cognitive participation.

2.3 Interface Layer

When discussing the entities of the interface layer, we refer to the general interface environments offered by the mobile operating system: the springboard and the notification center. Both present the user access to information in an entirely different way – respectively grid-based icons versus chronological snippets – and are characterized by different interactions (see next subsection). Individual applications are represented by their own interfaces, often striking similarities within their application category, but these are located outside of the scope of our system-focused model.

2.4 Interaction Layer

The concept of pull or push applies to all user-driven processes in the current mobile interaction model. Respectively, the user either manually retrieves information or gets interrupted by it while performing any other activity. We specifically define both concepts as follows:

$$F_{pull} = A \wedge B$$

$$F_{push} = \neg A \wedge \neg B$$

With *A* symbolizing the intent of information access[2] and *B* the presence of human-computer interaction, represented by a set of physical actions. Following this definition, pull-style communication requires both the user's intent of accessing certain information, as well a set of as bodily behavior towards this goal. Push-style communication on the other hand is not directly triggered by intent or by the user's actions. There exists a relation between the interface and interaction layer entities of the present-day interaction model. The springboard is designed to mainly serve as a pull-style navigational structure – with widgets and badges being the only exceptions to the rule – while the notification center has a mixed use. It provides both push-style notifications as well as a pull-style accessible chronological summary of missed information deemed relevant by the system.

[2] The choice of selecting, consuming, evaluating... specific information.

2.5 Edge Values

The application-specific interfaces and notification center interface serve as the end-points for information flow in our model. Upwards the data is filtered and presented while downwards it is accessed by the user. The uni or bidirectional arrows between the layers represent the edges and their flow orientation. Within every interlayer – the area between two adjacent layers in the model – the weight or value of all edges can be analyzed and compared. These are the topic of related and future work; the interaction-interface interlayer has already been partly covered by our previously referenced research into the use and perception of mobile notifications.

3 Ambient Interaction Model for Mobile Computing

With the user needs and trends – cited in the introductory section of this paper – in mind, we analyzed the current interaction model for mobile computing and allotted its shortcomings. In this section, we clarify those while suggesting improvements that make up the proposed ambient interaction model. A visualized overview is presented in Figure 2.

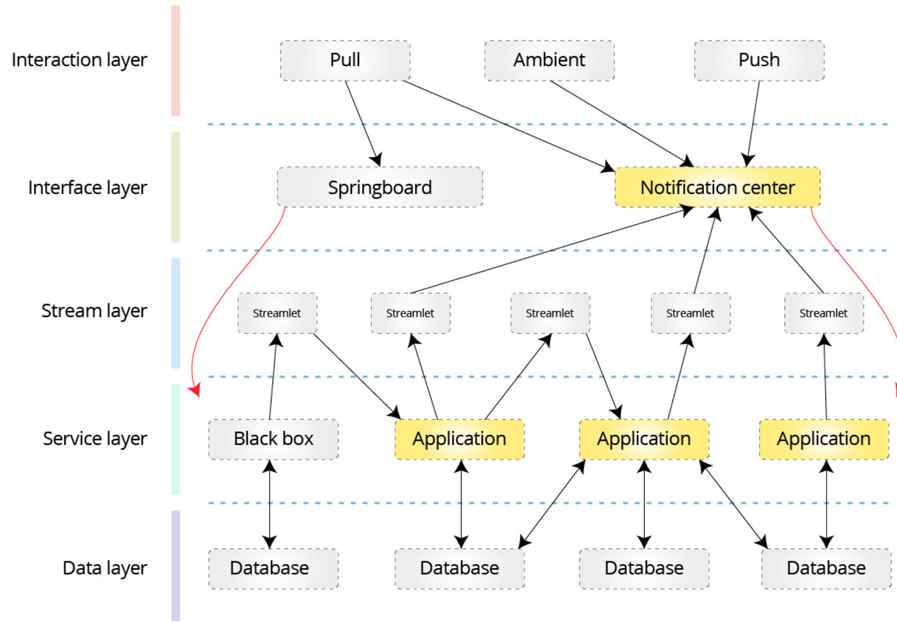

Fig. 2. Proposed five-layered interaction model

We define five distinct layers: data, service, stream, interface and interaction. The entities in the lower two layers are application-specific while those in the upper three layers are regarded as system-specific. As before, the figure represents the information flow towards the user, with the visual endpoints being highlighted. Below, we clarify the interpretation and scope of each individual layer, while focusing on the iconic distinctions with the present-day interaction model: modularity, notification-centricity and ambience.

