Designing Architecture of a Rule-Based System for Managing Phone Call Interruptions

Iqbal H. Sarker
School of Software and Electrical Engineering,
Swinburne University of Technology, Melbourne, Australia.
msarker@swin.edu.au

Alan Colman
School of Software and Electrical Engineering,
Swinburne University of Technology, Melbourne, Australia.
acolman@swin.edu.au

Muhammad Ashad Kabir
School of Computing and Mathematics,
Charles Sturt University,
NSW, Australia.
akabir@csu.edu.au

Jun Han
School of Software and Electrical Engineering,
Swinburne University of Technology, Melbourne, Australia.
jhan@swin.edu.au

UbiComp/ISWC'17 Adjunct, September 11-15, 2017, Maui, HI, USA

ACM ISBN 978-1-4503-5190-4/17/09.
https://doi.org/10.1145/3123024.3124562

Abstract
Now-a-days, mobile phones are considered to be "always on, always connected" but mobile phone users are not always attentive and responsive to incoming phone calls. Incoming call notifications such as ringing at an inopportune moment (e.g., meeting) can cause interruptions for both the users and the surrounding people. In this paper, we present a *system architecture* for managing call interruptions according to individual's *call response behavioral rules*.

Author Keywords
System architecture; Phone call interruptions; Mobile data mining; Rule discovery; Context-aware; Call response behavior; Personalization.

ACM Classification Keywords
H.2.8 [Database Applications)]: Data mining; H.3.4 [Systems and Software)]: User profiles and alert services

Introduction
Now a days, mobile phones have become part of our life. The number of mobile cellular subscriptions is almost equal to the number of people on the planet [11]. The explosion in the number of cell phones has made them the most ubiquitous communication devices and are considered to be "always on, always connected". However, the mobile phone users are not always attentive and responsive to incoming

communication in the real life. The incoming calls sometimes cause interruptions for both the user and the surrounding people of the user. Such interruptions may create embarrassing situations not only in an official environment, e.g., meeting, but also affect in other activities like examining patients by a doctor, driving a vehicle, etc. These kind of interruptions may reduce worker performance, increase errors and stress in a working environment [11]. In this work, we aim to design a system *architecture* to manage such interruptions according to the *call response behavioral rules* of individual mobile phone users.

A number of authors have studied rule-based call interruption management systems. For example, Khalil et al. [8] use calendar information to infer user's activity and to automatically configure cell phones accordingly to manage interruptions. Seo et al. [15] propose a context-aware configuration manager for smartphones PYP (Personalize Your Phone) to block a phone call without bothering the user. Dekel et al. [3] design an application to minimize mobile phone disruptions. Kabir et al. [9] propose a socially aware phone call application to minimize call interruptions. An intelligent context-aware interruption management system has designed by Zulkernain [19]. However, the main drawback of these systems is that the rules used in applications are not *behavior-oriented* and *personalized* rather users need to specify the rules manually.

In this paper, we address this issue and design a *system architecture* for managing call interruptions according to individual's behavioral rules. The system utilizes individual's mobile phone data, e.g., context logs, and discovers individual's *call response behavioral rules* based on multi-dimensional contexts such as time-of-the-week, user's social situation, location, social relationship between caller and callee. As the behaviors of different individuals are not identical in the real life, behavioral rules may differ from user-to-user according to their unique behavioral patterns. Such behavioral rules make the system intelligent for managing call interruptions on mobile phones.

Architecture Overview

This section presents our rule-based system architecture for managing call interruption. As individual's phone call response behavior differs from user-to-user, our system takes individual's mobile phone data as input and discovers individualized *call response behavioral rules* based on multi-dimensional contexts. The system consists of three layers - context processing layer, rule generation layer and rule management layer. Figure 1 shows an overview of our system architecture.

(I) Context-Processing Layer:
Context Log Data
Recent advances in mobile devices and their sensing capabilities have enabled the collection of rich contextual information and mobile device usage records through the device logs, e.g., context logs [16, 20]. The main characteristic of such kind of log data is that it captures the actual behavior of the users in different contexts.

