A Context-aware, Info-bead and Fuzzy Inference Approach to Notification Management

Kieran Fraser
ADAPT Center, School of Computer Science and Statistics
Trinity College Dublin
Dublin, Ireland
kieran.fraser@adaptcentre.ie

Bilal Yousuf
ADAPT Center, School of Computer Science and Statistics
Trinity College Dublin
Dublin, Ireland
yousufbi@scss.tcd.ie

Owen Conlan
ADAPT Center, School of Computer Science and Statistics
Trinity College Dublin
Dublin, Ireland
owen.conlan@scss.tcd.ie

***Abstract*— Managing the growing influx of notification data being pushed at mobile phone users is an increasing necessity in today's world of social media and ubiquitous computing. The challenge of effectively managing a diverse range of notification types, and analyzing their current context with relation to an individual user, is a difficult task. This paper proposes a notification management framework which aims to contextually deliver notifications at the peak opportune time of users, thus relieving them of distractions caused by contextually irrelevant notifications. The proposed design is implemented using an info-bead modeling approach and fuzzy inference system and the solution is evaluated through a comparison between simulated and expected results for two real-world notification data-sets. It was concluded that the solution is effective when rich data sources are available and the fuzzy inference system is adequately personalized to the user.**

Keywords—Context awareness; interruption management; notification management system; social networks; fuzzy logic; info-bead

I. Introduction

With the evolution of the social web and ubiquitous computing, there has been an unprecedented increase in the amount of notifications being pushed at mobile users [1]. Research has shown that incoming notifications distract the user severely from their current tasks, regardless of whether the notification is given immediate attention [2]. A significantly high influx in notifications has also been shown to resonate negatively with those receiving the notifications [3, 4]. Due to the rapid rise in mobile usage throughout the world, it has become normal for vast amounts of information to be at our fingertips 24 hours a day. It is being physically "pushed" at us multiple times a day and from multiple sources, commanding our attention and our time. Push notifications from various social media sites, email accounts and other mobile apps, compete continuously for precedence. The problem with this scenario is that it occurs in real-time, with no awareness or empathy for the consumer of the information [5]. The point is such that, while the information being sent may be something that the consumer wants to see, it may not be necessary for them to see it at the particular time the notification was "pushed" to them.

Within this paper, a framework is proposed to manage the incoming notifications of a user by delivering them in a contextually relevant manner. This involves the creation of an intelligent system which attempts to predict whether a user should receive a notification immediately or at some other contextually relevant time. In order to achieve this, both the incoming notification and the user are modeled (using the info-bead modeling approach) and the contextual relationship between them is analyzed resulting in an inferred delivery time for the notification (using fuzzy logic). Similar research in the area of contextual management of notifications has exploited combinations of user-defined rules [6], machine learning techniques [7, 8] and device sensor tendencies [9] to infer the current contextual relevancy of a notification to a user. Few of these approaches address the explicit attributes of the notification, such as the sender, subject and receiving application while doing so, and those that do assume this sensitive information is freely available. This paper proposes a novel info-bead and fuzzy inference solution to mobile notification management which aims to exploit a user's online persona to contextually deliver notifications to a user. In contrast with others, the proposed solution is also designed to function with incoming notification details abstracted in order to preserve the privacy of the sensitive information contained within a user's notification thus simulating a true real-world scenario. Fuzzy inference has yet to be applied as a solution in the domain of contextual management of notifications. The proposed system aims to deliver notifications, not simply when there is a break between tasks or a lull in an environments conversation level, but instead when the notification's attributes (subject, sender etc.) have contextual relevancy to a user's current state. Intercepting, contextually categorizing, and adapting notification delivery based on the current consumer need for privacy and minimal intrusion, is therefore the key motivating factor behind the proposed solution.

The remainder of this paper is structured as follows – Section II analyses the info-bead modeling approach, fuzzy inference systems and other similar research efforts in the domain of contextual notification management. Section III discusses the design and implementation of a novel info-bead and fuzzy inference Notification Management System (NMS). Section IV evaluates the results and investigates the limitations of the NMS. Finally, Section V summarizes the work and contribution.

