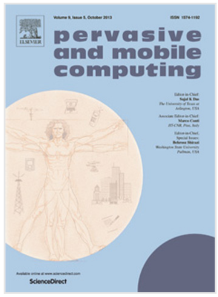

Field evaluation of context aware adaptive interfaces for efficient mobile contact retrieval

Athanasios Plessas, Vassileios Stefanis, Andreas Komninos, John Garofalakis

PII: S1574-1192(16)30034-7
DOI: http://dx.doi.org/10.1016/j.pmcj.2016.04.011
Reference: PMCJ 700

To appear in: *Pervasive and Mobile Computing*

Received date: 12 February 2014
Revised date: 30 November 2015
Accepted date: 28 April 2016

Field Evaluation of Context Aware Adaptive Interfaces for Efficient Mobile Contact Retrieval

Athanasios Plessas[1,2], Vassileios Stefanis[1,2], Andreas Komninos[3], John Garofalakis[1,2]
{plessas, stefanis}@ceid.upatras.gr, andreas.komninos@strath.ac.uk, garofala@cti.gr
1. Department of Computer Engineering & Informatics, University of Patras, Patras, Greece
2. Computer Technology Institute and Press "Diophantus", Rion, Patras, Greece
3. Department of Computer & Information Science, University of Strathclyde, Glasgow, UK

Abstract

Our paper discusses the implementation and field evaluation of a context-aware mobile contact retrieval application. We examine the performance of our underlying prediction algorithm in real world conditions and report on the suitability of our hybrid interface design, as a replacement for traditional contact retrieval interfaces (e.g. phonebook and recent call list). We find that users are best served by an alphabetical ordering of prediction matches and show that hybrid interface designs can provide a modest benefit in users' ability to find a contact, in cases of non-successful predictions. We also discuss users' alternative strategies for retrieval in such cases.

Keywords: Context Awareness, Mobile Personal Information Management, Mobile User Interfaces

1 Introduction

Modern mobile devices allow their users to generate and store large volumes of personal information items (e.g. contacts, photos, videos, calendar events, notes etc.), making their management a difficult task, due to the limitations (screen size, interaction means etc.) that these devices impose. While handheld devices are used, among others, to access the World Wide Web, as well as cameras, email clients and digital notebooks, they remain primarily communication devices, with applications like the phonebook or SMS still being the most frequently used [1].

While communication can be initiated by manually entering a telephone number, starting a new communication session most often involves the retrieval of contacts from one of the main communication information repositories, namely the phonebook, call log or related applications and widgets. As a result, searching for a contact in a phonebook or in a recent call list is a common task for mobile users [2]. It is noted in literature that mobile users' contact repositories continuously grow in size over time, hindering contact search and retrieval [3][4]. This effect has been attributed to users' reluctance to remove contacts, even if they are seldom used. As mobile communication is not only limited to phone calls and text messaging, but also involves social networking interactions, many mobile applications allow the merging of the classic mobile phonebook used for calling and texting, with the users' Social Networking Services (SNS) contact network. One's SNS contact network size is often larger than the contact network size of the classic mobile phonebook [5], exacerbating the retrieval problem in merged contact lists. Evidence of the ever-growing contact list size can be found in the comparison of findings from a recent study on the composition of the mobile phonebook [4], where it was found that the average size of the contact repository is 308 contacts. This number is considerably higher compared to previous studies (e.g. 92 contacts in [3] or 107 contacts in [6]).

It is commonly accepted that there is a need for new, more efficient mobile phonebook UIs. According to [7] the increase of the contact list size poses new design challenges and efficiency of contact retrieval is an important design driver. In [4] it is noted that despite the sharp increase of the phonebook size, the high number of 'unknown' contacts and the fact that over 92% of communications occur with ten contacts (80% with just five contacts), most mobile phonebooks still provide the A-Z list as the most common selection mechanism, impeding access the most frequently contacted people in one's life.

To address the retrieval problem, in previous work [8][9] we proposed a smarter UI based on a context aware algorithm for predicting the most probable contacts to be called at any time. The top *n* suggestions of this algorithm can be used in an adaptive dial list, facilitating contact retrieval [10]. The algorithm is not limited to the retrieval of contacts but it can be adapted for any information item type; however in our investigation we focus on mobile contacts as a timely and relevant problem domain. The prediction performance of the algorithm, which considers the contextual dimensions of frequency and recency of communication for each contact, was evaluated through simulation with very promising results (outperforming other proposed outgoing call prediction algorithms), using extracted call logs from real-world mobile use datasets. Our algorithm can be ran with settings to employ just the recency or frequency context, emulating the behaviour of classic UIs for contact retrieval, such as the recent calls list and the frequently used numbers list, present in most modern smartphones. During comparisons with execution using real-world datasets, we found that our approach highly improves prediction accuracy over classic UIs [8]. A further lab experiment [10] showed that our adaptive dial list offers lower retrieval times than traditional UIs (contact list and recent call list) when the contact is within the suggestions, however when it is not present in the list there seems to be a penalty of 2-3 seconds.

As such, the problem of enhancing contact retrieval speed seems addressable in two respects: Firstly, improving the algorithm upon which suggestions are made; Secondly, optimizing the manner in which suggestions are displayed and improving the fallback support, i.e. what is displayed after suggestions have been made, when none of the suggestions are not appropriate. In this paper, we explore the latter issue through alternative user interfaces, with the intent to facilitate the task of contact retrieval. In [2] it was found that the most important user requirement for adaptive call lists is the existence of fallback mechanisms when the recommendation algorithm fails. Having this in mind and having already evaluated the adaptive dial list against traditional UIs through simulation, we propose two hybrid interfaces that combine the suggestion list with traditional means of accessing the contact repository (recent call list and list of contacts), which act as fallback alternatives in case of prediction failure. The objective of our paper is twofold: First to examine the performance of the prediction algorithm under real world conditions, as apart from [2] there are no other reports in literature on real-world evaluations of such algorithms; Second, to evaluate the appropriate presentation interfaces, for suggestions and for fallback support. For this reason, we designed an Android application and performed both a lab and a field experiment, the results of which we present later on.

The rest of the paper is organized as follows: section 2 presents related work on this field, while section 3 describes our experimental application. Section 4 presents a lab experiment which guided the design of our application. The field experiment and the analysis of the results are outlined in section 5, while in the final section we discuss our conclusions and intentions for future work.

2 Related Work

Personal Information Management (PIM) has been an important research field for many years for the case of desktop computers. Technological advances in the area of mobile computing resulted in smart handheld devices that allow users to create, receive and store daily significant amounts of personal data, including contacts, photos, music, videos, notes, tasks, etc. [11]. However, the characteristics of these devices (such as small screen size, navigation and interaction/input modes) hinder search and retrieval [12], a problem that is exacerbated by the unwillingness of users to remove old items [13]. Myers et al. [14] stress the need for mobile users to access quickly the right information at the right time and highlight how important it is for PIM tools to help users accomplish their tasks efficiently. On the other hand, existing mobile PIM tools require extensive involvement of human users, making the management of personal information such as to-dos and contacts a time consuming process [12]. To aid with these issues, context can be taken into account when designing PIM systems in order to enhance them [15], since the information needs of users highly depend on it while on the go [16].