Context Analysis
We identify four different types of contexts for modeling call response behavior of individual mobile phone users in order to manage call interruptions. These are:

Temporal context - As each phone call activity is associated with a timestamp (e.g., 2015-04-25 08:35:55), temporal context plays a primary role to model individuals' call response behavior. Halvey et al. [6] have shown through the analysis of a large sample of user data that time-of-the-week is an important factor in modeling user behavior in a mobile-Internet portal. As exact time is not informative, we

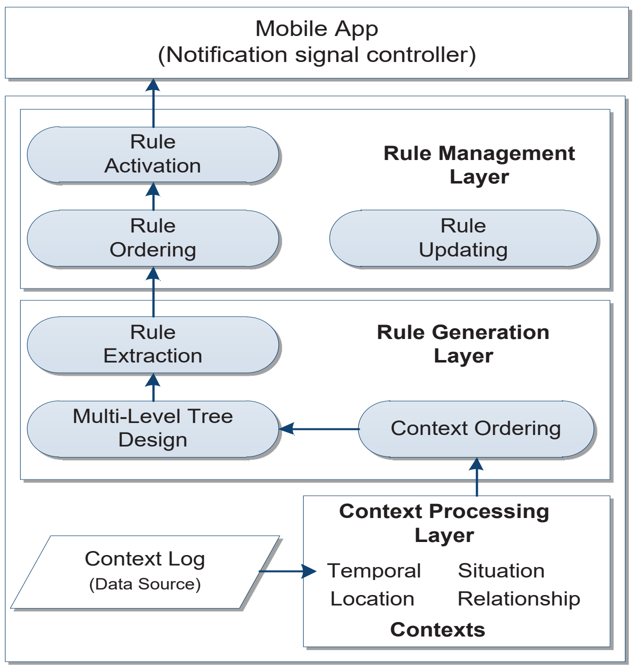

Figure 1: An overview of the system architecture

use behavior-oriented time segments [17] as the basis for exploratory rule induction.

Social situation - People are well differ from each other in how they respond to incoming phone calls during various events. One individual may be happy to answer the incoming calls during an event whereas another may not [13]. Even a particular individual may respond differently subject to what type of event is occurred [3]. For example, an individual's call response during a 'professional meeting' may be well different from her response during a 'lunch-break' event.

Social relationship - The social relationships such as family, friend, colleague, unknown, between the caller and callee, have a strong influence on users to make call handling decision [9]. For example, a user typically rejects an incoming call during an event 'official meeting', however, he answers if the call comes from his boss. Thus, we need to take into account social relationships in modeling user's call response behavior.

Location - Location-based services are one type of context-aware service that takes advantage of location-sensing technologies and location information. Xu Sun [14] showed that location information can be used to improve mobile user experience by providing relevant mobile services. Advances in location tracking systems such as GPS and history have made it feasible to obtain location information [7].

(II) Rule Generation Layer:
In order to design a rule-based call interruption management system, we need a complete set of "effective" (*non-redundant*, and *reliable*) rules of individuals based on above multi-dimensional contexts. The techniques commonly used to discover rules are classification rule learner [12] and association rule learner [1]. The classification rule learner is low reliable [10] and cannot ensure the high predictive accuracy in discovered rules [4]. On the other-hand, the association rule learner produces a huge number of redundant rules [5]. Therefore, we do not use these rule learning techniques directly in our system. In order to discover individual's effective behavioral rules, we employ our rule discovery approach [18] that produces not only the general rules but also their specific exceptions for a given confidence threshold.

Multi-Level Tree Design

We first design a context-tree by taking into account the precedence of contexts (context ordering) [18]. The tree is built top-down, starts from a root node and involves partitioning the data into subsets that contain instances with homogeneous values. Each node of the tree contains a particular *behavioral class* and it's corresponding *confidence* value. The confidence value of each node is calculated by counting the support value of associated contexts. The tree is incrementally developed according to the number of contexts. The final result is a multi-level tree with various nodes, associated contexts and node's confidence value. After this, we simplify the tree by bottom-up pruning [18]. We do such pruning in order to avoid redundancy in rule generation.