3.1 Data and Service Layers

Whereas the changing dynamic between the data and service layer primarily fueled the structural changes in the desktop environment user experience (made up by the interaction and interface layers) over the past decades, we propose a different evolutionary approach for the mobile setting. Both application-specific layers and their mutual relationships remain chiefly unchanged, with the prime proposed hypotheses being linked to the three system-specific layers. Concerning the data layer, we envision local and connected information storage to support system services and applications in a similar way. As access to the right (amount of) data is of paramount importance as an asset to developing effective and successful mobile applications, the service layer remains an important pillar to the devised ambient interaction model. However, in subsection 3.4 we will introduce a new interface environment supporting our proposal to replace the application-centric paradigm with notification-centricity.

3.2 Stream Layer: Modularity

Novel to our interaction model is the inclusion of a notification stream, an intermediary layer that handles the chronological data flow between the applications in the service layer and the notification center in the interface layer. Preceding research conclusions have shown that the users of current-generation mobile operating systems perceive smartphone notifications as too many and of low importance/relevance. Providing the user with granular control over which applications can send notifications under certain conditions is not desirable either; computers are complicated systems and a *deus ex machina* experience is desirable for most mobile usage cases. The root of the problems surrounding the current notification-handling framework can be traced back to two concepts inherent to its design: same weight push notifications and a direct connection between every allowed application and the device's user.

Assigning the same weight to all notifications infers them being of same importance to the user while a direct connection, as shown in Figure 1, implies that the applications ultimately control what for (and thus how often) the user gets interrupted. The stream layer offers a solution to both issues while adding an array of new prospects. Based on the portlet concept, entities in the service layer – both applications and system processes – are required to publish their outgoing information into a chronological stream, a layer without a visual component, instead of directly to the notification center. This not only assures that the user doesn't get directly notified for every single event, as not all data will be relayed upwards in the model, but also changes the currently difficult balance in favor of (much) more shared data as the user experience trade-off obstacle has been eliminated. Further reaching are the following possibilities offered by the inclusion of the stream layer.

When the operating system is given access to the entirety of the stream layer data, mobile analytics and search take on new dimensions. Across applications, it becomes for example possible to compute the value of social connections or to determine usage patterns. Removing the barrier of user interruption simply allows for a bigger stream of data. This can be beneficial both as a means towards system personalization and to the user in a more direct way. Moreover, the general definition of search states that something has to be found. Far more often than the unknown, present-day mobile operating system-wide search focuses on retrieving known locally stored information and thus acts as a shortcut rather than as a true searching process. When certain information isn't ruled as immediately relevant to the user but still retained by the system, it can be retrieved later on by a search algorithm calculating the relevance to the changed user environment and/or search query.

Besides making each data package or *streamlet* accessible to the operating system, it can also be made available to other applications (with privacy and safety rules in place). This access renders the entire stream layer modular. Applications can not only learn from information shared by others, but also attach their own information and thus communicate and ameliorate data. The specifics of the stream layer format are the topic of future research.

An overhead algorithmical system process determines the multidimensional weight of every streamlet and pulls those scoring over a certain threshold up to the notification center in the interface layer. The aforementioned scoring dimensions are being determined by the *contextual*, *social* and *time relevance* of the data. This makes that the notifications are not shown to the user based solely on a timestamp match[3]. Instead, all streamlets are characterized by a *time to live* or TTL for reuse and notification purposes – beyond that point, they are archived for search and analytical purposes. At any moment within the TTL timeframe, notifications can be displayed in the interface layer when deemed contextually, socially or timely relevant. Examples of contextual relevance are the geographical location of the user[4], the time of day[5] or the environment as determined by local or connected sensors[6], while the social relevance is determined by the psychological and/or physical proximity of a connection referenced in the streamlet data. The time relevance is defined by sequence and pattern detection: when certain activities appear in relation to each other (relative) or to time (absolute), the streamlets of linked activities become important once one activity or moment is detected. The study and explicit definition of the scoring dimensions and the selection process comprises the topic of our future research.

As shown in Figure 2, certain streamlets can be deemed not important or content-rich enough for transfer to the notification center, but will be after data is added by another service layer entity. This creates opportunities for *black boxes* or applications without a proper interface. These upload selected data to the stream layer without

---

[3] Example of a timestamp match: if person A sends you a text message at 12:18PM, you will be notified of it by the operating system at 12:18PM.

[4] E.g. The position of the user or the change in position (static/dynamic).

[5] E.g. The precise time (07:33AM), morning/evening or holiday/day at work.

[6] E.g. Surrounding sound volume, light intensity, temperature or indoor/outdoor detection.

providing a front-end for the device user. As mobile applications can easily be bought from virtual stores, this could establish a mobile market for direct data monetization.