II. Related Work

The info-bead user modeling (IBUM) approach [10, 11] is a component based software development approach whereby a user model is made up of atomic elements named info-beads which hold single attribute values of a user. A combination of info-beads linked together make up the attributes used to generate generic user models. As IBUM is a new modeling approach it has only been prototyped and evaluated in a limited number of domains with limited connections to external services. The proposed NMS of this paper aims to be the first to apply the IBUM approach in the domain of contextual notification management with a number of diverse external connections to social media services. This paper also addresses the issue that, in a real world scenario, full and unobstructed access to user data is not always available or welcome. Brusilovsky, Kobsa and Nejdl highlight an aspect of ubiquitous computing whereby user models in mobile devices need to be able to perform with uncertain or partial data [12]. The proposed NMS uses info-beads to create an abstract view of incoming notifications and users in order to limit the intrusion on privacy and remove the risks surrounding the storage of sensitive data.

An example application of the info-bead user model in practice is the social behavior analysis application implemented by Dim, Kufflick and Reinhartz-Berger in the Hecht archaeology museum which aimed to identify patterns of visitor behavior in the museum [10]. The museum is equipped with various proximity sensor technology which the info-bead user model leverages as sensor evidence data in order to infer a particular behavior type about the visitors in the museum. Three info-beads collect evidence data from the various sensors located around the museum - location, proximity, and azimuth. Data is gathered and inferred in these three info-beads and pushed onward to subsequent info-beads to determine visitor orientation at exhibits. This application limits itself to contextual data provided by the physical actions of visitors in the museum. A broader and more in-depth analysis of a user's behavior can be achieved however, with more specific data pertaining to an individual user. The harvesting of social media data, as demonstrated in this paper's NMS, can provide an additional snapshot of a user's contextual state as well as a history of habits and goals for the future.

In the domain of notification management, exploiting social media and user habits in addition to sensor technology is the aim of Corno, Russis and Montanaro [7]. They propose a modular system comprised of a core decision maker module which implements machine learning algorithms to identify who is to receive the notification, what is the best moment to deliver it, on which device is it shown and its method of notifying the user based on contextual information gleaned from the environment and the user. The decision maker is fed data from user and environment context modules as well as a habits module. This preliminary design attempts only to predict which device is to be selected for the notification delivery however, and the notification data-set used for training the algorithms and evaluating the result of the system is partly synthetic and assumes the data available from the notification is explicit. In contrast, the NMS proposed in this paper attempts to predict the most opportune time for the notification to be delivered to the user using a data-set entirely harvested from real users, of which, only an abstract derivation is used by the NMS to predict the delivery time (to protect privacy).

Pan et al. propose a modular system for intelligent push notifications which addresses the problem of notification management from a different angle [13]. The proposed solution ensures the sender and receiver contribute equally in identifying the current context in order to enable the notification be delivered intelligently. Within this design a sender would push their notification message along with a description of the desired context of the receivers to desired recipients. The description of the context is parsed and new rules are added to a "content-matching engine". Users subscribed to a particular topic also have their current context parsed and rules stored in a context buffer. By matching context-rules of users subscribed to the notification topic with the intended context-rules of the notification, provided by the sender, only relevant notifications are delivered to recipients. In this solution, it is assumed that user context is provided by "context providers" and that senders of notifications will provide a desired recipient context. The NMS proposed in this paper aims to offer a complete end-to-end solution whereby the context of the user is provided by harvesting social media data as opposed being dependent on third party providers.

Qin et al. propose a Do-Not-Disturb (DND) service [9] which uses machine learning techniques to identify the relationship between a user's current context and a DND status. Sensors are used to identify users' current activity such as sleeping, walking or watching TV and the algorithm identifies, based on previous experience (a training sample), whether the user is available or not. This system is limited by its reliance only on the availability of the user and not the contextual relevancy of the notification. While it's true the user may not want a notification while they sleep, depending on the context of the notification, they also may not need to see the notification when they wake up, if it's not relevant for another week for example. Equally, if the notification pertains to an emergency, perhaps the user should be awoken. Hence, in contrast, this paper's NMS relies on both the user's context, the incoming notification's context and the relationship between them to calculate whether or not to notify the user.

Kern and Schiele similarly use sensors to identify the current context of a user and mediate notification delivery [14], but they investigate the feasibility of achieving this with the limit of only evaluating the cost of the notification. For example, a notification would be very costly if it alerted a user during a lecture as it would distract them personally and also distract the class. However, it would be less costly if it interrupted them while they are commuting on a train with nothing to do. Hence, the proposed system uses personal and social "interruptability" to ascertain whether a notification should be delivered. In contrast, the system proposed in this paper uses the value of a notification as the evaluating factor. For example, a notification would be very valuable during a lecture if its subject was that of a current emergency and less valuable if the subject was a social event weeks away.