Rule Extraction

Once the tree has been designed, rules are extracted from the tree by traversing from root to decision nodes. The node that satisfies the user-specific threshold, is considered as a decision node. The setting of this threshold for creating rules will vary according to an *individual's preference* as to how interventionist they want the call-handling agent to be [17]. For example, one person may want the agent to reject calls where in the past he/she has rejected calls more than, say, 80% of the time - that is, at a threshold of 80%. Another individual, on the other hand, may only want the agent to intervene if he/she has rejected calls in, say, 95% of past instances. Unlike decision tree [12], we take into account both interior and leaf nodes as decision nodes if they satisfy the threshold. As a result, we produce not only the *general rules* but also their exceptions, i.e.,*specific rules*.

(III) Rule Management layer:

Rule Ordering

As we produce both general rules and their specific exceptions, this module is responsible for ordering the discovered rules to be selected in runtime according to rule's type. Given two rules $A \rightarrow C_1$ and $AB \rightarrow C_2$, we call the latter more specific than the former or the former more general than the latter. In such cases, the priority of the specific rule is higher than the general rule while selecting rules to execute in run time. Let's consider the following two rules $R1$ and $R2$ of user X -

$R1 : situation \rightarrow meeting \Rightarrow behavior \rightarrow Reject$ (confidence=82%)

$R2 : situation \rightarrow meeting, relationship \rightarrow boss$ $\Rightarrow behavior \rightarrow Accept$ (confidence=100%)

Rule $R1$ states that the user *rejects* most of the incoming calls (82%) when he is in a meeting. Similarly, rule $R2$ states that the user always (100%) accepts his boss's calls though he is in a meeting. Hence, the rule $R1$ is a general rule and $R2$ is an exception of the rule $R1$. However, in this case, rule firing priority of $R2$ is higher than $R1$. General rules play a vital role to model behavior if no specific rule with related contexts is found. For instance, calls from friends, family, unknown, etc.

Rule Activation and Updating

Once the rules are ready for using, users can select rules to be fired in application according to their interests. For this, the newly discovered rules are presented to the users in a human-readable format to get their consent. Users can activate the rules, which they think are needed to be fired, for managing call interruptions. The other inactivated rules are re-processed during the next iteration of the mining process. The rules are constructed and updated weekly as we take into account weekly pattern for producing rules. We do this when the phone is idle and not in use so that the application does not directly affect users' mobile experience. The phone is idle whenever it is charging, there are no foreground applications, and the battery level is at least 80% [16].

Notification Signal Controller

Ringing mode is a common feature of mobile phones for notifying incoming calls [2]. According to [2, 8] mobile phones have four states to controlling call notification signals such as normal mode, silent mode, vibrate mode and phone off (inactive). A phone plays sounds when the phone is in normal mode. In vibrate mode, sound is suppressed, but the phone still vibrates. In silent mode, there is no sound or vibration (but the screen or flashing light still activates). To avoid call interruptions, the mobile phone users typically change these configurations, e.g., normal to silent / vibrate. For instance, most of the users put their phone in a quiet state (i.e., Silent or Vibrate) when they did not want the phone to interrupt them and would return to the mode where they could feel or hear notifications afterwards to maintain awareness of notifications [2]. As the behavior of different individuals are not identical in the real life, such notification configurations may differ from user-to-user. Therefore, in order to assist them, the produced *call response behavioral rules* from individual's mobile phone data can be used to control such signals dynamically.