As contact lists are one of the most frequently used mobile apps [6], there have been several efforts to improve their usability, mainly by taking advantage of the user's context. Oulasvirta et al. [17] presented ContextContacts, an augmented contact list with situation cues. Similarly, Bentley et al. [18] created an augmented phonebook that allowed close friends and family to share a simple presence cue, such as motion, while Connecto [19] allows users to share information about location and their ringing profile. Another contact list replacement is Friendlee [20] that also permits users to share key aspects of their context. In [21], Bentley et al. describe their effort to "re-invent" the mobile phonebook, combining on-device communication with data from social networks. In [22] an alternative UI that filters information from social networking services and associates it with contacts is presented.

Nevertheless, while most of these efforts aim at creating a context-augmented contact list that will drive users' communication and increase awareness of each other, none of them is designed to facilitate the task of contact retrieval. Bergman et. al [3] found that unused or rarely used contacts that distract user's attention in such limited screen capacity account almost for half the contacts in their repositories and based on this remark proposed a prototype that reduces retrieval time significantly. Jung et al. [7], aiming at evolving the mobile contact list, included in their prototype an interface that showed the top 10 contacts (either selected manually or automatically based on communication frequency) on the top of the traditional contact list, in order to make it easier to access important contacts. This is the only case of a hybrid interface for the contact list focusing on contact retrieval that we were able to find in related literature. Unfortunately, while the authors report a positive acceptance of this UI from their users, they do not provide further details on its evaluation results. Other researchers focused on proposing alternative techniques for presenting large lists of items (e.g. contacts) on small screens to simplify user interaction and search tasks, such as a spiral layout [23] or a representation of the contact book as a social graph [24].

Several other works focus on predicting outgoing calls for the creation of dynamic/adaptive lists of contacts by analyzing features of mobile communication history. In [2] five communication dimensions (day of week, weekend/weekday spans, time of day, dayparts of a day, 1-hour slots of a day) are proposed as recommendation conditions. The approach in [25] takes into account frequency and regularity of communication, while another research team considered as important factors of the calling pattern the time period of day, the day of each call, and the "reciprocity" [26]. The same researchers developed also a probabilistic predictive model based on call departure and inter-departure times [27]. We have already shown in previous work [8] that our algorithm outperforms these approaches when applied to the NOKIA mobile data challenge dataset [28]. Sun et al. [29], based on our experimental results [30], proposed an almost identical adaptive algorithm to ours, for building a dynamic contact list, investigating both ranked-based and score-based fusion of the frequency and recency dimensions. Their findings were also encouraging and hint that ranked-based fusion may offer a 2% performance gain, though a statistical significance for this difference is not provided. All proposed interfaces include only top-k lists of contacts, acting as speed dial shortcuts and none of them (with the exception of that presented in [7] where detailed results are not provided) try to combine it with other representation means for lists of contacts, such as the traditional alphabetical contact list or the recent calls list, in order to function as an alternative to the traditional phonebook. We should also emphasize that with the exception of [2], all other approaches have been evaluated only over extracted call logs and not under real world conditions.

3 The "Calchas" Application

As we wanted to evaluate alternative interfaces that could replace the traditional means for accessing the mobile contact repository and facilitate the task of contact search and retrieval, we developed a Google Android application called Calchas (*kal'-huhs*\ - a seer in Greek mythology who predicted the fall of Troy). The Calchas application attempts to predict the contacts that users are more likely to call at any time, based on their communication behavior and presents these suggestions using hybrid interfaces. The application consists of three modules: *prediction module*, *user interface* and *logging module*.

3.1 Prediction Module

In the core of our application resides the context aware prediction algorithm, which analyzes the call log of the user and suggests the top-k most probable contacts to be called. The algorithm incorporates two context features, frequency and recency of communication and the first experimental results against real mobile datasets were very promising [8][9]. In brief, the prediction algorithm works as follows: each contact c in the mobile phonebook is represented as a context vector with two dimensions (F_c,R_c), corresponding to the communication frequency and recency with this contact respectively. Whenever the user opens the application, a recent portion of the call log, called training window (15 days in our case), is analyzed, including both ingoing and outgoing communication. For each contact, the scores F_c and R_c are calculated to reflect how frequent was the communication with this contact within the training window and how recent was the last communication with it within a recency window of 12 hours. A weighted combination of the two features provides the total score T_c, which is used to sort the contacts in descending order and pass the top-k of them to the user interface module as the best suggestions.

$$T_c = w_f * F_c + w_r * R_c$$

Based on previous experimental results, we set the weights of our prediction algorithm to w_f=0.5, w_r=0.5.

3.2 User Interface

Every time a user opens Calchas app, the prediction module produces the list with the suggested contacts and passes it to the UI module, which is responsible for presenting it to the user. In our approach, we have designed two hybrid interfaces combining the top-k suggestion list with traditional means of contact retrieval, such as the contact list and the list of recent calls (Fig. 1). In [2], as the recent call list was provided separately from the adaptive dial list, in case of prediction failure the users had to press a key in order to switch to an alternative retrieval interface. We assumed that, according to [3, 15], combining the fallback UI with the adaptive dial list could provide more efficient contact retrieval when the prediction algorithm failed, as less key presses would be required. Our two interfaces are designed as follows:

- *Interface A*: this interface combines the suggestions' list with the traditional contact list. The suggestions' list includes 7 contacts (k=7), a number that provides prediction results close to optimal according to our previous findings [8] and allows all suggestions to be visible in a single screenful, negating the need for scrolling in most modern smartphones (i.e. "above the *fold*", to use a term from web-design). The list is alphabetically sorted, according to the results of our lab experiment (presented in section 4). Lee et. al [2] report that the most important requirement elected by the users in their experiment was fast switching to a search page or a recent call list when the suggestion list failed. Hence, at the top of the interface there are two clickable icons allowing the user to leave the application and open directly the contact list or the recent call list of his/her phone and below these icons there is a search box, used for filtering contacts. Finally, below the list of suggestions resides the traditional contact list of the phone, stripped from the contacts included in the suggestion list.
- *Interface B*: this interface combines the suggestions list with the traditional recent call list. The design is similar to interface A, however in this case there is no search box and below the suggestions' list, the recent call list (stripped again from duplicates) is embedded.

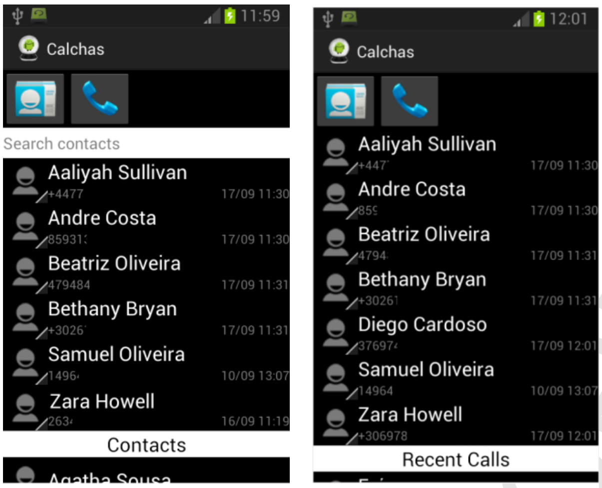

Figure 1. Calchas available interfaces: the list with suggested contacts (A-Z) followed by the embedded contact list (left) and the list with suggested contacts (A-Z) followed by the embedded list of recent calls (right)

3.3 Logging Module

The logging module implements a mechanism that tracks Calchas usage, which is necessary for the quantitative evaluation of the application and its interfaces. The anonymous statistics are saved in an SQLite database in the user's device and sent on demand to a remote server.

