An Approach to Context Aware Event Reminding

Dušan Zeleník*
Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
zelenik@fiit.stuba.sk

Abstract

In this paper we present a method for event reminding. Our method is designed for scheduling leaving time from one location to catch the event at different location. We use a mobile device to track user activities and discover patterns in her activities by analyzing locations where she spends time, time when she is travelling, type of transportation she uses and weather conditions. These observed contexts are used to remind the user when to leave, when to wake up, etc. Our idea is to help organise activities of the user and adapt to the behavioural patterns which we discover. We present in this paper the solution which discovers patterns and models the user using records made by mobile device in energy efficient manner.

Categories and Subject Descriptors

I.5.1 [**Models**]: Statistical, Structural; H.3.4 [**Systems and Software**]: User profiles and alert services

Keywords

user behavior, pattern discovery, context, monitoring, location, time, planning, reminder, mobile device

1. Introduction

Nowadays almost everyone uses mobile device to communicate with others and organize their lives. Mobile devices are always with us and even cheaper alternatives are capable of monitoring us. We could be always located by our provider using telestrial GSM antennas and signal strength or GPS modules. This triangulation could be also used locally by every peer with simple, but programmable mobile phone. Moreover, we could use these locations for our own benefit. Monitoring contexts [8] get us through advanced analysis to adaptation what eventually leads to comfort. More advanced mobile device we use, more adaptation we can provide. We present here a method for reminding upcoming events such as meetings, lectures or work start, school start. Our idea is to help a user to decide when to leave and never miss any event. We adapt currently simple alarm clock to the user needs. We remind meeting earlier if it is snowing outside but do not bother the user when she is already on her way. To illustrate situation, imagine scenario where Petra sleeps in the morning and her alarm clock is set to 7:00 am 1 hour before her lecture starts. But it started snowing during night and there is a traffic collapse on her usual route. With today alarm clock she is woken up but she is late because of the traffic jam though. To avoid such an unpleasant happening we propose the method to adapt alarm clock and we wake up Petra earlier according to current situation. In this case Petra's alarm clock is set to 6:40 am and she is at her destination on time.

This is about past, current and future contexts [12] which have to be considered. Petra does not know what would happen next morning. Even system does not know this. It relies on the current situation and known contexts. Alarm clock has to be adapted as soon as it is aware of future context. In mentioned or similar scenarios we need to know details about current situation and we should have knowledge about how to solve such situation. Details about situation are monitored as contexts. We focus on

- current location (current context),
- destination (where is the user heading - past/future context),
- time (current context),
- time to transport (past/future context),
- type of transportation (past/future context) and also
- weather (future context).

These are contexts which we monitor in long term to successfully adapt our reminder when it is needed. The location is monitored permanently and stored with timestamp. We analyze these logs to discover patterns of user behavior and how she solved past situations, how long did

*Doctoral degree study programme in field Software Engineering. Supervisor: Professor Mária Bieliková, Institute of Informatics and Software Engineering, Faculty of Informatics and Information Technologies, STU in Bratislava. Work described in this paper was presented at the 7th Student Research Conference in Informatics and Information Technologies IIT.SRC 2011.

Zeleník, D. An Approach to Context Aware Event Reminding. Information Sciences and Technologies Bulletin of the ACM Slovakia, Special Section on Student Research in Informatics and Information Technologies, Vol. 3, No. 2 (2011) 126-130

it take from one place to another relatively to conditions like weather and type of transportation. The location and the time is the only input which we need to analyze and predict user behavior. Other information is either downloaded (weather from the web using time) or calculated (type of transportation using movement patterns).

Our method does not need exact locations. We do not need GPS attributes to adapt. We neither need internet connection but in this case we use only analyses based on user behavior. Weather as a condition or analyses based on other users are not considered in this case. For online users we also use power of crowd to discover patterns especially when our user is new or current location is unknown for her. Side product of such collaborative information gathering is a map of transport times.

2. Related work

The term context in is often misinterpret. We understand context as synonym of situation or condition [12]. The context of the user could be simple as her actual time or location but also more complex such as accompanies or even her emotions.

Ciglan et al. in [6] introduced work where event was understood as happening relevant for masses. Event in this work is meant to be for example New Year, Valentine's Day or Christmas. They used Wikipeadia page views statistics to find out what is happening in the selected period of time. They analyzed topics and connections to these events. Similarly to our work they analyzed behavior of users. In our work we do almost the same but with individuals separately. Our work focuses on the simplest events which are not relevant for masses but for individuals. Furthermore, we need to find behavioral patterns using periodicity in records to discover repeating events. For example the ritual of working from 8:00 am to 16:00 on weekdays is considered to be event relevant for our user.

We do analysis of the individual to discover events which are connected to the need of transportation. Similarly to work on task recommendation [13] we are monitoring user using mobile devices. Task in mentioned work is defined to be upcoming activity for analyzed user. They are trying to interact with user to e.g. change phone ring to loud mode if user is outside and make it silent when he is watching movie. We focus only on one type of task - transportation from one location to another. We are specialized on transportation which we have to predict if it is needed.

