A Context Aware Interruption Management System for Mobile Devices

Sina Zulkernain, Praveen Madiraju, and Sheikh Iqbal Ahamed

Dept. of Mathematics, Statistics & Computer Science
Marquette University, Milwaukee, WI – 53233, USA
{sina.zulkernain,praveen.madiraju,sheikh.ahamed}@marquette.edu

Abstract. To prevent unwanted interruptions from cell phones, this paper proposes a system solution considering user's unavailability. We first look at desirable characteristics of the system, then design a system architecture which takes as input user preferences, relevant context information and then produces as output if an incoming call should be allowed to ring. We also present a case study application that benefit by using the interruption management system. Finally, we discuss evaluations of the system by (i) evaluating the prototype and (ii) undertaking cognitive walkthroughs of the application.

Keywords: Context Aware System, Interruption, Ubiquitous Computing, Unavailability.

1 Introduction

Cellular phones have only been in use for mass communication during the last decade or so. However, the International Telecommunication Union estimates that cellular subscriptions worldwide have reached approximately 4.6 billion by the end of 2009 [39]. Along with its major role as a phone, cell phones have features like text messaging, voice messaging, data transferring and even the Internet. So it is not too astonishing to realize the rapid growth in mobile phone production. In a very short time period phones have come a long way, and the newer versions are smartphones with a number of applications built in like camera, games, GPS, calendar, alarm clock, notes, speech recognizer, touchpad etc. There has been a tremendous growth in smartphone applications too. Recent statistics indicate that there are over 100,000 active iPhone applications [42]. The massive number of users and enormous number of applications make cell phone a device integrated to our daily life. A University of Michigan study [40] shows that 83% people think cell phones make life easier and they choose it over the Internet.

Definitely with mobile phones, there is the obvious benefit of all the moment communication, but irrespective of time and place, we do expect a phone to ring. A ringing phone interrupting at an inopportune moment can be very disruptive to the current task or social situation [17]. In a survey of 1000 senior executives, it was reported that undesirable interruptions constitute 28 percent of the knowledge worker's day, which translates to 28 billion wasted hours to companies in the United

States alone [34]. It results in a loss of 700 billion dollars per year, considering an average labour rate of $25 per hour for information workers [43]. A University of Oxford experiment suggests that in cognitively demanding situations, the advantage that 18-21 year olds enjoy over 35-39 year olds is reduced by an interruption caused by electronic communication technology [41]. Interruptions are mostly not beneficial to the immediate task and moving them a few minutes into the future could greatly benefit many users [2]. Undesired disruption causes interrupted users to take up to 30% longer to complete and commit up to twice the number of errors [4]. In order to mitigate the aforementioned problems, we propose a mobile interruption management system that will decide in real-time whether the user should be interrupted or not.

1.1 Necessary Characteristics of the System

Analyzing different scenarios, we identified the desired characteristics of an interruption management system in our earlier work [35]. In short the required characteristics are:

Mobility (C1). The system must be installable on a small handheld device being mindful of data transference costs, memory and CPU limitations.
Customizable (C2). Rules and outcomes must be customizable by and for each user.
Adaptable (C3). The system must be able to change itself to different environments from CPU power, screen size to input methods.
Context Aware (C4). The system has to be aware of its contexts i.e. take inputs from its surroundings.
Automated (C5). The system must make decisions all by itself without user interaction.
Unavailability Aware (C6). The system should take into account different modes of unavailability like audio, visual or touch by changing the interruption method to ring, vibrate or go silent.

1.2 Similar Researches

Several research studies have investigated the issue of interruption management in general [8, 15, 16, 20] and also specific to mobile devices [11, 17]. Dekel et al. [11] built an application that minimizes mobile phone interruptions by changing profile settings intelligently. Savioja [32] addresses different kinds of alarms for different types of interruptions in control room environments. Khalil & Connelli [25] use calendar information of the phone to minimize disruptions. Marti & Schmandt [28] devised an application for a group setting where a phone had to get all of the members' votes before ring. Also a methodology and design process for building interruption aware system is proposed in [15]. However, the distinguishing aspect of our work in comparison to the aforementioned ones is we have identified desirable characteristics of the system and show that our solution satisfies all of them.