Whenever the application process comes to the foreground, a new usage session is recorded. In a usage session, the logging mechanism records the session start and end time, the interface that is active and the event that took place (if any). We define the following three events: a) the user clicks on a contact and starts an outgoing call, b) the user launches the default contact list and c) the user launches the list of recent calls. In all cases, the time that the event took place is recorded. Moreover, for the event (a) we record the full list of suggestions (contact id or the hashed number if the suggestion is not a contact), the place where the called contact appeared (list of top suggestions or embedded list of contacts or recent calls, according to the selected interface) and its frequency and recency scores. Finally, whenever the user performs an outgoing or receives an incoming call, an Android broadcast receiver records it in our database, creating a copy of the phone's call log.

4 Optimization of suggestion presentation

As stated earlier, one of our targets in this paper with respect to improving retrieval times with an adaptive dial list is to investigate the optimal presentation of suggestions. The top-k list of suggestions can be sorted either alphabetically or by probability score. In [31], the researchers found that there are no significantly faster zones in a mobile screen for the task of searching, when the user knows the name of the search target. In [10], we assumed that the presentation order of suggestions does not play an important role, since we did not find statistically significant differences regarding retrieval time for contacts that are in the bottom or in the top of the adaptive dial list (though with a small sample of participants). Here we discuss two short experiments aimed at validating the results presented in [31] and [10]. Our aim was to contribute to the limited available literature on the subject of user performance when searching through mobile lists and to use these results to determine how to best present our top-k suggestions on the users' mobile. The first experiment compared alphabetically sorted suggestion lists (single screen) with randomly ordered lists, in order to obtain a first impression of whether there is an advantage in alphabetical sorting. Our second experiment used participants' own contact lists to compare alphabetically sorted suggestion lists with probability ranked lists, the difference being here that we were using participants' own data and their implicit knowledge of the communication patterns with their contacts.

4.1 Alphabetically vs. randomly ordered lists

To examine if alphabetic ordering is more efficient for an interface of an adaptive list of contacts that occupies the length of one mobile phone screen, we organized a more extensive lab experiment, where 24 participants (12 male, 12 female, aged 19-45, from varied backgrounds) were presented with a mobile application consisting of an interface representing a top-k list (k=7) of randomly picked contact suggestions from an artificially generated list of 300 UK-English names (equal m/f distribution, mean character length m=11.74, sd=2.3). This dataset, although artificial, is very close to the contact name length data from 34 real users [9] (m=11.88, sd=5.3) hence does not present problems for the results' external validity. Using artificial names instead of participants' own contacts allowed us to control for the visual target length and hence prevent participants from possibly finding some contacts too quickly because of their short given names. The participants were asked to perform 12 search tasks, 6 of them with the suggestions alphabetically sorted and 6 with the suggestions sorted in a random order, which effectively simulated how a probability-ordered list might look. To derive the randomly ordered list, we assigned random fake probability scores to the contact set and presented the top-7 suggestions in the order of these scores. In each case, two of the search targets were at the top of the mobile screen, two at the middle and two at the lowest part. Half of the participants carried out the experiment starting with the alphabetical list, while the rest of them with the randomly ordered one and the search targets in each round were presented in a random sequence. The users were informed about the sorting method of the list before each search task, as we wanted to examine if this knowledge would help them find the target faster for the case of the alphabetical list. The application recorded the time to perform each search task and at the end of the experiment the participants were asked to declare their personal preference regarding the sorting method (alphabetically or by simulated probability score) of a top-k list of mobile contacts.

We analyzed the collected data to obtain the total average time required to accomplish a search task with each sorting method and the respective average times according to the position of the search target in the mobile screen. The statistical analysis of these results with paired-sample T-tests revealed that there is no statistically significant difference between the two sorting methods of the top-k list (when occupying one mobile screen) for the total mean time (m_{alpha}=1563.54ms, SD=395.62ms, m_{random}=1693.53ms, SD=297.88ms, t(23)=1.287, p=0.211) and for the cases that the target is in the top (m_{alpha}=1443.46ms, SD=616.39ms, m_{random}=1601.63ms, SD=396.33ms, t(23)=1.213, p=0.237) or in the middle of the mobile screen (m_{alpha}=1859.48ms, SD=687.65ms, m_{random}=1673.69ms, SD=563.47ms, t(23)=-0.995 p=0.330). However, a Wilcoxon signed rank test reveals a statistically significant difference when the target is in the lowest part of the screen (m_{alpha}=1387.69ms, SD=596.48ms, m_{random}=1805.29ms, SD=311.87ms, Z=-3.286, p<0.01) in favor of the alphabetical list. This result is expected, as users have some expectation of how far down the list a contact might be placed in the alphabetic interface, whereas this expectation cannot apply to a list that is ranked in a seemingly random manner. Finally, 17 participants preferred the alphabetically sorted interface (compared to 7 who preferred the simulated probability ranked one). To conclude, this lab experiment showed that the alphabetical list provides better results when the search target is in the lowest part of the mobile screen, while in all other cases there is no discernible difference and that participants show subjective preference for the alphabetic interface.

4.2 Alphabetically vs. probability ordered lists

Our previous experiment had the shortfall of using an artificial data set, to which the participants could not relate. This presented some problems with the validity of the results, given that a user searching in a real probability ordered list might be aware of the communication patterns with the contact that they wish to retrieve. For example the user might be aware that there have been other communications with other contacts since they last communicated with the desired contact, hence they might expect to find the desired contact lower in the list of suggestions. To address this shortcoming, we conducted a further experiment, using participants' own contact lists. For this, we built a custom version of our application and installed it on participants' own devices. This application version presents only the top-7 suggestions (as in the previous experiment) derived from each user's actual call log and contact list. We then asked each participant to find and click on a contact of our choice, from either the top, middle or bottom part of the list, using the first the alphabetic and then the probability ordered interface. To prevent any learning effects, which might arise if participants saw the lists multiple times, each participant was thus asked to pick a contact that we dictated from one position on the list (e.g. top) using one interface, and then switched interfaces and was asked to pick another contact again from the same position on the list (i.e. again from the top). We measured the time taken to retrieve each contact and the subjective participant interface preference. We used a homogeneous set of 60 users (20 picking from the top of the lists, 20 from the middle and 20 from the bottom), all university students aged between 22-28 (m=24.7, sd=3.2) and 26 were female. Overall the participants took longer to find and select contacts with the alphabetic interface (m=2158.75ms, SD=1031.10ms) than with the probability ordered interface (m=1887.17, SD=589.09ms), though the difference is not statistically significant (Wilcoxon signed-rank test, Z=-0.743, p>0.05). We split the participants' data according to the target position on the list (top, middle and bottom). Again, Wilcoxon signed-rank tests did not reveal any statistically significant differences for the top (m_{alpha}=1682.78ms, SD=564.40ms, m_{prob}=1996.56ms, SD=619.56ms, Z=-1.244, p>0.05), middle (m_{alpha}=2681.50ms, SD=1004.06ms, m_{prob}=1855.38ms, SD=670.20ms, Z=-1.820, p>0.05) or bottom part of the screen (m_{alpha}=2173.29ms, SD=1330.47ms, m_{prob}=1782.86ms, SD=512.27ms, Z=-0.676, p>0.05). Although we note differences in mean times for the three screen positions and for each interface, Kruskal-Wallis tests show no statistically significant differences for neither the alphabetic ($\chi^2_{(2)}$=4.186, p>0.05) or probability ranked interface ($\chi^2_{(2)}$=0.594, p>0.05), showing that participant performance is similar regardless of interface or position in the list. These results show that the performance advantage of the alphabetic interface in our previous experiment is negated by the participants' knowledge of their own communication patterns, which enables them to estimate the position of a contact in the probability ranked lists just as they would in an alphabetic list. Finally, 46 participants indicated a subjective preference for the alphabetic interface, with the majority justifying their preference by indicating that they would be confounded by an probability-ordered listing, as it might be unclear to them how the ranking works. Given the results of both these experiments and those presented in [31] and [10], we decided to sort the top-k list of the proposed hybrid interfaces in an alphabetical order, basing our decision mostly on the participants' subjective preference.

