Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 34 (2014) 625 – 632

Procedia Computer Science

International Symposium on Emerging Inter-networks, Communication and Mobility (EICM)

# A Cloud-based Interaction Management System Architecture for Mobile Devices

Edward R. Sykes*

*Sheridan Institute of Technology and Advanced Learning, 1430 Trafalgar Road, Oakville, Ontario, L6H 2L1, Canada*

**Abstract**

The number of interruptions people experience on a daily basis has grown considerably over the last decade and this growth has not shown any signs of subsiding. In fact, with the exponential growth of mobile computing, interruptions are permeating the user experience. Systems must be developed to manage interruptions by reasoning about ideal timings of interactions and determining appropriate notification formats.

In this work, an architecture for a cloud-based interruption management system for mobile device users is presented. The system draws from rich contextual information from the mobile device (i.e., user, task and environment dimensions) and real-time observations of the user's activities and then reasons about ideal times to interact with the user. The reasoning component (interruption algorithm) is situated in the cloud and implemented using a novel machine learning technique (an Adaptive Neuro Fuzzy Inference System). This research addresses the complex problem of determining the precise time to interact with a mobile device user and in so doing aims to reduce the negative aspects of interruptions. This paper also presents a new interruption taxonomy built on an existing framework, and a report on the current prototype developed.



*Keywords:* Human Computer Interaction; mobile devices; user modeling; intelligent agents; machine learning.

* Corresponding author. Tel.:+1-905-845-9430 Ext 2490; fax: +1-905-815-4035.
*E-mail address:* ed.sykes@sheridancollege.ca


doi:10.1016/j.procs.2014.07.086

## 1. Introduction

The unprecedented growth in the mobile computing field comes with a price. "Mobile phones have become the most hated device that people cannot live without. For its primary usage as a communication device, it has surpassed any other medium. But it comes with a high price—*interruption—anywhere anytime*." pg 2060, [1]. The problem of mobile device interruptions has gained the attention of a variety of experts in psychology, transportation, health, and pervasive and mobile computing [1]. In 2011, it was discovered that undesirable interruptions account for 28% of the knowledge worker's day, which translates to $28 billion in unproductive hours in companies in the United States [2]. The financial implications to this reality is a loss of $700 billion each year based on an average labour rate of $25 per hour for knowledge workers [3]. To date there have been no signs of this abating [3].

Unfortunately, determining when to interact with a mobile device user at appropriate times is a non-trivial problem. From an algorithmic perspective, it is difficult to determine the precise time to interrupt a user. Thus, systems must be developed to manage interruptions in terms of reasoning about ideal timings of interruptions and determining appropriate presentation formats for the interruption message. Furthermore, in the context of mobile devices, there have been only a few interruption systems developed and, in these solutions, have several limitations. For example, one drawback in one system is that it resides on the mobile device itself and therefore has some restrictions: a) battery life of the device, b) cpu computational limitations, and c) user modelling limitations [1].

A cloud-based interaction management system for mobile devices aims to address these and other issues, reduce untimely interruptions, and improve the user experience for mobile device users. This work may also shed light in other areas such as, office software suites, scheduling systems, and mixed-initiative interaction systems because, from a broader perspective, the problem of deciding when to interact with a user is still largely unsolved. In this work, a cloud-based interaction management system for mobile device users is presented including a brief description of the machine learning algorithm, which is the core of the cloud-based service. The cloud-based interaction management system is implemented using a client-server model architecture wherein the mobile device-based agent (client) extracts real-time user, task, and environment contextual information and communicates with a cloud-based service that houses the interaction management system (server). This system was designed with the following design characteristics:

1. make accurate decisions when uncertainty is present;
2. be computationally efficient (i.e., the cloud-based service must be able to make interruption decisions in real time, however the machine learning process may take longer than real-time response);
3. employ a user model (e.g., preferences, frequently used mobile apps and tasks, etc.) that may be used in the interruption decision making process;
4. provide a reasoning process that is easily interpretable by a human (i.e., the algorithm's decision making process to interrupt or defer an interruption must be easy to be examined and understood by a human). This characteristic provides the opportunity for deeper reasoning into why an interruption occurred as well as insight into the form and content of an appropriate interruption message;
5. support supervised learning (capable of accepting input-output patterns and learning these associations);
6. learn quickly and efficiently (limited user-interruption session data may be available);
7. consider the limitations of the mobile device (i.e., network bandwidth and availability, battery consumption characteristics, memory and cpu limitations);
8. draw on knowledge representation gained from a collective user model representative of similar users – the cloud-based service may use this collective user model during the reasoning process to determine appropriate times to interrupt the user.

## 2. Literature Review

Interruptions happen for a multitude of reasons. The purpose of an interruption could be for many reasons, such as answering a call, responding to an SMS text message, receiving a notice of an event in the calendar among other situations. There are four known strategies for managing interruption: (a) immediate, (b) scheduled, (c) negotiated, and (d) mediated [4, 5].

The *immediate* interruption strategy involves interrupting the person immediately regardless of what they are doing in a way that insists that the user stop what they are currently working on and respond to the interruption. The *scheduled* strategy involves restricting the agents' interruptions to a prearranged schedule, such as, every 10 minutes. The *negotiated* interruption strategy would have the agent announce their need to interrupt and then support a negotiation with the person. This approach gives the user full control over how to deal with the interruption—when or even at all. The fourth strategy, called *mediated*, involves agents indirectly interrupting and requesting interaction through a broker. The broker would then determine when and how the agents would be allowed to interrupt the user. In this research the cloud-based interaction management system provides the mobile-based agent the intelligence in reasoning about when to interrupt the user.

Most of the current research is focused on mediated and negotiated strategies with research in the mediated strategy area growing considerably over the last few years. This trend is attributed to the fact that the immediate and scheduled interruption strategies are well-defined, have undergone considerable research, and are significantly easier to design and implemented. Associated with mediated strategies are intelligent systems that observe the user as s/he is performing tasks to decide when to interrupt the user and how best to present the pertinent information. Despite the progress that has been made in systems supporting mediated strategies, negotiated strategies (user determined) are still the best overall solution when considering factors such as cost of interruption, resumption lag, and overall performance in performing tasks on mobile devices [6-8].

Much of the research to date has been centered on using attributes from the task domain. However, there are other aspects surrounding the problem of determining when to interrupt a user. These other aspects involve the user context and environment context [9]. Models have been proposed to provide a more encompassing perspective of interruption; however, at this time, there is no standard or unified model that has been accepted in the research community. At this time, there is only one interruption taxonomy and it was proposed by Gievska & Sibert [10]. In their model three dimensions were used to situate the context of a potential interruption: user, task, and environment. In this work, we extend and update their model to include another dimension, *temporal* that is intended to work in conjunction with the other 3 dimensions when reasoning about ideal interruptible moments.

### *2.1. Interruption and Mobile Computing*

Mobile Computing is pervasive in our society; every aspect of our daily lives has been touched by the ubiquitous nature of mobile devices. We have experienced an unprecedented growth of Mobile Computing—a trend that seems to have no limit. The industry recently reached a tipping point, where smartphone and tablet shipments exceeded desktop and laptop shipments combined [11].

Most mobile devices do not employ an interruption management system. However, research and development in this area is gaining momentum. Recent versions of Apple's iOS platform provide a built-in *notification center*. This center centralizes all notifications pertaining to emails, texts, calendar events, friend requests and many other informational messages. This center queues these pending messages and places them in an unobtrusive yet easily accessible location for the mobile user to retrieve with a simple downward gesture from the top of the screen [12]. In previous versions of the iOS, notifications were simply presented to the user when they arrived. This resulted in many unwanted interruptions for the mobile user.