The system we propose uses the capabilities from ubiquitous computing and context aware systems to programmatically learn about the environment and achieve our goals. Modelling context information and software engineering framework for context aware pervasive computing are already built in [18, 19]. We also have location and environment aware handheld systems [24] and frameworks in development for generalizing the sensor interfaces [14]. We have distributed resource discovery [33] and trust models for anonymous sensors [3]. Altogether, the avenue is clear for revolutionary system development awaiting only the sensor deployments.

1.3 Contribution of This Paper

The contribution of this paper is an intelligent interruption management system for mobile phones. The system is intelligent because of its adaptability, awareness for both context and unavailability, and also automatic decision making capability.

In our earlier poster paper [35], we proposed preliminary system architecture for an interruption management system with initial results. In this paper, we are extending on our previous work and furthermore provide the following contributions:

- Designed and developed system architecture for the system.
- Carefully surveyed the state of the art.
- Implemented a case study application using the developed system architecture.
- Evaluated the prototype application.
- Gathered users' feedback regarding the usability of the case study application.

The rest of the paper is organized as follows. Section 2 provides the system architecture and Section 3 focuses on the case study implementation. Evaluation of the system is done in Section 4 followed by related works in Section 5. Finally, Section 6 concludes our findings and paves the way for future works.

2 General System Architecture

In this section, we present our proposed general system architecture for mobile interruption management system. The general components of the system are shown below in Figure 1.

The large unit on the right, labeled Unavailability System is the system installed on the handheld. The system is divided into three tiers. The first tier includes the Context Service Interface in the upper left and Context Data Store in the upper right of the main unit. Upon reception of data, this tier collects and stores them in a continual process. In the middle tier, we have the User Interface Component that saves user preferred configurations in the Training Rules Data Store. The main component of the third tier is the Tree Generator. The tree generator collects the available context information from the first tier and user preferences from the second tier. This is the processing stage whereupon the decision structures are made. Tree Generator provides its decisions to the Content Provider Interface. The Content Provider Interface then instructs the Ringer Application on the phone to either ring or not ring. Finally, Ringer Application on the left is external to our system, but internal to the handheld's operating system.

Tier 1 (Context Information). Context Service Interface in Figure 1 is a persistent service on the mobile device that actively searches for context information from publicly available sensors. It aggregates information from the internal sources and generates a Context object to be stored in the Context Data Store. This object is passed to the Tree Generator (Decision Tree tier) for processing.

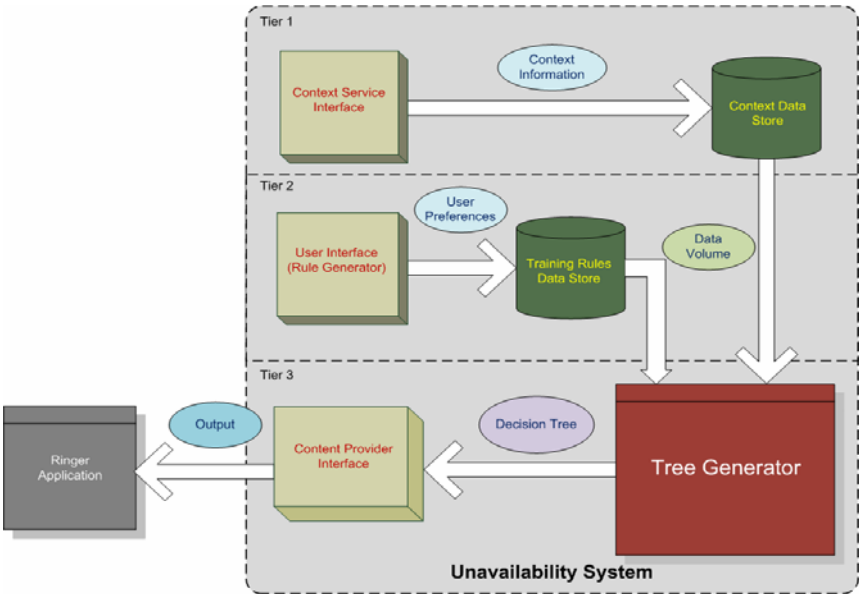


Fig. 1. General System Architecture

Tier 2 (User Preferences). Each user has a set of preferences about when to let calls through or when to deny them in any circumstances. These rules appear contradictory between users. So we need some sort of training data that needs to be collected either through a survey or another form of user inputs e.g. from the user interface.
Tier 3 (Decision Tree). The Tree Generator receives inputs of Context Data Point from Tier 1 and user preferences from Tier 2 and then generates a decision tree structure. A decision tree is a structure of conditional code that classifies a data set into one or more categories. A decision making piece of code is one that takes in context data such as sensory data and then activates the correspondingly correct action.