5 Evaluation of Calchas

5.1 Methodology

In order to assess and evaluate the Calchas application we conducted a field trial. We uploaded our application to Google Play and then we made a public announcement to our university's community in order to gather participants. In total, we gathered 103 respondents.

Each participant was informed via a detailed email about the necessary steps in order to participate. Also, we clearly stated that our application collects anonymous usage data. If a participant agreed with those terms, then she/he had to download the Calchas application from Google Play. Moreover, we prepared a 3 minute video (uploaded to YouTube) with detailed instructions about the application. Finally, we showed participants how to create a shortcut for the Calchas application to their Android home screen, in order to ensure that its accessibility would be as good as the device's default calling applications (dialer, contacts).

Each participant was given a personal 4 digit PIN number. This PIN was necessary to unlock the application at the first run, after the installation. We chose to use a PIN number because it was necessary to know when each participant started using our application and when each participant provided to us the anonymous usage report. Furthermore, we divided the participants into two groups. The first group began the field trial with interface A of the application, while the second group started with interface B. When 15 days of usage were completed the application automatically changed the current interface to the other one. As such, the participants had the opportunity to evaluate both interfaces for the same period, resulting in a counter-balanced experiment. After completing 30 days of use, a message appeared at every launch of the application, informing the user that the field trial is over and that they should submit the anonymous report (Internet connection was required). Each anonymous report was effectively an SQLite dump of the database that the logging module of the application creates. At the end of the field trial, participants were also invited to submit an online self-assessment questionnaire, in order for us to obtain qualitative data about their experience.

Out of the 103 initial respondents, we received data from 73. For the quantitative and qualitative analyses that follows, from these participants we excluded those that had not completed 30 days of use, those that used mainly one of the two available interfaces (>75% usage of certain interface) and those that launched the application very few times (e.g. 3 times in 30 days). Finally, we excluded participants who did not complete the online questionnaire. As a result, we were left with valid data and questionnaire sets from 44 participants. From these participants, 34.1% (15) were female. In terms of their age, 63.6% (28) were between 18-24, 31.8% (14) were between 25-34 and 4.5% (2) were between 35-44 years old.

5.2 Experimental Results

In this section we present the experimental results of the conducted field trial. We divide our analysis into two parts: the first part shows the quantitative results (e.g. usage time, number of sessions etc.) as measured from the logging module of Calchas application, while the second part presents the answers of the participants to the self-assessment questionnaires.

5.2.1 Quantitative Results

In Table 1 we present a summary of the measurements for the variables that we were tracking in this field experiment. We note that on average 59.05% of the outgoing calls of our subjects during the experiment were made from the Calchas application. As it was expected, it seems that there are situations where it is easier to adopt other means for starting an outgoing call, e.g. using a notification after a missed call or when receiving an SMS. For all other cases that Calchas was not used we cannot be certain if it was because the participants preferred to use another application (e.g. list of favorites, recent call list, contact list etc.), because they were not accustomed to use alternative means in order to retrieve an item from the repository of contacts or because they just failed to find the contact they were looking for in our application. To obtain a better estimate of the actual usage of our application, we took into account calls made within one minute from the moment that a Calchas session (not resulting in a call) ended, considering these calls as actual failure of the user to be served by our application (e.g. the user opened Calchas, failed to find the desired contact and closed the app, then opened another app such as the dialer in order to make the call). When excluding calls for which the Calchas app had not been previously opened within this short timeframe, the percentage of calls having been made through Calchas raises to 85% (SD=7.97%). This fact shows that when Calchas was purposefully used, the participants were usually able to retrieve the desired contact. This also explains the mean numbers of clicks to dialer and contact list icons, which are considerably low (m=8.73, SD=9.85 and m=8.32, SD=10.61 respectively). The average duration of sessions that resulted in calls corresponds to the contact retrieval time, which is on average 5.61 seconds (SD=1.51). Moreover, we observe that the mean number of calls from the embedded list (either the recent call list or the list of contacts) that resides below the suggestion list is considerably lower (ratio 1:14) than the number of calls from the suggestion list. This remark suggests that the success rate of the prediction algorithm is such that usually allows users to retrieve the desired contact from the first screen of the application, without the need to scroll or search.

Table 1. Summary of experiment's measurements

	Minimum	Maximum	Mean	Std. Deviation
Number of sessions	58	1248	**401.66**	245.21
Average session duration (seconds)	4.09	32.98	**8.70**	5.71
Average duration (seconds) of sessions resulting in a call	3.15	9.19	**5.61**	1.51
Total outgoing calls	65	651	**224.73**	103.12
Outgoing calls from Calchas	20	408	**132.23**	74.27
Percentage of calls from Calchas	12.43%	92.41%	**59.05%**	19.73%
Estimated percentage of calls from Calchas	63.28%	100%	**84.93%**	7.97%
Calls from list of suggestions (in Calchas)	17	363	**115.57**	68.88
Calls from list of recent calls (in Calchas)	0	31	**8.64**	6.94
Calls from list of contacts (in Calchas)	0	25	**8.07**	5.84
Clicks to dialer icon	0	43	**8.73**	9.85
Clicks to contact list icon	0	57	**8.32**	10.61

We note from Table 1 that the recent calls fallback interface resulted in an additional 7.5% calls from within the Calchas application, while the contacts fallback interface in an additional 7%. These results show that a modest, yet measurable advantage is offered by the fallback interfaces, saving users time by allowing them to find the desired contact without leaving the app and initiating search from another interface. Usage of the two hybrid interfaces was almost equal, with users making 4.8% more calls on average (Table 2) during the contact list interface condition. A Wilcoxon signed rank test shows that this difference is not statistically significant ($Z=-1.173$, $p=0.241$). Comparing the average duration of sessions that resulted in a call (Table 2) for the two proposed hybrid interfaces, we can see that there is a difference of 0.57 seconds in favor of the interface with the embedded recent call list, which is found to be statistically significant (Wilcoxon signed rank test, $Z=-2.637$, $p<0.01$). It is noteworthy that for both interfaces the mean is rather closer to the minimum observation.