Classifying whether or not a notification is valuable enough to send to a user however, is a vague concept. Fuzzy logic deals with managing uncertainty in expert systems [15], hence a fuzzy logic controller is suited to the problem of notification management. Fuzzy expert systems have been most notably been implemented in the medical sector to aid in the diagnosis process of patients [16]. Within the realms of medicine the knowledge which is used to diagnose patients is generally uncertain as is the relationship between symptoms and diseases [17]. Linguistic terms are used to record the current physical states of patients for example, as they are sometimes difficult to express quantitatively. CADIAG-IV (Computer Assisted Diagnosis) is a medical consultation system which aids internal medicine by implementing fuzzy logic concepts to deal with the uncertain knowledge surrounding medicine. It uses fuzzy sets and membership functions to transform observed data and test results into linguistic variables. Medical knowledge is then expressed in the form of antecedent-consequent rules in a knowledge base. MedFrame [18] is one such application which implements this medical consultation system. The system is also composed of an explanation tool which illustrates how it came to a specific diagnosis. Similarly, within the domain of notification management, there is no clear divide between what constitutes a contextually relevant and important notification. For example, a notification sent from a particular sender could mean the notification is deemed important enough to disturb the user, however, if the sender is important but the subject is not contextually relevant, then perhaps the notification can be delivered later when it is more contextually relevant, despite the status of the sender. There is a degree of uncertainty surrounding the level of importance a subject would need to have in order for this scenario to occur. Uncertainty such as this can be managed through use of fuzzy membership functions and a heuristic knowledge base as demonstrated by CADIAG-IV.

Another example of this in practice is the fuzzy Decision Support System (DSS) suggested by Saleh et al. [19] which implements a Mamdani inference method to identify the individual patient risk status in treatments for breast cancer. Six input variables undergo *Fuzzification*, *Inference*, *Composition* and *Defuzzification* in order to obtain a subsequent output variable representing risk.

III. NMS Design and Implementation

The development of the Notification Management System (NMS) was split into three parts, which together formed an end-to-end solution: notification capture, notification uplift and delivery simulation. The decision to split the NMS into three parts was to enable the research to focus on the development of the intelligent framework which would be implemented in the NMS. This intelligent framework is comprised of an info-bead model implementing a Mamdani Fuzzy Inference System (FIS).

A. Notification Capture

The first step in the design process of a NMS is to intercept and redirect incoming notifications. In order to achieve this a mobile application, NAbsMobile, was developed which had access to users' incoming notifications as well as permission to read the multiple layers of data it provides. The objective of this application was to capture real-world notification data-sets. NAbsMobile was designed as an Android application using Android Studio. It was decided that in order to keep the privacy of data to a maximum it would be better to use the mobile application as a separate entity with which to capture the data necessary to test with, and then develop the intelligent framework in a desktop application where security and control could be exploited to a higher degree. Therefore, the proposed design of this mobile application doesn't include integration with the underlying NMS, the data is simply captured with it and the delivery of notifications is simulated later using the captured data-set. NAbsMobile was installed upon the mobile phones of two users (the number of test users was limited due to privacy constraints). It was also decided that, for the purposes of aiding the effectiveness of the heuristic rules to be implemented in the fuzzy controller, a certain amount of similarity should be present in both users' personas (which again limited the number of test users). Hence, user A was a student and user B a lecturer, both attending the same university and both with a working relationship in the same department. The two users were willing to have their notifications observed on an ongoing basis and, as a preliminary study into users' acceptance of a system

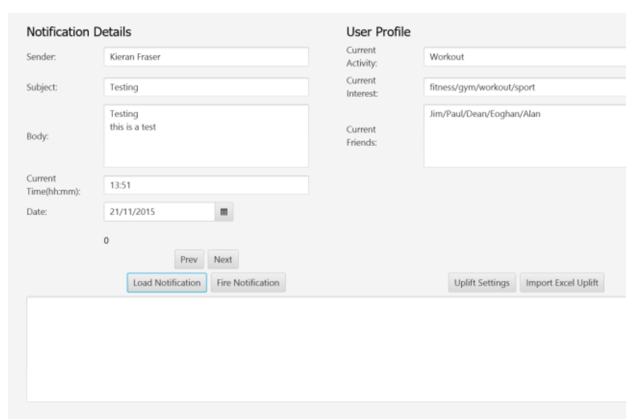

Fig. 1. NAbsUplift desktop application

monitoring and managing their notifications, they were also asked to rate the trustworthiness of the system. The functionality of the application included local logging of notification attributes, a means to view the captured notifications and a method of exporting the captured data-set.