Mobile device users experience a loss of productivity due to interruptions [1]. Furthermore, these interruptions are not pertinent nor useful to the current task at hand [1]. With the growth of mobile computing, researchers are exploring new ways to manage smartphone disruptions using advanced features such as ubiquitous computing and context aware systems [3, 12-14]. These issues combined with the unprecedented growth in mobile computing provide further motivation for this research.

### *2.2. Interruption Management Systems for Mobile Devices*

Some interruption management systems have been proposed and developed. For example, Khalil [15] used calendar information of the phone to minimize disruptions. Marti [16] created an application that decided whether or not to let a phone ring based on all of the members' votes for group meetings. Their application took into account the time of day and the user's contact database. Dekel [17] built an application that minimizes mobile phone interruptions by intelligently changing profile settings. Zulkernain [3] created a comprehensive solution however, their interruption management system runs on the device and therefore does not address the device's limitations nor leverage the potential benefits of cloud computing.

## 3. Architectural Model

This research is based on related work, namely [18] for the mobile-based agent data collection and cloud-based services. This section presents (a) a new interruption taxonomy (extended from [10]) that provides the foundation for this model, (b) the high-level architectural model of our Interaction Management System, (c) the mobile device-based agent and (d) an overview of the Interaction Algorithm.

### *3.1. Interruption Taxonomy*

The foundation of the architectural model is a new interruption taxonomy based on [10]. Fig. 1. shows this new interruption taxonomy. In our extended taxonomy, the three dimensions from the original are still present (User, Task, and Environment), however, an additional one is now represented (Temporal). To effectively reason about appropriate times to interrupt a mobile based user requires information from the temporal context (e.g., time of day [typical working hours, evening, etc.], type of day [weekend, weekday, etc.]). This aiding context becomes essential as contextual information from the other dimensions are brought into the decision making process (e.g., Personal Relations [parameter in the User Context] [Contacts like family, friends, acquaintances, etc.], Situational factors [Environment Context] [calendar schedule, etc.]). Thus, this new dimension is represented in the new taxonomy.

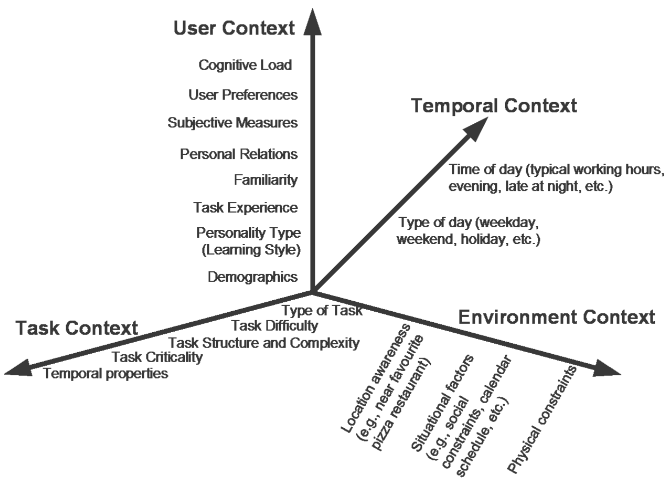



Fig. 1. Extended Interruption Taxonomy for Mobile Computing Interaction Modeling (extended from: [10]).

### *3.2. Interaction Management System Architectural Model*

This section presents the architectural model for the interaction management system. Fig. 2. shows the model depicting the relationship between the mobile device-based agent and the cloud-based agent (interaction management system) and other significant components of the model.

The architectural model has three distinct parts: the mobile device with which the user (*U*) interacts; the Data Management Module (centre of the interaction management system); and the Interaction Algorithm (located on the right). The Data Management Module is made up of two main units: (a) the Temporal, Task and Environment, and (b) the User Model.