Table 2. Comparison of the average time required to perform a call between the hybrid interfaces

	Fallback UI	Minimum	Maximum	Mean	Std. Deviation
Average duration (sec) of sessions resulting in a call	Recent Calls	3.00	10.33	**5.31**	1.64
	Contacts	3.08	11.31	**5.88**	1.73
Average number of sessions resulting in a call	Recent Calls	5	239	**64.55**	43.317
	Contacts	11	169	**67.68**	33.791

In our previous work [8] we have defined a "socialness" metric that characterizes the communication pattern of a user. We showed that based on this metric we are able to categorize mobile users in three groups: Least Social, Medium Social and Most Social. The more "social" a user is the more difficult it is to predict the contacts she/he communicates with, as he/she interacts frequently with many different contacts. Computing the value of the "socialness" metric for our participants for the calls that were recorded during the field experiment, we found that there was 1 participant in the Least Social group, 12 in the Medium Social group, while 31 participants fall into the Most Social group. Use of this metric to further break down the analysis of our results, provides more meaningful interpretations of the data and allows a better understanding of participant behaviour and how the UIs affected their experience.

Table 3 shows the quantitative results of our experiment grouped by "socialness" (the Least Social group contained only one participant and was excluded). For most variables we did not find statistically significant differences. However, regarding the percentage of calls from Calchas, a Mann-Whitney U Test revealed a statistically significant difference ($U=109$, $p<0.05$), showing that the 'Medium Social' users made approximately 7% more calls from Calchas. This result was expected since 'Most Social' users' behavior has been shown to be more difficult to predict [8]. A further observation is that the fallback interfaces provide an increased advantage to the most-social group of users (8.9% with recent calls and 8% with contacts) compared to the medium-social users (4.7% in both cases).

Table 3. Analysis of variables per "socialness" group (2=Medium social, 3= Most social)

Variable	***n***		**Mean**		**Std. Deviation**	
	2	**3**	**2**	**3**	**2**	**3**
Average duration of sessions resulting in a call	12	31	**4.95**	**5.90**	1.14	1.57
Total outgoing calls	12	31	**231.08**	**227.06**	154.31	75.84
Outgoing calls from Calchas	12	31	**152.00**	**128.19**	111.09	52.80
Percentage of calls from Calchas	12	31	**66.64%**	**57.17%**	24.94%	16.35%
Estimated percentage of calls from Calchas	12	31	**86.77%**	**83.88%**	8.6%	7.61%
Calls from list of suggestions (in Calchas)	12	31	**138.92**	**109.68**	100.75	50.08
Calls from list of recent calls (in Calchas)	12	31	**6.58**	**9.71**	8.93	5.86
Calls from list of contacts (in Calchas)	12	31	**6.58**	**8.84**	5.77	5.83

The results of the field experiment confirm our previous findings [8], where we ran the prediction algorithm under ideal conditions (i.e. for every outgoing call) over the extracted call logs from the NOKIA mobile data challenge dataset [28]. In Figure 2 the success rates of the prediction algorithm are illustrated. In order to compute the prediction performance of the algorithm in the field experiment, we consider as successful only calls from the list of suggestions and not from the lists of contacts and recent calls that took place inside Calchas, in order to be able to directly compare with our previous findings. We observe that the estimated success rate of our algorithm is only slightly lower than the one computed when applying it on the NOKIA dataset, where we considered a suggestion list of 8 contacts instead of 7. In addition to this, we confirm that the prediction rate is higher for users in the Medium Social group than users of the Most Social group, whose communication interactions we showed are harder to predict.

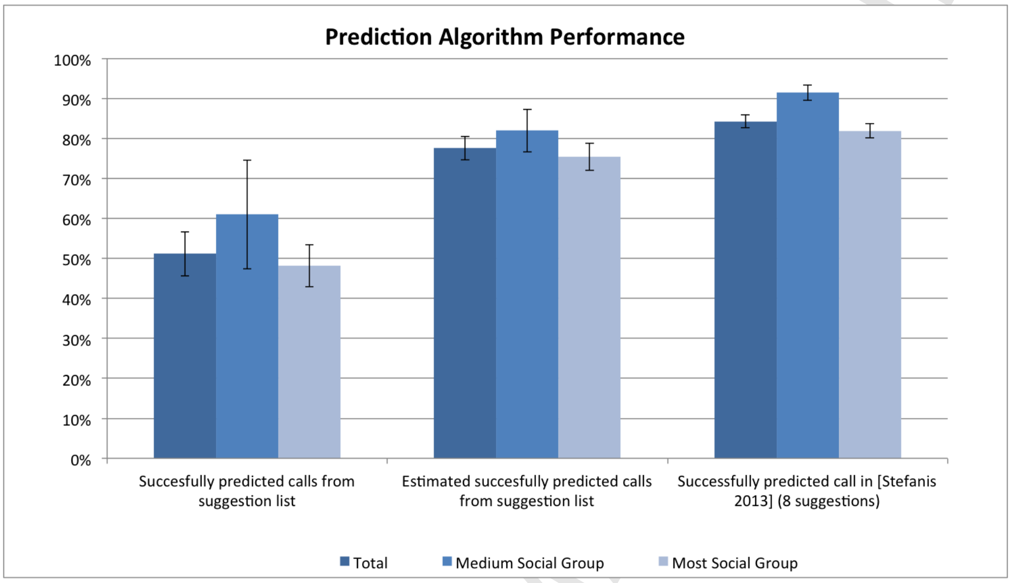

Figure 2. Success performance of the prediction algorithm, compared to [Stefanis 2013]

Moreover, we can compare our findings with those reported in [2], where a similar field experiment involving 10 participants for 3 months took place to evaluate their adaptive dial list. Lee et al. took into account only calls performed from their application and as a result their measure of success rate corresponds to our estimated prediction performance of our algorithm. The researchers report a number of successfully predicted calls that corresponds to a success rate of 72.62%, while we found a success rate of 77.57%. However, their suggestion list consists of 5 contacts. On the other hand, we do not have any evidence on the "socialness" of their users, which we have shown to be an important factor when predicting outgoing calls. In addition, the total number of calls that were recorded (427 calls) in their experiment is much lower than those performed from the suggestion list in our field trial (5085 calls).

Figure 3 sums up the percentages of outgoing calls that were presented earlier in this section, outlining the improvement that the hybrid interfaces have over the simple adaptive dial list on both the real and the estimated percentage of calls. Again, for the charts corresponding to the speed dial list we take into account only calls from the suggestion list, while for the charts of the hybrid interfaces we consider all calls from Calchas. We observe that a hybrid interface has a statistically significant positive effect in all cases of our analysis (p<0.01 for all pairwise comparisons) using Wilcoxon signed rank tests for the Medium Social group and paired-sample T-tests for the other groupings, resulting in an improvement ranging from ~5% to ~13%.