B. Notification Uplift

In order to abstract away the sensitive data within the notifications and provide a means to map notification data to user attributes, a predefined set of terms was created and used to uplift the notifications. This set of terms was created by first studying the gathered notifications, and subsequently categorizing them in general terms, in a manner which described them at a higher level of abstraction and did not give away any personal information pertaining to the user.

This process was done individually by both users on their own dataset of notifications respectively in order to get an overlapping set of terms which covered all notifications in both data-sets (Fig. 2 and 3). It also ensured that the notification data gathered was only worked upon by the owner of the notification data which aided in keeping the intrusion of privacy to a minimum.

NAbsUplift was hence developed in order to maximize the privacy of the collected data-sets and prepare the data for simulation in the NMS. Simulation here meaning the notifications being redelivered to the user, simulating their original delivery. However, in the simulation, the notification detail will have been abstracted by NAbsUplift meaning the NMS will need to cope with less granular data. For example, a notification before uplift will contain the sender's explicit name, such as "John Doe", whereas the uplifted notification will have an abstract term, such as "close friend", as the sender. A snapshot of the user's current context on that day (via social media) is also available to the NMS at the point of simulation. The main functionality of NAbsUplift is comprised of an interface to allow a user to create and update a set of uplift terms for their notifications, a means to extract the notification data from an SQLite database and convert it to an easily editable format for uplift, and finally, an interface for tracking a user's personal "importance" rankings of the uplift term set. The list of rankings was necessary to gather from the users for the inference process within the FIS. Autonomously generating users' importance rankings was out of scope for this research however and, while the rankings were manually added by both users in this study, future work will explore deriving these values from habits and behaviors via social media personas or IoT sensors. Consequently, the NAbsUplift application enables the owner of the notification data-set to manually uplift their own notifications and rankings without external assistance, thus ensuring sensitive data is kept private (Fig. 1).

Sender		Application		Subject		Body		Date	
Family	10	Facebook	8	Social	5	Social	5	Ocassion	8
Close Friend	9	Gmail	8	Interest	4	Interest	4	Holiday	3
Colleague	7	Tinder	4	Work	8	Work	8	Not Significant	1
Stranger	4	Android	2	Android	2	Android	2		
Acquaintance	5	Flipboard	3	Family	10	Family	10		
Friend	7	Whatsapp	8						
Myself	1	LinkedIn	6						
Automatic	1	Viber	7						
		Skype	9						
		Twitter	5						
		Android SMS	2						
		Calendar	5						
		Run Keeper	4						

Fig. 2. User A uplift term set and importance ranking

Sender		Application		Subject		Body		Date	
Family	10	Facebook	10	Social	8	Social	8	Ocassion	5
Close Friend	8	Gmail	9	Interest	9	Interest	9	Holiday	5
Colleague	6	Tinder	1	Work	6	Work	6	Not Significant	5
Stranger	1	Android	3	Android	2	Android	2		
Acquaintance	3	Flipboard	2	Family	10	Family	10		
Friend	7	Whatsapp	10						
Myself	10	LinkedIn	3						
Automatic	2	Viber	6						
		Skype	7						
		Twitter	8						
		Android SMS	2						
		Calendar	9						
		Run Keeper	7						

Fig. 3. User B uplift term set and importance ranking

C. Delivery Simulation

NAbsDesktop, was developed for the final part of the study which was comprised of managing incoming notifications on behalf of the user. This involved simulating the delivery of incoming notifications using the uplifted notification data-sets provided by the two users and evaluating the resulting output (an inferred delivery time) of the NMS. The development of the NAbsDesktop application first involved the creation of a generic info-bead library with which to model the notification and user (the design of the implemented info-bead model is illustrated in Fig. 5). A number of external data sources also needed to be harvested for the system to ascertain contextual relevancy of a user. For example, using social media, such as a user's *Google Calendar*, a user's location, activity and who they're with at any given time, is accounted for. This of course is dependent on the engagement a particular user has with social media. For the purposes of this project it was assumed that a user had a high level of engagement with their *Google Calendar*, updated their schedule regularly, and added information such as the subject of the event and the people it engages with in the description of the event. These assumptions are quite specific, however, for the purposes of testing the info-bead model and the FIS, these assumptions are adequate. With additional social plugins it's possible that this information could be identified, but that was out of scope for this study.