3.3 Interface Layer: Notification-Centricity

Structurally, the interface layers of the current and proposed interaction models – as presented in Figures 1 and 2 – look alike, sporting both a springboard and a notification center as entities. We advocate two major changes however: a redistribution of the interaction balance between both entities and a redesign of the visual embodiment of the notification center. It is not our vision to fundamentally change the representation of the springboard, but rather to replace it as the primary visual environment on mobile devices. With a notification-centric approach we want to place relevant information front and center, instead of a navigational structure guiding the user through a myriad of application icons. When the screen is activated, information accessed from the stream layer – as described in the previous subsection – is first off visualized in a graphic-rich style, as displayed by the mock-ups[7] in Figure 3.

Fig. 3. Mock-ups of a notification-centric interface

As indicated by the exclusion of any notifications in the first iterations of popular mobile operating systems iOS and Android, the current dedicated notification centers were more of an afterthought to an already established interaction model. We support our claim for placing a redesigned notification center paramount to the springboard by affirming that currently one in three smartphone users checks their phone for missed notifications at least once an hour [1], on top of attending to found ones and given the limited breadth of present notification centers and their contents, making it a critical interaction environment. In our prototypes, we incorporate solutions to all user needs explicated in the introductory section of this paper. As mock-ups, these are prevailingly interpretations for exemplary and research purpose. Photographic pictures of social connections and concepts known to the user (such as places or products) are used when available to create an interface environment suitable for ambient interaction (see next subsection) and small screens (see the subsection 3.5), but also to add

[7] More high-resolution interface mock-ups can be downloaded at `http://bit.ly/AIMinterfaces.`

emotions and agility to the communicated information. The contextual surroundings are the dominant prerequisite for inclusion of information into the notification center, a dynamic structure – in contrast to the underlying static springboard – that primarily changes based on the user's situation. All retained information is visually organized based on the social graph of the user, rather than on the timestamp of the notification or the category of the application it originated from. Research teaches us that mobile devices are primarily communication tools – hence the proposed social focus – and a logging[8] and discovery[9] tool second. The latter is supported by visually dividing the notification center into three time sections: the past, the present and the future. The past addresses all alerts, the present handles the user's contextual surroundings and the future relates to offered suggestions. By navigating between these, the user can change the balance between briefing and discovery on a three-point scale.

3.4 Interaction layer: ambience

In section 2.4 we explicated the concepts of push and pull in relation to mobile notifications. In our proposed model we introduce an intermediate option, ambient notifications, we define as follows:

$$F_{\text{ambient}} = A \land \neg B$$

Here, *A* symbolizes the intent of information access and *B* the presence of human-computer interaction, represented by a set of physical actions. Ambient notifications thus encompass the action of looking for information, but without the component of action taking. When information discreetly lives in the peripheral field of attention, it doesn't require a summoning action - differentiating it from pull-style access – nor the obtrusiveness of a push notification. In the table below, the interrelation between push, pull and ambient interaction is visualized.

Table 1. Truth table for interaction styles

	User action (B)	No user action (¬B)
User intent (A)	**Pull-style interaction**	**Ambient interaction**
No user intent (¬A)	*Unintentional interaction*	**Pull-style interaction**

For clarification, we use the analogical example of the communicated information regarding a political election. Searching or browsing the Internet for retrieving the views and stances of a certain party is considered pull-style interaction, while overhearing an outdoor speech on your way to work is labeled as push-style interaction. Changing focus to the radio news when the poll results are being discussed is an ambient interaction, while being called up on the night before by swing vote influencers is an unintentional interaction.

8 E.g. Note taking, picture taking.
9 E.g. Internet browsing, social networks, weather forecast.

As the stream layer establishes the groundwork for the reimagined interface layer, so is the latter designed to support ambient interaction. This theoretical concept translates well to communication by gaze. Only important (stream layer) pieces of visually rich (interface layer) information are displayed in the tangential ocular field of the user (interaction layer), culminating in a relevant and immediate information transfer at the user's convenience. For a select minority of notifications, such as incoming phone calls, alarms or calendar reminders, push-style communication is still preferred and supported by the ambient interaction model. As is pull-style access to information displayed by application-specific interfaces, accessible either through navigating the springboard or through interaction with an element in the notification center.

3.5 Trend Inclusion

Besides offering a solution for the shortcomings of current mobile operating systems, represented by the earlier established user needs, our proposed interaction model was devised with the wearable computing and screen independence trends in mind. We are at the advent of integrated and small-footprint mobile device adoption, such as smart watches and smart glasses, which require a new interaction model. Scaling down (a part of) the current interface to fit significantly smaller screens doesn't suffice as current attempts that failed to gain traction have proven. At the same time, the different usage scenarios require a new interaction approach. Smart watches and smart glasses are suitable for displaying information to the user as described in the previous subsections. Both device categories resonate with the ambient interaction concept: they are always on (unlike smartphones and tablets we like to put away when not in use), they're part of the peripheral vision sphere and comprise of a small display surface that requires select relevant information with a high communication transfer rate. In certain related works, wearable devices are referred to as ambient due to their characteristic of physically blending into the surroundings by merging technology with wearable goods, thus labeling their interfaces as ambient. In this paper however, we describe and propose methods for ambient interaction – rather than an ambient interface – between the user and a mobile device, either wearable or portable.