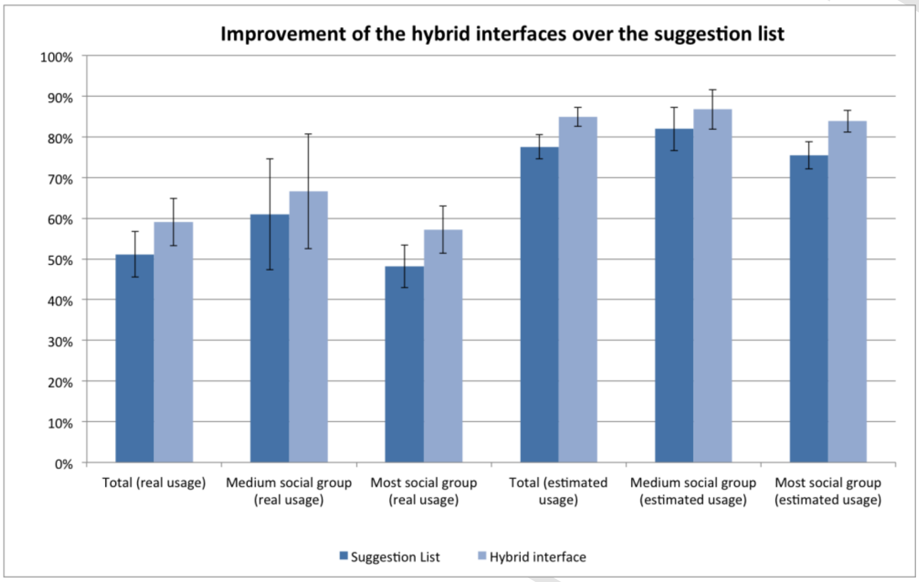

Figure 3. Improvement of the hybrid interfaces over the speed dial list for all cases

5.2.2 Qualitative Results

Participants were asked to complete an online survey at the end of the experiment, which aimed to collect their subjective feedback on using the two interfaces. The questions in this survey were selected and adapted from [32]. The survey was split in three parts: first, a comparative assessment of the two interfaces, followed by a general feedback section to compare previous experience with the hybrid interface concept and, finally, a section for providing feedback in free text form.

Table 4. Comparative evaluation questions

Question	Scale Type
How easy was it to learn to use the interface?	A
Do you think the interface allows you to quickly find the contact you are looking for?	B
Was it easy to find the contact you were looking for with this interface?	B
How easy was it to remember the interface functions?	A
Interaction with this interface required a lot of mental effort	B
I found this interface to be frustrating	B
I would miss this interface if I had to uninstall it from my phone	B
The icons, graphics and labels were relevant to the function that they referred to	B
The interface response time was satisfactory	B
The available volume of information in the interface was satisfactory	B

The number of actions (e.g. clicks, scrolling) that I needed to perform in order to make a call was small	B
Scale Type A:	1=Very hard 5=Very easy
Scale Type B:	1=Strongly disagree 5=Strongly agree

For the comparative evaluation, the questionnaires included 11 questions (Table 4) that were common for both hybrid interfaces, allowing for a direct qualitative comparison between them through responses on 5-point Likert scales. Figures 4 and 5 present the distribution of user feedback from these scales. As one can infer from this graphical representation, the participants were highly satisfied regarding all aspects from both interfaces. A statistical analysis of this data using Wilcoxon signed rank tests, confirms that there is no significant difference in the responses between the two hybrid interfaces for all of the questions.

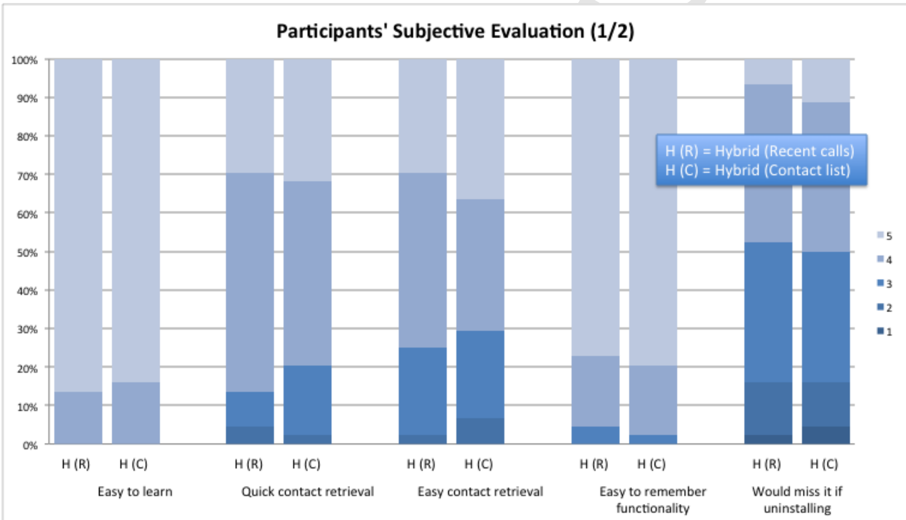

Figure 4. Subjective evaluation of the two hybrid interfaces (1/2)

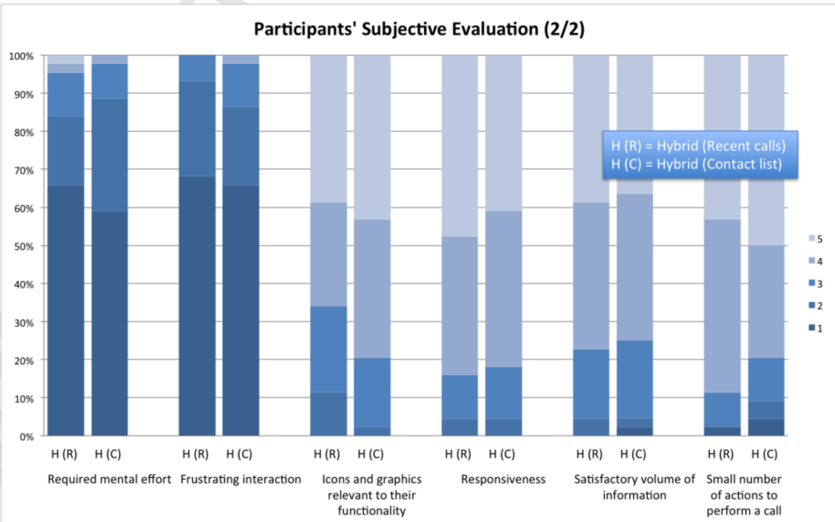

Figure 5. Subjective evaluation of the two hybrid interfaces (2/2)

Table 5. Usage frequency of traditional interfaces for making outgoing calls

	Mean	Std. Deviation
Recent Call List	4.57	.873
Notifications	3.95	.963
Contact List	3.43	1.021
List of Favorite Contacts	2.18	1.483
SMS	1.66	.888
Contact Groups	1.36	.810

In the second section, we attempted to obtain a comparison of participants' previous experience to the use of the hybrid interfaces. Table 5 shows, in descending order, how frequently (1=rarely, 5=very often) the participants declared that were using the traditional means provided by smartphones in order to start an outgoing call before the experiment. The recent call list (second more popular choice in [2]) is the place where users use to lookup more often in order to retrieve a contact, while notification popups seem to facilitate interactions with contacts and the contact list follows in popularity.

Table 6 presents the responses of the participants to the question "how helpful did you find Calchas compared to these alternatives for making an outgoing call" (1=much less, 5=much more). We observe that the participants found Calchas on average slightly more helpful compared to all traditional means of contact retrieval. A remark that we would like to emphasize here is that Calchas is rated on average more helpful than the recent call list and contact list, while it is rated almost equal to notifications in terms of helpfulness. This underlines the utility of each retrieval tool under the context in which it is being used and that participants are probably better served by multiple tools, instead of just one.