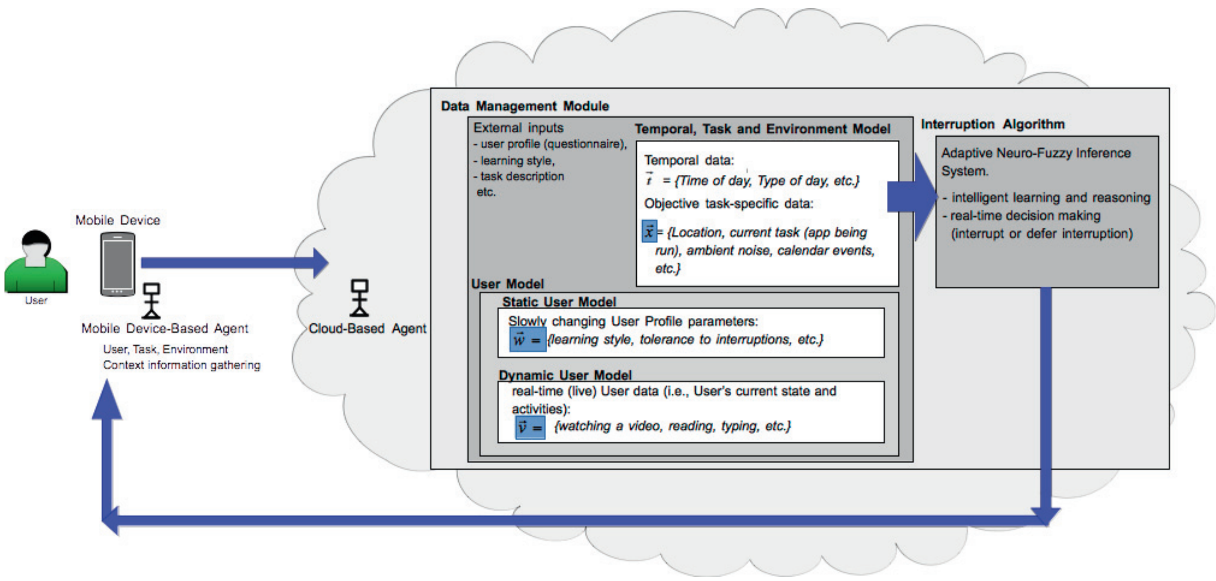



Fig. 2. Architectural model for the cloud-based Interaction Management System.

The Temporal, Task, and Environment model represents specific details regarding the task (and all of its constituent subtask components), temporal elements, and situational and contextual environment details. The User Model represents static and slowly changing data (static user model) and real-time data (dynamic user model). All of the data in the Data Management Module is contained in the vectors $\vec{t}, \vec{v}, \vec{w}$, and $\vec{x}$. The Temporal vector, $\vec{t}$ represents time and date characteristics (e.g., time, type of day, etc.). The Dynamic User Model contains vector $\vec{v}$ represents live (real-time) data consisting of *U*'s current state and activities (e.g., determining if *U* is browsing, texting or talking, etc.). The Static User Model holds vector $\vec{w}$, which represents static user data (e.g., user modeling information such as tolerance to interruptions, frustration level, distractibility level, etc.). The Temporal, Task and Environment Model also holds vector $\vec{x}$, which represents objective task-specific and environment-specific information (e.g., Task complexity, current GPS coordinates, situational and contextual information). Parts of $\vec{x}$ may be set before *U* (mobile-based agent) commences interaction with the system. The arrow on the bottom represents a feedback-cycle from the Interaction Algorithm back to *U* indicating that an interruption should be allowed or to defer the interruption until a more suitable time.