Due to space constraints, we are omitting a detailed description of the system architecture; however interested readers may refer to our earlier poster paper in [35] for a thorough discussion of the system architecture.

3 Case Study Implementation

Here we present a case study implementation where a private party (anonymous for privacy reasons) wants to send all its sales employees performance metrics each hour. These metrics represent their production, sales and the employee's current rank compared to others. They use it to make their sales force competitive within the company. Each sales person has a specified work area and the company wants to send the metrics to him/her whenever s/he is in the zone. Some sales people work indoors and they are mostly in the managerial positions. Now even though a sales person may be in his/her work area, s/he might be busy in a meeting with superiors. Whenever the sales person is in some scheduled event, the company does not want to send the metrics. Based on these requirements, we developed our prototype application. Whenever the sales person is in their designated area, not busy in a scheduled event and also when the time is between his/her work hours, we show the metrics on the device. But when any of these cases fail, we do not provide the metrics.

The prototype is developed on the Android, the operating system of a new class of smartphones which was designed primarily at Google in participation with the Open Handset Alliance. The reason for choosing Android is that it is Linux to the core and entirely open sourced. Most importantly when there is no call (in this case when no data is sent from the server); our application can run as a background process using minimal CPU and battery resources. The application needs to be installed in the receiver phone and Android is the only platform that allows full control of the ringer actions i.e. the interrupter. In the future, we also plan to implement it on other platforms as well. For the prototype, we used three contexts: location, schedule, and day of week along with time of day. We used Google Calendar as our scheduler. So our assumption is whoever uses our system will have some sort of scheduler where the application can query into. We used GPS service provided by Google to identify location and system clock service to find day and time of the week. Now we show step by step screen shots (see Table 1 – Figure T1-T8) to explain how our application is working.

Figure T1. The application first looks for its office location. It knows from its data storage what user's office address is and pinpoints that location using Reverse GPS service. Figure T1 in Table 1 shows user's office address.
Figure T2. The application then looks for user's current location. It uses GPS service and shows a region of radius 1000 meters where user could be. Figure T2 in the table shows that region by a black circle. Measuring the distance from the centre of this circle to user's office location, the application decides that user is at office.
Figure T3. Now the application uses the system clock to get today's day of week and current time. From its data storage the application determines that it is user's working hour.
Figure T4. It shows user's scheduler, in this case the Google Calendar. The application now queries the calendar to get user's current schedule.
Figure T5. The application sees that user has no event specified at the current time. So it decides to show the salesperson metrics sent from the office.
Figure T6. This figure shows the things the application considered before showing the metrics. The first line shows user's current location in latitude and longitude and user's current address in next. Then it shows the distance between user's current location and his/her office location. Next line shows current day and time. The application then shows user's status. The last line in the figure shows the metrics.
Figure T7. Now we make a change in the calendar and put an event there. Now the user is supposed to be busy. So the application sees that user is busy and now takes a decision not to show the metrics to the user.
Figure T8. This figure shows in the last line the event the user is currently attending. Also it shows the time left for this event to finish.

Table 1. Screenshots of the application

Figure T1	**Figure T2**

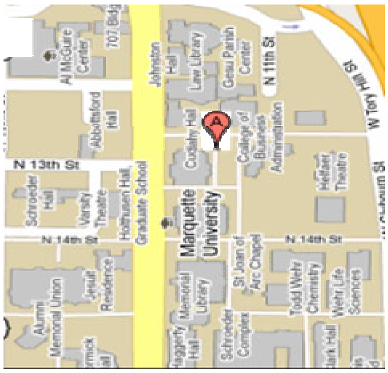
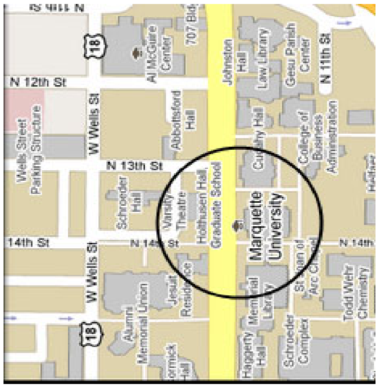

	