Table 6. Comparison of preference between Calchas and traditional interfaces for outgoing calls

	Mean	Std. Deviation
Calchas – Recent Call List	3.73	.973
Calchas – Notifications	3.05	1.033
Calchas – Contact List	3.64	1.014
Calchas – List of Favorite Contacts	3.50	1.355
Calchas – SMS	3.32	1.343
Calchas – Contact Groups	3.25	1.366

The final question in the second section was "*Which of the two hybrid interfaces would you choose as your preferred one?*". While the qualitative evaluation of the two hybrid interfaces in section 1 reveals that participants did not feel there was an actual difference between them, in this question the clear majority (28) reported that they would prefer an application with the recent call list embedded below the suggestion list. One participant claimed no preference over the two interfaces and the remaining 15 would prefer an embedded contact list.

The final survey section allowed participants to freely enter any comments or feedback on their experience. Analyzing the participants' comments about their experience with Calchas we confirm that in general they found it useful ("*very helpful application, fast and easy to use*", "*I liked it very much*" etc.). Several participants suggested indirectly the adoption of other context dimensions for the prediction algorithm ("*It would be useful if there was a connection with my calendar and the suggestions included contacts on their birthday*") and expressed their desire for customization ("*I would like to be able to change the suggestion list size*", "*I would prefer to manually set the weights of the prediction criteria*"). Most of our subjects found the prediction algorithm sufficiently accurate ("*In most cases I was very satisfied with the suggested contacts*", "*The application predicted correctly most of the calls I wanted to make*"). The most negative comments target the UI of the application ("*I didn't like the UI*", "*My experience with the UI was quite bad*", "*A better UI is needed*"). This is quite a surprise, since both interfaces are combinations of a speed dial list with existing mobile interfaces that are already used to retrieve contacts. Although these comments may relate to the aesthetics of the interfaces provided, we see this finding as an indication that users are not satisfied with traditional UIs regarding contact retrieval, since the layout and concepts were not radically different from existing UI designs.

6 Conclusions

Although predictive algorithms for contact lists have been proposed and evaluated through simulation in literature, with the exception of [2] where only 10 participants were recruited using their application sparsely (0.5 calls per day and user, while usage in our field trial reached 3.85 calls per day for each user), they have never been evaluated in the field. Moreover, all related papers describe top-k lists as UIs and none of them (with the exception of [7], where details of the evaluation results were omitted) examine hybrid interfaces as a fallback mechanism for the algorithm failures. The objectives of our work were to address the shortcomings of previous literature, by examining the performance of our prediction algorithm under real world conditions and evaluating the two hybrid interfaces that provide alternative means of access to contacts in case of prediction failure. Our field experiment provides valuable insight on both the performance of the prediction algorithm under real world conditions and the efficacy of hybrid interfaces as fallback mechanisms. Hence, we feel our work has made several valuable contributions.

Our first contribution concerns the assessment of the use of the algorithm in the real world. The participants performed on average almost 60% of their total outgoing calls through our application. If we take into additional account those calls that were made within a short timeframe after a Calchas session has ended, we can single-out the cases in which users were unable to retrieve the desired contact either from the suggestion list or the embedded list (in all other cases our application was not used at all). Excluding all other calls from our analysis, we get an estimation of the actual usage performance of Calchas. These estimated percentages are significantly higher than when considering just the Calchas sessions resulting in a call (on average around 78% for the suggestion list alone and 85% for the hybrid interface) and very close to those that we found when the prediction algorithm was applied to extracted call logs [8], confirming both the accuracy and efficacy of the algorithm in the field, as well as the validity of the simulation in our previous work. The experimental results also confirmed our findings in [8] regarding how users' "socialness" in terms of communication behavior affects the success of predictions. We show that users with more complex communication patterns are more difficult to predict, a hypothesis that was also proposed in [2].

Our second contribution follows the performance findings for our algorithm, and concerns the use of hybrid interfaces as fallback mechanisms. Our analysis shows that the vast majority of calls from Calchas were made from the list of suggestions and only a small number of contacts were retrieved from the embedded call log and contact list elements of interface. Additionally, only a small number of clicks to the icons that open the traditional contact list or recent call list were recorded. As the high estimated success rate of the prediction algorithm suggests, in most cases the desired contact is found in the first screenful of our application's content (list of suggestions) without the need to scroll or search elsewhere. We have also seen in Figure 3 that embedding a list of recent calls or a list of contacts in a hybrid interface provides a slight but measurable improvement over the simple adaptive dial list, particularly for highly social users (9% more calls from within the Calchas app than using the suggestions list alone). Our observations seem to contrast Bergman's theory about demotion in mobile devices [3], at least when the prediction algorithm performs well. Although hybrid interfaces can provide some advantages, particularly for hard-to-predict users, our findings recommend that enhancing the prediction algorithm for contact retrieval is where further research efforts should be concentrated, especially for users that belong to the Most Social group, where there is still significant room for improvement. In our future work we intend to work more on the prediction algorithm, incorporating more context dimensions, such as location, which we believe plays a significant role regarding mobile communication patterns.

The qualitative evaluation of the application shows that the participants did not find any significant difference while using the two proposed interfaces, something that was also verified from the quantitative analysis. In general, participants found the application useful for contact retrieval and in average they would prefer it over all other available interfaces that a mobile device offers. Nevertheless, this preference for a hybrid interface is slight rather than strong. This, in combination with the only modest increase in performance that the hybrid interfaces offered compared to a suggestion list, provides further evidence that such interfaces may not be an optimal design to support algorithmic failure during personal information item search and retrieval in mobile devices. We should note that most negative comments regarding the application are about the UI, which we see as reflective of the general public concern on contact list retrieval interfaces. Consequently, it appears that there is a need for new redesigned mobile UIs for personal information management and in the future we intend to work towards this direction.

7 Further Work

Our findings provide several other research directions for future work. We found that there are some usage sessions of the application near which there are no outgoing calls, which was also observed during the field trial of the prototype presented in [7]. It seems that from time to time the participants used to open the application just to check the suggestions and get an idea of the contacts that are more probable to call. Moreover, for the 40% of calls that were made without using Calchas, we cannot be sure which other facility of the users' Android device was used to initiate them (e.g. notifications, SMS etc.), however it seems that under different contexts, mobile users adopt different means for retrieving a contact in order to start an outgoing communication session. For example, when a user finds a notification for a missed call, it is possible that he/she will call back that contact by directly clicking on the notification. Another such case is when an outgoing call is made after the reception of a text message or when the user remembers having talked with a contact recently and searches for it in the recent call list. Currently, as far as we know, there is no literature investigating user behaviour and their use of communication modes and tools under different contexts – further research should look into this subject, which may yield contextual cues that could improve prediction results.

Our findings also suggest that if the predictive algorithm is highly successful, the need for a fallback method is limited. We are hence currently working on a novel UI that focuses on the prediction and presentation of suggestions only, and will allow mobile users to project future and past predictions to the present, adapting the prediction algorithm to future or past contexts (Figure 6 left). At this point, our application has been re-designed for better aesthetics and, along with several bug fixes and the removal of the logging module, is available as a free-to-download Android app[1] (Figure 6 right).