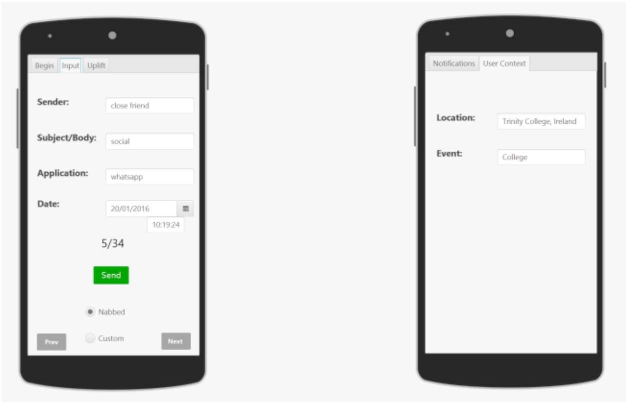

Fig. 4. NAbsDesktop simulation screen

It was then necessary to implement a FIS within each uniquely developed info-bead attribute in order to identify contextual relevancy of notifications. A *Mamdani* fuzzy inference method was implemented for this purpose. A knowledge base of heuristic rules was used in conjunction with fuzzy membership functions to describe the membership of linguistic variables and their relationships with the output crisp value which would determine a delivery time for a notification. The FIS was implemented using *jFuzzyLite*, a Java fuzzy logic control library [20]. *JavaFx* was then used for creating a GUI through which a notification, a user's current context and simulation results (delivery times for the notification) could be viewed (Fig. 4).

IV. Simulation Results and Evaluation

The NAbsMobile application was installed on both users', A and B, mobile devices for a period of approximately 10 weeks (68 days) from November 30th 2015 to February 5th 2016. To evaluate the effectiveness of the NAbsMobile application, the resultant data-sets captured from both users using the application were evaluated. Two data quality metrics were used to evaluate the data-sets [21].

1) Believability

$$min\ (T_S\ ,\ T_C\ ,\ T_E) \qquad (1)$$

T_S, T_C and T_E stand for a rating of trustworthiness (on a scale of 0 to 1). The two users were asked to give a trustworthy rating for: a) The data source (T_S) – The data source, in this case, being the NabsMobile application. This rating gauges how confident users are that the application logged their notifications correctly. b) A common standard (T_C) – A common standard in this domain is the Android system which displays notifications to users as they are delivered. This rating gives insight into the confidence users have in the current system to display their notifications correctly. c) Previous Experience (T_E) – This is a trustworthy rating based on users' experience of other similar systems (e.g. Chrome browser notifications) displaying their notifications correctly.

The believability of the notification data-set is the extent to which it is trusted by the users and relied upon as being credible. It also reflects the users' trust of the data with respect to common standards and previous experiences. If the data is not to be trusted by the users, then the uplift process could be compromised by a user deciding a notification-attribute value is wrong, and thus uplifting it incorrectly. Also, as inferences are being later made on the notification data captured by the NAbsMobile application, if it is considered untrustworthy at this point, then the NMS can't hope to satisfy a user by recommending delivery times for the notifications.

2) Completeness

$$1 - Cells_{\ empty}\ /\ Cells_{\ total} \qquad (2)$$

The completeness of the data-set is the degree to which data is not missing. $Cells_{\ empty}$ and $Cells_{\ total}$ are the number of empty cells and the number of total cells respectively in the resultant data-sets obtained from the NabsMobile application for both users. There were five key attributes that the NAbsMobile application attempted to log from users' incoming notifications: *sender*, *subject*, *body* (of the message), *application* and the *date/time* the notification was delivered. Completeness demonstrates whether or not the NAbsMobile application logged the notification data as expected.

The resulting output from the NAbsUplift application were two data-sets, one from each user.