Discussion

To the best of our knowledge, this is the first study of managing phone call interruptions based on individual's *behavioral rules*. Recently, Mehrotra et al. [10] design a rule-based system for managing mobile apps notification. Though the contexts, user behavior type and the purpose are different, however, we can make a comparative discussion as both systems are automatic and rule-based. First, for generating rules they use temporal context statically segment time into arbitrary categories (e.g., Morning, Afternoon, Evening, etc.). However, it's very difficult to assume such time-slots to capture individual's phone call response behavior, as users' behaviors are not identical in the real life. Second, in order to produce rules, they manually choose the number of features (e.g., activity, location) and eventually ignore a number of contexts for discovering association rules. However, such type of manual selection of contexts may not suitable for phone call response behavioral rules as we have no prior knowledge about individual's behavior. Third, they do not take into account redundancy analysis in producing rules as it is well-known that association rule learner produces a huge number of redundant rules [5]. In our system, we take into account these issues and produce effective behavioral rules (e.g., general and their exceptions) based on multi-dimensional contexts [18]. To produce rules, we use behavior-oriented time segments [17] according to individual's unique behavioral patterns rather than choosing arbitrary time segments. We argue that our rule-based system would be able to discover more accurate rules based on multi-dimensional contexts in mobile phone datasets and be able to manage call interruptions according to such behavioral rules of individuals.

Conclusion

In this paper, we have presented a system architecture that uses call response behavioral rules of an individual in order to manage phone call interruptions. We believe that our approach opens a promising path for future research on intelligent context-aware software systems utilizing log data.

REFERENCES

1. Agrawal et al. 1994. Fast algorithms for mining association rules. In *VLDB*.
2. Chang et al. 2015. Investigating Mobile Users' Ringer Mode Usage and Attentiveness and Responsiveness to

Communication. In *HCI with Mobile Devices and Services*. ACM.

3. Dekel et al. 2009a. Minimizing mobile phone disruption via smart profile management. In *HCI with Mobile Devices and Services*. ACM.
4. Freitas et al. 2000. Understanding the crucial differences between classification and discovery of association rules: a position paper. *ACM SIGKDD Explorations Newsletter* 2, 1 (2000).
5. Fournier-Viger et al. 2012. Mining top-K non-redundant association rules. In *International Symposium on Methodologies for Intelligent Systems*. Springer.
6. Halvey et al. 2006. Time based patterns in mobile-internet surfing. In *Human Factors in computing systems*. ACM.
7. Han et al. 2014a. Adaptive content recommendation for mobile users. *Pervasive and Mobile Computing* 13 (2014).
8. Khalil et al. 2005. Improving cell phone awareness by using calendar information. In *Human-Computer Interaction*. Springer.
9. Kabir et al. 2014b. User-centric social context information management: an ontology-based approach and platform. *Personal and Ubiquitous Computing* (2014).
10. Mehrotra et al. 2016a. PrefMiner: mining user’s preferences for intelligent mobile notification management. In *UbiComp*. ACM.
11. Pejovic et al. 2014c. InterruptMe: Designing intelligent prompting mechanisms for pervasive applications. In *UbiComp*. ACM.
12. Quinlan et al. 1993. C4.5: Programs for Machine Learning. *Machine Learning* (1993).
13. Rosenthal et al. 2011a. Using decision-theoretic experience sampling to build personalized mobile phone interruption models. In *Pervasive Computing*.
14. Sun et al. 2009b. The role of spatial contextual factors in mobile personalization at large sports events. *Personal and Ubiquitous Computing* (2009).
15. Seo et al. 2011b. PYP: design and implementation of a context-aware configuration manager for smartphones. In *SmartApps’ 11*.
16. Srinivasan et al. 2014d. Mobileminer: Mining your frequent patterns on your phone. In *UbiComp*. ACM.
17. Sarker et al. 2016b. Behavior-Oriented Time Segmentation for Mining Individualized Rules of Mobile Phone Users. In *Data Science and Advanced Analytics (IEEE DSAA), Canada.*
18. Sarker et al. 2017. An Approach to Modeling Call Response Behavior on Mobile Phones Based on Multi-dimensional Contexts. In *Mobile Software Engineering and Systems, Argentina*.
19. Zulkernain et al. 2010. A Mobile Intelligent Interruption Management System. *J. UCS* (2010), 2060–2080.
20. Hengshu et al. Zhu. 2014. Mining Mobile User Preferences for Personalized Context-Aware Recommendation. *ACM TIST* (2014).