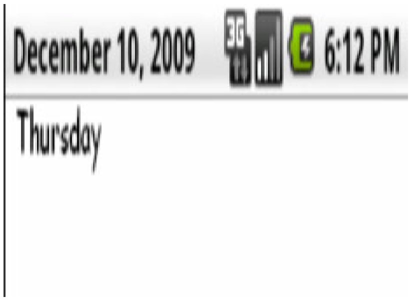
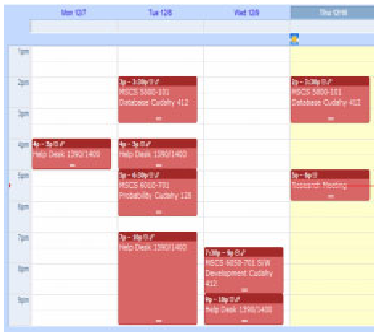
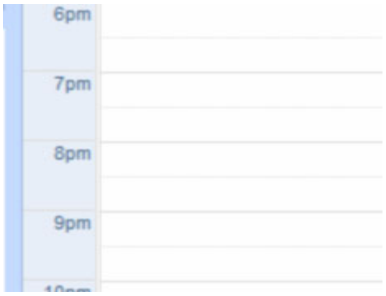
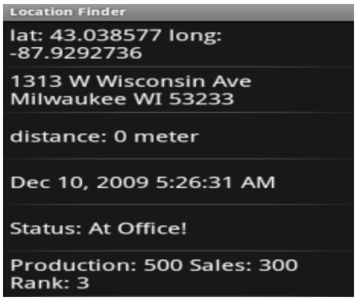
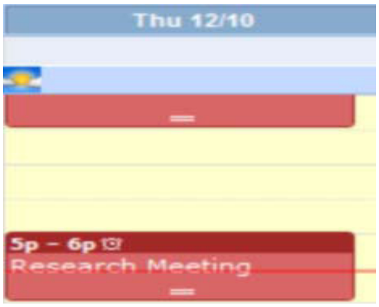
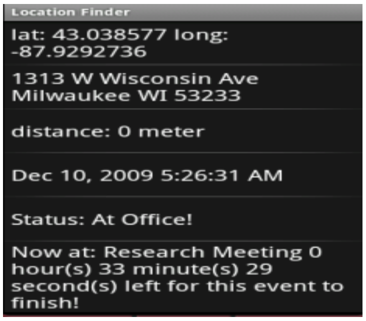
Figure T3	**Figure T4**
 December 10, 2009 6:12 PM Thursday	 Mon 12/7 Tue 12/8 Wed 12/9 Thu 12/10 2p – 3:30p MSCS 5800-101 Database Cudahy 412 4p – 5p Help Desk 1390/1400 5p – 6:30p MSCS 6000-701 Probability Cudahy 128 5p – 6p Research Meeting 7p – 10p 7:30p – 9p MSCS 6050-701 S/W Development Cudahy 412 9p – 10p
Figure T5	**Figure T6**
	 Location Finder lat: 43.038577 long: -87.9292736 1313 W Wisconsin Ave Milwaukee WI 53233 distance: 0 meter Dec 10, 2009 5:26:31 AM Status: At Office! Production: 500 Sales: 300 Rank: 3

Table 1. (*Continued*)

Figure T7	**Figure T8**
	

lat: 43.038577 long: -87.9292736

1313 W Wisconsin Ave Milwaukee WI 53233

distance: 0 meter

Dec 10, 2009 5:26:31 AM

Status: At Office!

Now at: Research Meeting 0 hour(s) 33 minute(s) 29 second(s) left for this event to finish!

4 Evaluation

We have already done a simulation of our system for scalability in our earlier work [35]. Here we provide a detailed evaluation the system by evaluating the prototype and cognitive walkthroughs of the application.

4.1 Prototype Evaluation

In our system, data volume received is calculated below 50 kilobytes at any given environmental change. We have over 64MB to use and the memory usage requires a fraction of that. With respect to data transference, this is a perfect conformity to characteristic C1 (mobility). The system is installed on a cell phone with dimensions only 117.7 mm × 55.7 mm ×17.1 mm and in respect to size and portability, it conforms to C1 (mobility). To acquire information, our system uses sensors built into the device to gather data as to the local context. So it is C4 (context aware). Also, the unavailability system accepts varying sensor sets in differing situations and environments. Our prototype platform has built in GPS for instance, and this is one of the only data sources that can be assumed to exist for all deployments and use cases. Thus the system is C3 (adaptable). Decision tree traversal is a linear process. So the CPU power usage is very low along with the battery concerns. Our system keeps a ready to use decision tree so that when a call comes in, it can immediately make a decision and prevent the interruption without interference by the user. This prevents interruption and satisfies C5 (automated). With each decision tree having a different structure due to the per user customization, we have easily satisfied C2 (customizable). The user interface component is a constant time computation; again less CPU and memory usage. This system also utilizes the varying modes of unavailability; vision, hearing and touch. These levels of permissions for incoming communication attempts make the system smarter and more user friendly, providing appropriate attention to the different sorts of unavailability modes and thus satisfies