### *3.3. Mobile device-based agent*

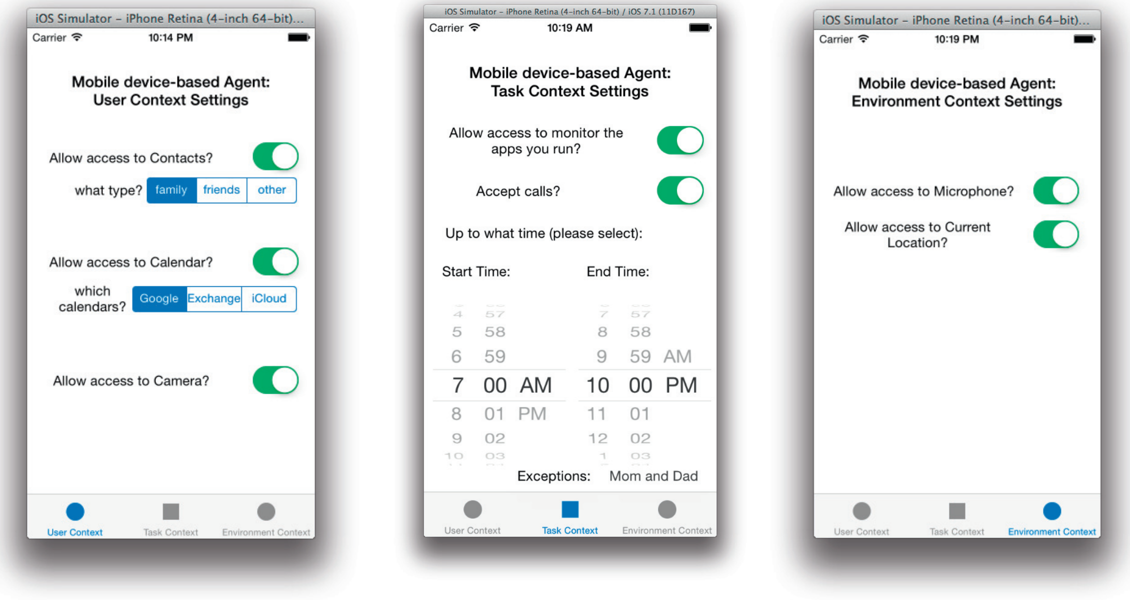
The mobile device-based agent is currently implemented as an iPhone app and a variety of different information from calendar, microphone, location, etc. depending on the user's selections. Based on the user's choice, the corresponding information is collected and transmitted to the cloud-based service for processing. Fig. 3. shows the app with various options for the user to control how the agent extracts information from the three contextual dimensions.

### *3.4. Overview of the Interruption Algorithm*

This section presents an overview of the Interaction Algorithm. The algorithm needs the following information to determine appropriate times to interrupt the user:

$\vec{t}$: vector representing temporal contextual information (e.g., time of day, type of day, etc.);

$\vec{v}$: vector representing *U*'s real-time activities (e.g., texting, reading, talking, etc.) (This data is extracted real-time from the mobile device-based agent);




(a) (b) (c)

Fig. 3. Mobile device-based agent implemented as an iPhone app for various settings: (a) User Context, (b) Task Context, and (c) Environment Context.

$\vec{w}$: vector representing the static data for the user (e.g., user modeling information such as learning style, personality traits, frustration level, distractibility level, etc.);
$\vec{x}$: vector representing task and environment specific information (e.g., task complexity, current GPS coordinates, situational and contextual information, etc.);

The initial design of the algorithm is based on a Fuzzy Inference System where linguistic variables and membership functions were identified and rules were identified that use information from the four contextual dimensions. Table 1 shows the high-level description of the algorithm. The algorithm takes the input from $\vec{t}$, $\vec{v}$, $\vec{w}$ and $\vec{x}$ as previously discussed, and, in addition, the algorithm takes two other arguments: a trained ANFIS network and a threshold limit value. The threshold limit value represents a means to control the degree of sensitivity in the operation of the network. This value is [0,1]. Suppose the threshold value is set at 0.75. If the output of the network is above this threshold, then the algorithm will report a 1 (i.e., interrupt now) decision. Any computed output less than the threshold will result in 0 (i.e., defer the interruption).