B. User A data-set

This data-set had a total of 1982 notifications over the period of 68 days. The average number of notifications per day therefore amounted to 29.1, which translates to approximately 29 interruptions to the user throughout a single day. The result of the believability metric for this data-set was 0.6. This was

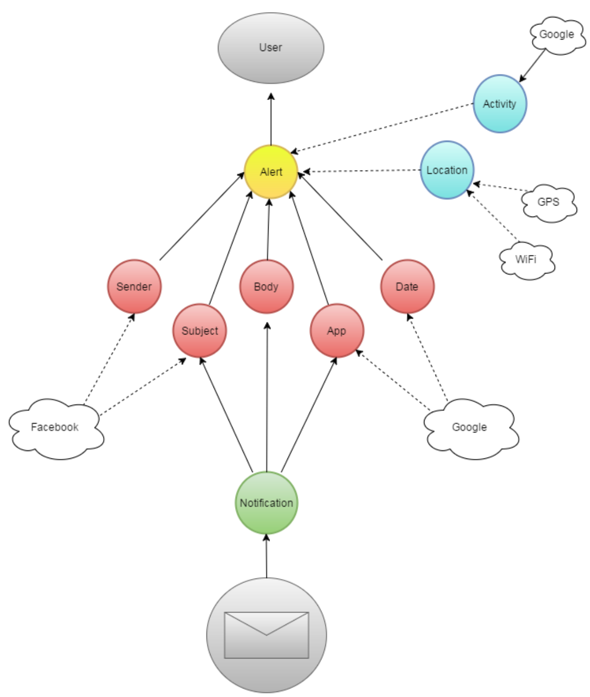

Fig. 5. Info-bead model functional design

the minimum value found of the three believability variables from (1) for this dataset rated by user A: $T_S = 0.8$, $T_C = 0.9$, $T_E = 0.6$. This value translates to the data source being 60% trustworthy, which means there is room for improvement on the transparency of the application which captures and reads users' notifications. Potentially this value indicates that, if a user is skeptical with regard to the competency of a system to collect their notification data without error, a user may not trust the full management of their notifications to a NMS. The result of completeness for this data-set is 0.99 (of a maximum value of 1). Equation (2) found that $Cell_{empty} = 98$ and $Cell_{total} = 9910$. The 98 empty cells were all due to an error in the *Sender* column of the data-set. It was found that some, but not all, of the *Facebook Messenger* notifications were being logged without a sender. The value of completeness for the data-set in total is good and ensures that the uplift process has maximum potential of being correct.

C. User B data-set

This data-set had a total of 1192 notifications over the period of 68 days. The average number of notifications per day amounted to 17. It must be noted that for this user, not all social media user accounts had push notifications active during the 68 day period, hence this would not be a complete picture of the number of notifications this user receives on a daily basis. The result of the believability metric for this data-set, is 0.5. This is the minimum value found of the three believability variables from (1) for this dataset rated by user B: $T_S = 0.6$, $T_C = 0.8$, $T_E = 0.5$. This value translates to the data source being 50% trustworthy, which is a further 10% less than the previous data-set, again highlighting the emphasis which needs to be placed on further transparency throughout the NMS, and also perhaps empowering the user to have a greater input into the delivery mechanism. There is an element of this already designed in the NAbsDesktop application, as the user must enter a ranking of the uplifted terms. However, this could be expanded. The result of completeness for this data-set is 0.99 (of a maximum value of 1). Equation (2) found that $Cell_{empty} = 61$ and $Cell_{total} = 5960$. The 61 empty cells were made up of 60 errors in the *Sender* column and 1 error in the *Subject* column. The errors in the *Sender* column are again due to *Facebook Messenger* and the error in the *Subject* column is simply due to an email sent without a subject line (which is not a data quality issue as it is a characteristic of this particular notification). The value of completeness for the data-set in total, is again, good for this data-set.

The method for evaluating the performance of the NMS was to select a number of interesting days from both users' data-sets, and compare the simulated results from the NMS with the expected results created by the two users. "Interesting days" are selected by analysing the diversity of a user's incoming notifications and activities over a period of 24 hours. As each user had to manually annotate whether or not the NMS made a correct delivery choice it was not practical to include all 68 days in the evaluation. Each notification is simulated using the NMS and the resulting delivery time chosen is recorded and compared to an optimally managed value set by the user. Any discrepancy between the expected and simulated result is then evaluated by the users and either accepted as an error or categorized as a permissible error. A permissible error is one which does not match the expected result, but also one which the user will accept in the context of the notification. Through this comparison two new metrics can be calculated:

1) *Strict Correctness Ratio (SCR)*

$$(correct_{total} / notification_{total}) \times 100 \qquad (3)$$

This is the percentage of correct notification deliveries made by the NMS over the total number of notification deliveries. This value illustrates the effectiveness of the system to correctly categorize the notifications in a manner such that replicates the user themselves. The higher the SCR percentage, the greater intelligence the system has.