C6 (unavailability aware). Thus, all the characteristics outlined in Section 1 have been realized in the solution we have proposed.

4.2 Cognitive Walkthrough

To get the proper assessment of our application, we used the cognitive walkthrough strategy. We did a survey on a group of 30 people on the usability and usefulness of our application. First we explained the problem, briefly went over some of the issues we addressed and then showed the prototype application demo. The distribution of the participants is as follows: 17 undergraduate students, 8 graduate students, 2 faculty members, 2 entrepreneurs, and 1 other.

We handed 5 questions about the application over to each participant and requested them to answer them on a scale of 1 to 5. The questionnaire for the survey is given below:

Question 1. Overall, how would you rate the services? (*1 = Very Poor, 5 = Excellent*)

Question 2. What is the effectiveness of this application? (*1 = Not Effective at all, 5 = Very Useful*)

Question 3. How easy is it to give the input? (*1 = Very Hard, 5 = Very Easy*)

Question 4. Will you pay to use this application? (*1 = Definitely Not, 5 = Definitely Yes*)

Question 5. Would you recommend this application to a friend? (*1 = Surely Not, 5 = Surely Yes*)

The survey results are shown in Figure 2:

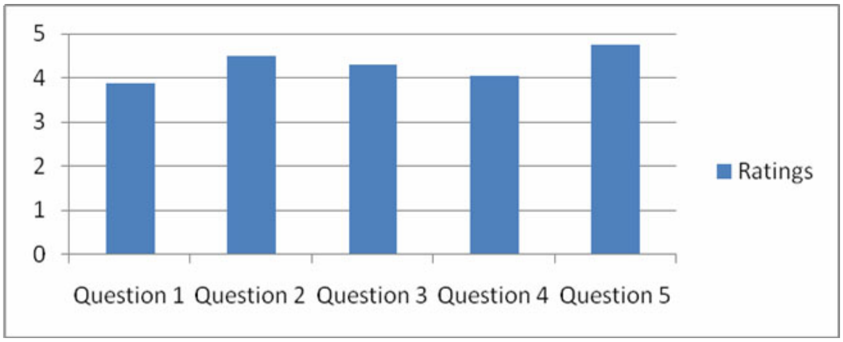


Fig. 2. Survey Results

The results from Figure 2 indicate that participants were enthusiastic about the application and its usability.

5 Related Works

When a call is made to a phone, the system decides beforehand not to the let the call go through if it is a costly interruption, The cost of interruption (COI) is a function of immediate task and the user's state of mind, which can also be seen as a function of the task at hand. A proper ubiquitous computing system can theoretically understand the task at hand and infer the user's state of mind and therein get a measure of COI. Hence the survey of literature spans into areas related to COI, interruption management and context aware systems.

5.1 Cost of Interruption

Adamczyk [1] measures the effects of interruption in terms of task performance, emotional state and social attribution. The study also aims to find the most suitable time to interrupt the user. Several researchers have addressed the issue of cost of interruption [5, 23, 16]. Mark [27] measures the COI based on additional time required to reorient back to the primary task and mental stress brought upon the interruptee. A user's pupil size increases due to the mental processing efforts and there is an upper bound on how much it can grow. Bailey [5] shows this could be a possible way to measure a user's mental stress and hence decide whether interruption could be detrimental or a bit refreshing anyway. The bottom line is to defer interruption when COI is high. This has been shown to not only increase worker efficiency, but also benefit morale [2].

5.2 Interruption Management

To manage interruption, first we need to specify the factors that make interruption a burden. Horvitz et al. [21] describe a system that builds decision-theoretic models by asking users about their perceived interruptibility during a training phase. Ho & Intille [20] consider 11 factors that impact the perceived burden of interruption. The authors suggest that an exhaustive model of interruptibility should include a weighted sum of the factors.