Users carrying numerous mobile devices also raise questions about the optimal usage of screen independent information. When considering this trend, we distinguish the independence of both data collection and display, by suggesting the stream layer of the ambient interaction model to be made internet-accessible (thus, by placing them in 'the cloud'). The systems and applications of multiple mobile devices can feed a common stream layer, from which all service layer entities across devices can access and remix data. The interface and interaction layers of each device are defined by the same interaction model, but can individually focus on content that best fits their usage and aspired user experience by adjusting the threshold parameters for notification center inclusion.

4 Conclusions and Future Work

In this research, we synthesized the information flow within current mobile operating systems and combined it with collected user needs and trends to determine its weaknesses and to define opportunities. Consequently, we proposed an adapted interaction model that introduces:

- A stream layer, enabling mobile data growth and monetization, reducing displeasure and managing all notifications in a modular, analyzable and searchable way;
- A notification-centric interface concept, transforming the notification center into a contextual, social and visual framework structured around time and with ambient usage in mind, while placing it chief to the system's static navigational interface;
- An ambient interaction paradigm that challenges the stasis between push and pull-style communication by heralding a new dynamic between the intent of information retrieval and performed user actions, offering possibilities for both existing and new mobile device categories.

All three topics require further research in order to refine their impact and define their contribution to the shift towards the next interaction model for mobile computing. Future work includes qualitative user studies of situational mock-ups based on the ambient interaction model, as well as a technical evaluation of the processes that construe the communication between the stream layer and its surrounding layers.

References

1. Elslander, J., Tanaka, K.: A Notification-Centric Mobile Interaction Survey and Framework. In: Jatowt, A., et al. (eds.) SocInfo 2013. LNCS, vol. 8238, pp. 443–456. Springer, Heidelberg (2013)
2. Scott McCrickard, D., Chewar, C.M.: Attuning notification design to user goals and attention costs. Commun. ACM 46(3), 67–72 (2003)
3. Maglio, P.P., Campbell, C.S.: Tradeoffs in displaying peripheral information. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2000), pp. 241–248. ACM, New York (2000)
4. Maglio, P.P., Barrett, R., Campbell, C.S., Selker, T.: SUITOR: An attentive information system. In: Proceedings of the 5th International Conference on Intelligent User Interfaces, IUI 2000 (2000)
5. Booker, J.E., Chewar, C.M., McCrickard, D.S.: Usability testing of notification interfaces: Are we focused on the best metrics? In: Proceedings of the 42nd Annual Southeast Regional Conference (ACM-SE 42), pp. 128–133. ACM, New York (2004)
6. Grudin, J.: Partitioning digital worlds: Focal and peripheral awareness in multiple monitor use. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2001), pp. 458–465. ACM, New York (2001)
7. Erickson, T., Kellogg, W.A.: Social translucence: An approach to designing systems that support social processes. ACM Trans. Comput.-Hum. Interact. 7(1), 59–83 (2000)
8. Saket, B., Prasojo, C., Huang, Y., Zhao, S.: Designing an effective vibration-based notification interface for mobile phones. In: Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW 2013), pp. 149–1504. ACM, New York (2013)
9. Hazlewood, W.R., Stolterman, E., Connelly, K.: Issues in evaluating ambient displays in the wild: two case studies. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2011), pp. 877–886. ACM, New York (2011)

10. Messeter, J., Molenaar, D.: Evaluating ambient displays in the wild: Highlighting social aspects of use in public settings. In: Proceedings of the Designing Interactive Systems Conference (DIS 2012), pp. 478–481. ACM, New York (2012)
11. Mankoff, J., Dey, A.K., Hsieh, G., Kientz, J., Lederer, S., Ames, M.: Heuristic evaluation of ambient displays. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2003), pp. 169–176. ACM, New York (2003)
12. Kim, T., Hong, H., Magerko, B.: Design requirements for ambient display that supports sustainable lifestyle. In: Proceedings of the 8th ACM Conference on Designing Interactive Systems (DIS 2010), pp. 103–112. ACM, New York (2010)
13. Ryu, H.-S., Yoon, Y.-J., Lim, M.-E., Park, C.-Y., Park, S.-J., Choi, S.-M.: Picture navigation using an ambient display and implicit interactions. In: Proceedings of the 19th Australasian Conference on Computer-Human Interaction: Entertaining User Interfaces (OZCHI 2007), pp. 223–226. ACM, New York (2007)