## 4. Methodology

The core of the interaction management system is the interruption algorithm. The interruption algorithm has been implemented using Matlab's ANFIS tool in the Fuzzy Logic Toolbox. During the initial development phase, fuzzy rules were constructed using knowledge engineering practices by observing 3 participants and their use of iPhones in different contexts over a period of 3 weeks. A subset of the rule base is presented in Table 2. Collectively, these rules comprise the initial model before ANFIS learning. The ANFIS model was then presented training data composed of interaction sessions using a supervised vector resulting in *interrupt* or *defer interruption*. The algorithm was then tested using representative yet unique data and evaluated using confusion matrix analysis.

Table 1. High-level description of the Interaction Algorithm.

```
Algorithm Interruption_Algorithm (t⃗, v⃗, w⃗, x⃗, trained ANFIS, interruption_threshold_limit)
Inputs : information from cloud-based agent and the user's mobile device (via mobile device-based agent):
        t⃗:  temporal contextual information
        v⃗: real-time user activity data         //  Input extracted externally from the software U is using:
        w⃗:  static user data  // Input  extracted externally from the software U is using:
        x⃗:  task and environment specific information
        trained_ANFIS
        interruption_threshold_limit  ← range: [0,1]  | 0 = do not interrupt, 1= interrupt
Output : interruptDecision:  ← range: [0,1]  | 0 = do not interrupt, 1= interrupt
Begin
    interrupt  ← 0     // default is defer an interruption
    while ( mobile device is operational )
        // run ANFIS with new inputs
        ANFIS ←  t⃗
        ANFIS ←  v⃗       // real-time data
        ANFIS ←  w⃗       // static user data
        ANFIS ←  x⃗       // task, environment, and temporal specific data
        If ( interruptDecision > interruption_threshold_limit  )  then
            interrupt User
        end if
    end while
End
```

Table 2. Sample subsets of fuzzy rules.

| Context | Fuzzy rule |
|---|---|
| *Temporal* | If (time_of_day is work_hours) and (not in a meeting) then (interruptDecision is yes) |
| | If (time_of_day is no_calls_permitted) then (interruptDecision is no) |
| *Environment* | If (type_of_day is weekend) and (time_of_day is calls_permitted) then (interruptDecision is yes) |
| | If (type_of_day is weekend) and (time_of_day is calendar_birthday_party) then (interrupDecision is no) |
| *User* | If (task_complexity is high) and (cognitive_load is high) then (interruptDecision is no) |
| | If (tolerance_to_interruptions is low) and (distractibility is high) then (interruptDecision is no) |
| *Task* | If (task_criticality is high) and (time_of_day is calls_permitted) then (interruptDecision is no) |
| | If (current_task is texting) then (interruptDecision is no) |

## 5. Discussion

The number of interruptions people experience on a daily basis has grown considerably over the last decade and this growth has not shown any signs of subsiding. In the field of mobile computing, this growth is growing exceedingly quickly. In this work an architecture for a cloud-based interruption management system for mobile device users was presented. The system draws upon a collection of contextual information from the mobile device (i.e., user, task, environment), real-time observations of the user's activities coupled with temporal contextual information to reason about ideal times to interact with the user. The reasoning component (interruption algorithm) is situated in the cloud and implemented using a novel machine learning technique (an Adaptive Neuro Fuzzy Inference System). This research shed light on the following:

1. The algorithm developed uses a user model in its' reasoning computations. Most of the research in this area has focused on task-based contextual information when designing systems that reason about interruptions. Researchers support additional work should be done in this area by including subjective preferences.
2. The algorithm's performance is quite promising at 96% accuracy in several models created.
3. The algorithm was implemented using an advanced machine learning technology—an Adaptive Neuro-Fuzzy Inference System—which is a novel contribution.
4. The algorithm developed does not rely on any user involvement. In other systems, users laboriously review video sessions after working with the system and record interruption annotations so that the system can learn.
5. This research shed light on reasoning about ideal interruption points for free-form tasks. Currently, this is an unsolved problem.