Next we need to use context aware services to manage interruption. Abundant body of literature has studied the issue of context management for personal computers [13, 22, 9]. Baladauf et al. [6] presents a survey of context aware systems. The typical contexts included are: location, time, day, and proximity. In relation to interruption management, several researchers have proposed other meaningful contexts. Petersen [30] mentions the challenges to face when pervasive computing becomes a reality and a part of our everyday life. Godbole & Smari [15] consider three types of contexts namely relational, social and interruptee's cognitive context to solve the interruption problem.

5.3 Context Aware Systems

A context aware system is a computing resource with knowledge of its environment and its user's situation. Research in autonomic computing [38] recognizes the complexity involved with applying or interfacing such a system with human users. The problem arises when we expect the context aware system and the pervasive environment to combine into one intelligent environment. Ziebart et al's [38] work on Learning Automation Policies serves to solve this problem. In a context aware system, information will come in from many sources rather than only one or two streams of input. The Context Toolkit is a java based library that facilitates development and deployment of context-aware systems [12].

With context aware systems assumed and available, the next step is aware and adaptive services. In [10], context awareness is extended into the service oriented architecture. Our unavailability system will require such rich information sources to properly diagnose a given situation. Privacy is a topic that is closely related to personal unavailability. In [26], interpersonal relationships in regard to data privacy preferences have been addressed. A system called Lilsys, which reads motion, sound and door-closed-state has been constructed in [8] to build a qualitative measure of user unavailability. Also, automated preference control on mobile devices has been tackled in [7].

5.4 Interruption Associated with Mobile Devices

There have been several works on how to manage interruption at inopportune moments using smartphones. Yu et al. [37] define user preference, terminal capability, location, time, activity and so on as context dimensions for smartphones. In [31], the authors suggest that an interruption technology adapting its response considering a person's feelings is likely to improve people's experience with that technology. Godbole & Smari [15] survey the type and extent of desired information about the incoming cell phone call. Guzman et al. [17] studied the context information users consider when they make a call and also the context information they wish others consider when they receive a call. In [36], the authors group the strategies for interruption management by filtering calls based on caller's identity, situation and time, and, status message sharing e.g. current location, activity etc. As users tackle

Table 2. Comparison of Various Interruption Management Systems

Characteristics $\rightarrow$ **Research Works** $\downarrow$	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
Toninelli et al. [36]	Y	Y	N	Y	Y	N
Godbole & Smari [15]	N	N	N	Y	Y	N
Picard et al. [31]	N	N	Y	N	Y	N
Bailey et al. [5]	N	N	Y	Y	Y	N
Ho & Intille [20]	N	N	Y	Y	Y	Y
Mark et al. [27]	N	N	N	Y	Y	N
Dekel et al. [11]	Y	Y	N	Y	Y	N
Guzman et al. [17]	Y	N	N	Y	Y	N
Khalil & Connelli [25]	Y	Y	N	N	Y	Y
Marti & Schmandt [28]	Y	Y	N	N	N	Y
Our System	Y	Y	Y	Y	Y	Y

interruption by taking some actions themselves, Toninelli et al. [36] suggest that the intelligent system should learn how the users act in some situations, learn from them and later take actions like them.

The aforementioned research works give us a solid basis for (i) which context needs to be considered, and (ii) how to evaluate such context. However, the chief distinguishing aspects of our work are (i) system architecture and prototype implementation with performance evaluations, and (ii) identification of desirable characteristics of the problem solution. In Table 2, we present a comparison of different interruption management system against the desirable characteristics.

6 Conclusion

In this paper, we have presented the design, development and evaluation of an intelligent interruption management system. The system architecture considers context information and user preferences and automatically filters out interruptions for mobile devices. We also presented a prototype case study that implements the system architecture. The system is fully analysed and the performance evaluations indicate that it is efficient to run within the constraints of a handheld device.

We plan to extend our work with additional features. The caller can be notified of the receiver's current state if s/he is not picking up. The receiver may not want to disclose this information to everyone. In some cases, s/he might just want the caller to know that s/he is "busy", wherein the other cases such as to a spouse s/he would like to inform the caller specifically of his/her current state. Again, this information can be passed to the caller in a simple text message or there can be a user interface for the caller in our application where this information is viewed. Secondly, receiver can inform the caller when to try calling again. Acquiring information from the user's task scheduler, our system can know when the current task is going to finish and notify the caller accordingly. In some cases, the receiver may just fail to notice that there is a call. In that case, the system can encourage the caller to try again instantaneously.