Figure 6. A mockup of an interface adapting the prediction algorithm to future contexts (left) and the publically available version of Calchas (right)

In two months since its release, it attracted approximately 1500 downloads, with approximately 300 users adopting the application for permanent use, while after one year and a half in which only minor bug fixing updates were released, there have been 1900 downloads and 160 active installations. In the future, we aim to take advantage of this user base in order to try out new UI designs and algorithm improvements in the real world, removing the limitations of trials under the context of a guided experiment.

8 References

1. LaRue EM, Mitchell AM, Terhorst L, Karimi HA (2010) Assessing mobile phone communication utility preferences in a social support network. Telematics and Informatics 27(4):363-369. doi: 10.1016/j.tele.2010.03.002
2. Lee S, Seo J, Lee G (2010) An adaptive speed-call list algorithm and its evaluation with ESM. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10), pp. 2019-2022. ACM
3. Bergman O, Komninos A, Liarokapis D, Clarke J (2012) You never call: demoting unused contacts on mobile phones using DMTR. Personal and Ubiquitous Computing 16(6):757–766. doi: 10.1007/s00779-011-0411-3

1 Calchas Application: https://play.google.com/store/apps/details?id=com.upatras.ceid.calchasapp

4. Bentley F, Chen YY (2015). The Composition and Use of Modern Mobile Phonebooks. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI ‘15), pp. 2749-2758. ACM.
5. Salehan M, Negahban A (2013) Social networking on smartphones: When mobile phones become addictive. Computers in Human Behavior 29(6):2632–2639. doi: 10.1016/j.chb.2013.07.003
6. Komninos A, Liarokapis D (2009) The use of mobile contact list applications and a context-oriented framework to support their design. In: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI’09), Article No. 79. ACM
7. Jung Y, Anttilla A, Blom J (2008) Designing for the evolution of mobile contacts application. In: Proceedings of the 10th international conference on Human computer interaction with mobile devices and services (MobileHCI’08), pp. 449-452. ACM
8. Stefanis V, Plessas A, Komninos A, Garofalakis J (2014) Frequency and recency context for the management and retrieval of personal information on mobile devices. Pervasive and Mobile Computing, 15: 100-112. doi: 10.1016/j.pmcj.2013.08.002
9. Plessas A, Stefanis V, Komninos A, Garofalakis J (2013) Using communication frequency and recency context to facilitate mobile contact list retrieval. International Journal of Handheld Computing Research (IJHCR), 4(4): 52-71. doi: 10.4018/ijhcr.2013100104
10. Stefanis V, Komninos A, Plessas A, Garofalakis J (2013) An interface for context-aware retrieval of mobile contacts. In: Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services (MobileHCI '13), pp. 492-497. ACM
11. Hang A, von Zezschwitz E, De Luca A, Hussmann H (2012) Too much information!: user attitudes towards smartphone sharing. In: Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design (NordiCHI’12), pp. 284-287. ACM
12. Zhou L, Mohammed A, Zhang D (2012) Mobile personal information management agent: supporting natural language interface and application integration. Information Processing and Management 48(1): 23–31. doi: 10.1016/j.ipm.2011.08.008
13. Boardman R, Sasse MA (2004) Stuff goes into the computer and doesn’t come out: a cross-tool study of personal information management. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI’04), pp. 583-590. ACM

14. Myers R, Zapata E, Singh G (2007) Linking information for mobile use. In: Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology (Mobility'07), pp. 607–613. ACM

15. Bergman O, Beyth-Marom R, Nachmias R (2008) The user-subjective approach to personal information management systems design: evidence and implementations. Journal of the American Society for Information Science and Technology 59(2): pp. 235–246. doi: 10.1002/asi.20738

16. Church K, Smith B (2009) Understanding the intent behind mobile information needs. In: Proceedings of the 14th international conference on Intelligent user interfaces (IUI'09), pp. 247–256. ACM

17. Oulasvirta A, Raento M, Tiita S (2005) ContextContacts: Re-Designing Smartphone's Contact Book to Support Mobile Awareness and Collaboration. In: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services (MobileHCI'05), pp. 167-174. ACM

18. Bentley FR, Metcalf CJ (2007) Sharing motion information with close family and friends. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07), pp. 1361-1370. ACM

19. Barkhuus L, Brown B, Bell M, Sherwood S, Hall M, Chalmers M (2008). From awareness to repartee: sharing location within social groups. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'08), pp. 497–506. ACM

20. Ankolekar A, Szabo G, Luon Y, Huberman B, Wilkinson D, Wu F (2009) Friendlee: a mobile application for your social life. In: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'09), Article no 27. ACM

21. Bentley F, Kames J, Ahmed R, Sterling Zivin R, Schwendimann L (2010) Contacts 3.0: Bringing together research and design teams to reinvent the phonebook. In: CHI '10 Extended Abstracts on Human Factors in Computing Systems , pp. 4677-4690. ACM

22. Cui Y, Honkala M (2013) A novel mobile device user interface with integrated social networking services. International Journal of Human-Computer Studies 71(9): 919-932. doi: 10.1016/j.ijhcs.2013.03.004

23. Huot S, Lecolinet E (2007) Focus+context visualization techniques for displaying large lists with multiple points of interest on small tactile screens. In: Proceedings of the 11th IFIP TC 13 international conference on Human-computer interaction, pp. 219-233. Springer-Verlag
24. Gaur S (2008) Mobile Phone Contact Book: The Social Approach. Services and social networking on the Internet, Antti Ylä-Jääski, in Laura Takkinen (eds.) TKK Technical Reports in Computer Science and Engineering.
25. Barzaiq O, Loke S (2011) Adapting the mobile phone for task efficiency: the case of predicting outgoing calls using frequency and regularity of historical calls. Personal and Ubiquitous Computing 15(8):857–870. doi: 10.1007/s00779-011-0401-5
26. Phithakkitnukoon S, Dantu R (2011) Towards ubiquitous computing with call prediction. ACM SIGMOBILE Mobile Computing and Communications Review 15(1):52–64. doi: 10.1145/1978622.1978628
27. Phithakkitnukoon S, Dantu R, Claxton R, Eagle N (2011) Behavior-based adaptive call predictor. ACM Transactions on Autonomous and Adaptive Systems (TAAS) 6(3): Article No. 21. doi: 10.1145/2019583.2019588
28. Laurila JK, Gatica-Perez D, Aad I, Jan Blom TM TD, Bornet O, Dousse O, Eberle J, Miettinen M (2012) The mobile data challenge: big data for mobile computing research. In: Proceedings of Mobile Data Challenge Workshop (MDC), in conjunction with Pervasive.
29. Sun C, Wang Y, Zheng J, Hsu DF (2013) Feature fusion for mobile usage prediction using rank-score characteristics. In: Proceedings of 12th IEEE International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC), pp. 212-217.
30. Stefanis V, Plessas A, Komninos A, Garofalakis J (2012) Patterns of usage and context in interaction with communication support applications in mobile devices. In: Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services (MobileHCI'12), pp. 25–34. ACM
31. Seipp K, Devlin K (2013) Landscape Vs Portrait Mode: Which Is Faster To Use On Your Smart Phone? In: Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services (MobileHCI'13), pp. 534-539. ACM
32. Ryu YS, Smith-Jackson TL (2006) Reliability and validity of Mobile Phone Usability Questionnaire (MPUQ). Journal of Usability Studies 2(1): 39–53.