This research addresses the complex problem of determining appropriate times to interact with a mobile device user and in so doing aims to reduce the negative aspects of interruptions. This paper also presented a new interruption taxonomy built on an existing framework and reported on the current state of the prototype developed.

## References


1. Zulkernain, S., et al., *A Mobile Intelligent Interruption Management System.* Journal of Universal Computer Science, 2010. **16**(15): p. 2060-2080.
2. Spira, J.B. and J.B. Feintuch *The Cost of Not Paying Attention: How Interruptions Impact Knowledge Worker Productivity*. Basex, 2005.
3. Zulkernain, S., P. Madiraju, and S.I. Ahamed. *A context-aware cost of interruption model for mobile devices*. in *IEEE International Conference on Pervasive Computing and Communications*. 2011. Seattle, WA.
4. Guinn, C.I., *Evaluating Mixed-Initiative Dialog.* IEEE Intelligent Systems, 1999. **14**(5): p. 21-23.
5. McFarlane, D.C. and K.A. Latorella, *The scope and importance of human interruption in HCI design.* Human-Computer Interaction, 2002. **17**: p. 1–61.
6. McFarlane, D.C., *Comparison of Four Primary Methods for Coordinating the Interruption of People.* Human–Computer Interaction, 2002. **17**: p. 63-139.
7. Lottridge, D., *Individual Differences and Their Impact on Responses to Immediate Versus Negotiated Notification in a Simulated Driving Task*. 2006, University of Toronto: Toronto.
8. Ramchurn, S.D., et al. *Minimising intrusiveness in pervasive computing environments using multi-agent negotiation*. in *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*. 2004. Los Alamitos: IEEE Computer Society.
9. Iqbal, S. and B. Bailey, *Leveraging Characteristics of Task Structure to Predict the Cost of Interruption*, in *CHI 2006*. 2006, ACM: Montreal, Quebec, Canada. p. 741-750.
10. Gievska, S., R. Lindeman, and J. Sibert, *Examining the Qualitative Gains of Mediating Human Interruptions during HCI*, in *HCII*. 2005: Las Vegas, Nevada. p. 22-27.
11. Cisco. *Cisco Visual Networking Index Global Mobile Data Traffic Growth (2010-2015)*. 2010 December 19, 2010]; Available from: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html.
12. Zatz, D. *Redesigning Apple's iOS 5 Notification Center*. 2011 June 24, 2011]; Available from: http://www.zatznotfunny.com/2011-06/redesigning-apples-ios-5-notification-center/.
13. Marshall, N. *iOS 5 third party notication center widgets headed to an iPhone near you.* 2011 June 24, 2011]; Available from: http://www.electricpig.co.uk/2011/06/24/ios-5-third-party-notification-center-widgets-headed-to-an-iphone-near-you/.
14. Rosenthal, S., A.K. Dey, and M. Veloso. *Using Decision-Theoretic Experience Sampling to Build Personalized Mobile Phone Interruption Models.* in *International Conference on Pervasive Computing* 2011.
15. Khalil, A., Connelly, K. *Improving cell phone awareness by using calendar information*. in *INTERACT Conference*. 2005. Rome, Italy.
16. Marti, S., Schmandt, C. *Giving the caller the finger: collaborative responsibility for cellphone interruptions.* in *CHI '05 Extended Abstracts on Human Factors in Computing Systems*. 2005. Portland, OR, USA.
17. Dekel, A., Nacht, D., Kirkpatrick, S. *Dekel, A., Nacht, D., Kirkpatrick, S.: Minimizing mobile phone disruption via smart profile management*. in *11th international Conference on Human-Computer interaction with Mobile Devices and Services, Bonn, Germany*. 2009.
18. Sykes, E.R., et al., *A Privacy-Enabled Mobile Computing Model Using Intelligent Cloud-Based Services*, in *SmartData*. 2013, Springer Science+Business Media: New York. p. 107-115.