We also plan to formalize the model for unavailability which takes into account context-aware services such as location based services. As a part of our goal, we are currently working toward a mathematical formulation of Cost of Interruption (COI). We also like to explore possible applications of our system in different application domains from cell phones to instant messaging, email clients, and social networking. These are some areas which operate by interrupting a user and we plan to incorporate our unavailability feature to them so that the cost of interruption is kept to a minimum.

Acknowledgements. We thank Karl Stamm for his initial work on interruption management described in [35].

References

1. Adamczyk, P.D., Bailey, B.P.: If not now, when? the effects of interruption at different moments within task execution. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Vienna, Austria, pp. 271–278 (2004)
2. Adamczyk, P.D., Iqbal, S.T., Bailey, B.P.: A method, system, and tools for intelligent interruption management. In: Proceedings of the 4th International Workshop on Task Models and Diagrams, pp. 123–126 (2005)

3. Ahamed, S.I., Sharmin, M., Ahmed, S.: A Risk-aware Trust Based Secure Resource Discovery (RTSRD) Model for Pervasive Computing. In: Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications, pp. 590–595 (2008)
4. Bailey, B.P., Konstan, J.A.: On the need for attention aware systems: Measuring effects of interruption on task performance, error rate, and affective state. Journal of Computers in Human Behavior 22(4), 709–732 (2006)
5. Bailey, B.P., Iqbal, S.T.: Understanding changes in mental workload during execution of goal-directed tasks and its application for interruption management. ACM Trans. Comput.-Hum. Interact. 14(4), 1–28 (2008)
6. Baldauf, M., Dustdar, S., Rosenberg, F.: A Survey on Context Aware Systems. International Journal of Ad Hoc and Ubiquitous Computing 2(4), 263–277 (2007)
7. Bayley, C., Jernigan, C., Lin, J., Shu, J., Wright, C.: Talk Android (2008), `http://www.talkandroid.com/android-forums/android-market-reviews/495-locale.html`
8. Begole, J., Matsakis, N.E., Tang, J.C.: Lilsys: Sensing Unavailability. In: Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, pp. 511–514 (2004)
9. Brown, P.J.: The Stick-e Document: a framework for creating context-aware applications. Electronic Publishing, Palo Alto (1996)
10. Conlan, O., Power, R., Higel, S., O'Sullivan, D., Barrett, K.: Next generation context aware adaptive services. In: Proceedings of the 1st International Symposium on Information and Communication Technologies, pp. 205–212 (2003)
11. Dekel, A., Nacht, D., Kirkpatrick, S.: Minimizing mobile phone disruption via smart profile management. In: Proceedings of the 11th International Conference on Human-Computer interaction with Mobile Devices and Services, Bonn, Germany, pp. 1–5 (2009)
12. Dey, A.K.: Enabling the use of context in interactive applications. In: Extended Abstracts on Human Factors in Computing Systems, CHI 2000, The Hague, The Netherlands, pp. 79–80 (2000)
13. Dey, A.K., Salber, D., Abowd, G.D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. The Human-Computer Interaction (HCI) Journal, 97–166 (2001)
14. Dey, A., Mankoff, J., Abowd, G., Carter, S.: Distributed mediation of ambiguous context in aware environments. In: Proceedings of the 15th Annual ACM Symposium on User interface Software and Technology, Paris, France, pp. 121–130 (2002)
15. Godbole, A., Smari, W.W.: A Methodology and Design Process for System Generated User Interruption based on Context, Preferences, and Situation Awareness. In: IEEE International Conference on Information Reuse and Integration, pp. 608–616 (2006)
16. Grandhi, S.A., Schuler, R.P., Jones, Q.: To answer or not to answer: that is the question for the cell phone users. In: Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems, pp. 4621–4626 (2009)
17. De Guzman, E.S., Sharmin, M., Bailey, B.P.: Should I call now? Understanding what context is considered when deciding whether to initiate remote communication via mobile devices. In: Proceedings of Graphics Interface 2007, Montreal, Canada, pp. 143–150 (2007)
18. Henricksen, K., Indulska, J.: A software engineering framework for context-aware pervasive computing. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications (PerCom), pp. 77–86 (2004)