2) *Permissible Correctness Ratio (PCR)*

$$((correct_{total} + permissible_{total}) / notification_{total}) \times 100 \qquad (4)$$

This is the percentage of the sum of correct deliveries and permissible errors over the total number of notification deliveries. This is a slightly more flexible metric and is used to judge whether the proposed NMS would satisfy users in a real world environment. The PCR is a good early indicator of acceptable notification delivery results. The results are illustrated in Figures 6 and 7.

The performance of the NMS in handling the notification data was different for both users, as the high levels of discrepancy in results demonstrates, suggesting a bias in the system. The evaluated results for user A's data-set were good with a high percentage of notifications being delivered at user-expected

Date	Correct Delivery	Permissible Errors	Undeniable Errors	Total Notif.	S.C. Ratio	P.C. Ratio
Jan 20th	21	9	4	34	**61.7%**	**88.2%**
Jan 21st	26	5	10	41	**63.4%**	**75.6%**
Jan 22nd	26	5	3	34	**76.4%**	**91.1%**
Jan 25th	18	2	5	25	**72%**	**80%**
Jan 26th	20	5	3	28	**71.4%**	**89.2%**
Jan 27th	20	0	2	22	**90%**	**90%**
Jan 28th	19	4	6	29	**65.5%**	**79.3%**
Jan 29th	26	4	2	32	**81.2%**	**93.7%**
Avg:	22	4.3	4.4	30.6	**72.7%**	**85.8%**

Fig. 6. NMS performance results for user A

Date	Correct Delivery	Undeniable Errors	Total Notif.	S.C. Ratio
Dec 2nd	4	23	27	**14.8%**
Dec 10th	4	31	35	**11.4%**
Dec 16th	7	11	18	**38.8%**
Avg:	5	21.7	26.7	**21.7%**

Fig. 7. NMS performance results for user B

contextually relevant times (Fig. 6). However, more research must be carried out to scale the system effectively as, in contrast with user A, relatively few of user B's notifications were being effectively managed (Fig. 7). Three reasons were proposed for the low percentage success rate of user B and should be investigated in future research. 1) An insufficient amount of data available in this particular user's *Google Calendar* - this highlights that if a NMS is to depend on social media, a user must have a baseline level of engagement with the social media sources used by the NMS in order for it to effectively manage notifications on their behalf. 2) The static ranking system (used to identify importance between uplift values and the user) was incapable of effectively modeling a dynamic user whose subject importance would realistically vary throughout the day – for example, a user might rank work as being highly significant during work hours of 9 to 5 but not at all important outside of those hours. 3) The static fuzzy membership functions (developed using heuristic knowledge) failed to effectively reflect this particular user's expected values.

Henceforth, if the above limitations are addressed, the info-bead model and fuzzy inference system is a viable option for the contextual management of notifications as illustrated by the results of user A.

V. CONCLUSION

The main goal of this project was to develop a NMS which would manage a user's notifications by delivering them in a contextually relevant manner. The proposed solution was to implement an info-bead modeling approach and fuzzy inference system to act as an intelligent framework between incoming notifications and the user. NAbsMobile, an Android application, was developed to log the details of incoming notifications. NAbsUplift was developed to abstract sensitive details of notifications to a term set and accumulate a list of user rankings. NAbsDesktop was developed to simulate the delivery of incoming notifications and run them through a NMS. Within the NMS, users and notifications were modeled using the info-bead modeling approach. A *Mamdani* Fuzzy Inference System was implemented as the inference mechanism of choice for the info-beads. After evaluation it was found that the NMS performed well for user A but less so for user B hence three proposed solutions were expressed for improving the performance of the system across all users and these three solution are to be investigated in future work. Finally, as the NMS was competent at managing the notifications of user A, the concept of combining the info-bead model with a FIS is concluded to be a potentially effective solution toward the problems surrounding notification delivery.