19. Henricksen, K., Indulska, J.: Modeling and using imperfect context information. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, pp. 33–37 (2004)
20. Ho, J., Intille, S.S.: Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Portland, Oregon, USA, pp. 909–918 (2005)
21. Horvitz, E., Koch, P., Apacible, J.: BusyBody: creating and fielding personalized models of the cost of interruption. In: Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, pp. 507–510 (2004)
22. Hull, R., Neaves, P., Bedford-Roberts, J.: Towards situated computing. In: Proceedings of International Symposium on Wearable Computers (1997)
23. Iqbal, S.T., Bailey, B.P.: Leveraging characteristics of task structure to predict the cost of interruption. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Montréal, Québec, Canada, pp. 741–750 (2006)
24. Jiang, X., Chen, N.Y., Hong, J.I., Wang, K., Takayama, L., Landay, J.A.: Siren: Context aware Computing for Firefighting. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 87–105. Springer, Heidelberg (2004)
25. Khalil, A., Connelly, K.: Improving cell phone awareness by using calendar information. In: Proceedings of INTERACT, Rome, Italy (2005)
26. Lederer, S., Mankoff, J., Dey, A.K.: Who wants to know what when? Privacy preference determinants in ubiquitous computing. In: Extended Abstracts on Human Factors in Computing Systems, CHI 2003, Ft. Lauderdale, Florida, USA, pp. 724–725 (2003)
27. Mark, G., Gudith, D., Klocke, U.: The cost of interrupted work: more speed and stress. In: Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems, Florence, Italy, pp. 107–110 (2008)
28. Marti, S., Schmandt, C.: Giving the caller the finger: collaborative responsibility for cellphone interruptions. In: Extended Abstracts on Human Factors in Computing Systems, CHI 2005, Portland, OR, USA, pp. 1633–1636 (2005)
29. McCrickard, D.S., Chewar, C.M., Somervell, J.P., Ndiwalana, A.: A model for notification systems evaluation—assessing user goals for multitasking activity. ACM Trans. Comput. Hum. Interact. 10(4), 312–338 (2003)
30. Petersen, S.A., Cassens, J., Kofod-Petersen, A., Divitini, M.: To be or not to be aware: Reducing interruptions in pervasive awareness systems. In: Proceedings of the Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies UBICOMM, pp. 327–332 (2008)
31. Rosalind, P.W., Karen, L.K.: Relative subjective count and assessment of interruptive technologies applied to mobile monitoring of stress. International Journal of Human-Computer Studies 65(4), 361–375 (2007)
32. Savioja, P.A.: Necessary Interruptions? Seminar on User Interfaces and Usability, HUT, Sober IT, pp. 121-900 (2004)
33. Sharmin, M., Ahamed, S.I., Ahmed, S., Li, H.: SSRD+: A Privacy-aware Trust and Security Model for Resource Discovery in Pervasive Computing Environment. In: Computer Software and Systems Conference, pp. 67–70 (2006)
34. Spira, J.B., Feintuch, J.B.: The Cost of Not Paying Attention: How Interruptions Impact Knowledge Worker Productivity, Basex (2005)
35. Stamm, K., Ahamed, S.I., Madiraju, P., Zulkernain, S.: Mobile Intelligent Interruption Management (MIIM): A Context Aware Unavailability System. In: Proceedings of the 25th Annual ACM Symposium on Applied Computing, Sierre, Switzerland (2010)

36. Toninelli, A., Khushraj, D., Lassila, O., Montanari, R.: Towards Socially Aware Mobile Phones. In: 7th International Semantic Web Conference (2008)
37. Zhiwen, Y., Xingshe, Z., Daqing, Z., Chung-Yau, C., Xiaohang, W., Ji, M.: Supporting context-aware media recommendations for smart phones. IEEE Pervasive Computing 5(3), 68–75 (2006)
38. Ziebart, B.D., Roth, D., Campbell, R.H., Dey, A.K.: Learning Automation Policies for Pervasive Computing Environments. In: Proceedings of the Second International Conference on Automatic Computing,, pp. 193–203. IEEE Computer Society, Washington (2005)
39. International Telecommunication Union, http://www.itu.int/newsroom/press_releases/2009/39.html
40. University of Michigan News Service, http://www.ur.umich.edu/0607/Apr02_07/02.shtml
41. Disruptive communication and attentive productivity, http://www.iii-p.org/research/disrupt_comm_report_v2.pdf
42. 148 Apps.biz, http://148apps.biz/app-store-metrics
43. Bureau of Labor Statistics, http://www.bls.gov/