REFERENCES

[1] M. Pielot, K. Church and R. de Oliveira, "An In-situ Study of Mobile Phone Notifications," in *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services*, New York, NY, USA, 2014.

[2] S. T. Iqbal and E. Horvitz, "Notifications and awareness: a field study of alert usage and preferences," in *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, 2010.

[3] A. Sahami Shirazi, N. Henze, T. Dingler, M. Pielot, D. Weber and A. Schmidt, "Large-scale assessment of mobile notifications," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2014.

[4] J. E. Fischer, S. Reeves, S. Moran, C. Greenhalgh, S. Benford and S. Rennick-Egglestone, "Understanding mobile notification management in collocated groups," in *ECSCW 2013: Proceedings of the 13th European Conference on Computer Supported Cooperative Work, 21-25 September 2013, Paphos, Cyprus*, 2013.

[5] A. Mehrotra, V. Pejovic, J. Vermeulen, R. Hendley and M. Musolesi, "My Phone and Me: Understanding People's Receptivity to Mobile Notifications," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016.

[6] R. Etter, P. D. Costa and T. Broens, "A Rule-Based Approach Towards Context-Aware User Notification Services," in *2006 ACS/IEEE International Conference on Pervasive Services*, 2006.

[7] F. Corno, L. D. Russis and T. Montanaro, "A context and user aware smart notification system," in *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, 2015.

[8] T. Okoshi, H. Nozaki, J. Nakazawa, H. Tokuda, J. Ramos and A. K. Dey, "Towards Attention-aware Adaptive Notification on Smart Phones," *Pervasive Mob. Comput.*, vol. 26, no. C, pp. 17-34, Feb 2016.

[9] Y. Qin, T. Bhattacharya, L. Kulik and J. Bailey, "A Context-aware Do-not-disturb Service for Mobile Devices," in *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia*, New York, NY, USA, 2014.

[10] E. Dim, T. Kuflik and I. Reinhartz-Berger, "When User Modeling Intersects Software Engineering: The Info-bead User Modeling Approach," *User Modeling and User-Adapted Interaction*, vol. 25, no. 3, pp. 189-229, Aug 2015.

[11] T. Kuflik, Y. Mumblat and E. Dim, "Enabling Mobile User Modeling: Infrastructure for Personalization in Ubiquitous Computing," in *Mobile Software Engineering and Systems (MOBILESoft), 2015 2nd ACM International Conference on*, 2015.

[12] P. Brusilovsky, A. Kobsa and W. Nejdl, Eds., The Adaptive Web: Methods and Strategies of Web Personalization, Berlin, Heidelber: Springer-Verlag, 2007.

[13] Z. Pan, X. Liang, Y. C. Zhou, Y. Ge and G. T. Zhao, "Intelligent Push Notification for Converged Mobile Computing and Internet of Things," in *Proceedings of the 2015 IEEE International Conference on Web Services*, Washington, 2015.

[14] N. Kern and B. Schiele, "Context-Aware Notification for Wearable Computing," in *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, Washington, 2003.

[15] L. A. Zadeh, "The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems," *Fuzzy Sets Syst.*, vol. 11, no. 1-3, pp. 197-198, Jan 1983.

[16] N. H. Phuong and V. Kreinovich, "Fuzzy logic and its applications in medicine," *International Journal of Medical Informatics*, vol. 62, no. 2–3, pp. 165-173, 2001.

[17] T. Vetterlein and A. Ciabattoni, "On the (Fuzzy) Logical Content of CADIAG-2," *Fuzzy Sets Syst.*, vol. 161, no. 14, pp. 1941-1958, Jul 2010.

[18] B. Sageder, K. Boegl, K.-P. Adlassnig, G. Kolousek and B. Trummer, "The knowledge model of MedFrame/CADIAG-IV.," *Studies in health technology and informatics*, vol. 43, pp. 629-633, 1996.

[19] A. A. E. Saleh, S. E. Barakat and A. A. E. Awad, "A fuzzy decision support system for management of breast cancer," *IJACSA Editorial*, 2011.

[20] J. Rada-Vilela, *fuzzylite: a fuzzy logic control library*, 2014.

[21] L. L. Pipino, Y. W. Lee and R. Y. Wang, "Data Quality Assessment," *Commun. ACM*, vol. 45, no. 4, pp. 211-218, Apr 2002.