Prediction is based on human rituals as it was mentioned by Bamis [4]. They suggest to use many different sensor data like room temperature to determine e.g. whether user is sleeping or not. We simplified user rituals to locations she visits. Even with this simple attitude we are able to determine locations where user stays. Further analyzes of the locations then reveals location where user lives, where she works or where she eats. We decided to simplify this monitoring to locations because they are satisfying for our method and mostly because of the practical reason. We want to track user everywhere so we can work with sensor data only using her mobile devices. Naturally, we want to cover as many types of devices as it is possible. Sensor data introduced by Bamis [4] are very useful but not easily conducted by common user.

Other option is to ask user to interact with the system. User is able to provide information about current contexts. The music recommender system presented in work by Baltrunas [3] uses complex ratings where the user is able to choose contexts to say when the music is appropriate for others. In work by Hsu [10] they even use an enhanced remote controller for your TV to set your current context (mood). Alternatively in work made by Cebrian [5] they present possibilities to use statistics to replace such interaction. Recommendations are then made by the mass of users using simple context of time.

We are constrained by internet connection which we assumed is not usual for our users. We have to analyze users as individuals and offline. We understand the intention of Mei [11] where power of crowd helps individually. In this work authors present how to suggest queries to individuals using queries popular by crowd. They use short term context to suggest in information retrieval process. They assumed users are interested in the same agenda in the same time. They connected these contexts to improve searching the web. However in our solution we have to work with contexts offline. We also designed solution for sharing contexts but we have to ensure that majority of the users without internet connection would be satisfied.

Since we work with locations we considered side product of our solution. We are able to create a map with times needed to transport from one point to another. Clarke [7] mentioned a game where mobile devices where used to track players who were searching for secrets on the map. They playground was set in the city. Players were moving from one point to another with an intention to beat others. The game emerged to a map with shortest paths was created. In our case we work mainly with locations and time. Every user using our method creates a map of her transportation times. If we were able to gather this information we would be able to reveal a map with times needed to transport from one point to another. Besides we can use other external conditions such as weather context to calculate time needed (related to these conditions). Contexts and adapting

Our method for event reminding is based on behavior patterns and discovered contexts. We can discover and use several contexts such as location, time or weather. Our method works with mobile devices which is always on and monitoring, i.e. ready for discovering several contexts. Firstly, we show how to get contexts and how our method monitors users. Secondly we describe how to discover behavior patterns in relation to these contexts. Some of these contexts are not interesting for pattern discovery but at least location and time as contexts coexist in behavior pattern which we are mainly interested in.

2.1 Location

Nowadays most of mobile devices provide functionality to locate themselves using GPS module. GPS module is in fact very precise and location obtained is easily translated to exact place. This technology has many disadvantages though. GPS excessively consumes battery and initialization often lasts too much. There is also problem inside buildings and covered places. Actually not every device is equipped with GPS and users rarely let it activated permanently. For this reason we decided to use antennas instead of GPS. The idea is in monitoring which telestrial GSM antennas are connected. We are able to locate user using her mobile phone and these antennas. This technology is not so precise and location needs to be translated into GPS coordinates to match exact place but these little drawbacks are good price for low battery consumption and better availability for our users.

Technology provides simple identification of the antenna which has the best connection quality and lets us know the distinct position of the user. Actually we do not need to translate this data to the exact GPS coordinates. Therefore we do not need to distribute and then actualize database with these translations at every user instance. Identification itself is enough to differentiate locations and calculate distance in time. Distance in time is exactly what we need if we want to adaptively remind events.

Every event which should be reminded happens at one of the locations. Our reminder learns day by day. Every event that passed is recorded. We know where user was and where he spent some time from these records. By monitoring user and learning his rituals we discover events. The event is a pair of location and time. After longer observing we are able to schedule events automatically. When event is going to be reminded its location becomes a destination. Current location and destination of following event are then used as attributes to calculate time distance.

In our work we discover event locations by monitoring and then predicting when these destinations are relevant. These locations are those where user spends some amount of time. These are not places where she just gets through nor locations which are random. For example home location is not considered to be event location. User spends lot of time here but she has no obligation to be at home so she does not arrive periodically. This pattern is recognized as home location and it is filtered out (it is not recognized as event). Figure 1 presents locations passed by one user. These locations were recorded in 5 minutes intervals. 5 minutes interval caused that one longer trip on the left side has only few intersections between ways there and back. Actually we could shorten this interval but there is no need since we want to focus on locations which are visited more times.

2.2 Time

Every location, even those which user passed only once for a short time, is recorded. Locations are stored with a timestamp. Time is another important context which we analyze. We are not aggregating events by periods during a day like researchers do in other similar projects [5]. We work with these timestamps which are then used to discover mentioned locations and time needed to transport between locations. We are searching for periodicity and rituals. Our method for event reminding uses recursive analyses of the records. The algorithm for repeating event discovery follows this sequence:

1. Nearby locations are merged (e.g. user has more locations in the same building)
2. Locations below threshold are removed (threshold for period is set to 15 minutes)
3. Interval is set to length of recording
4. Locations and timestamps are grouped by intervals

Figure 1: User movement during the period of 2 months. Each point represents location. Size of the point represents how many times it was visited. Two clusters on bottom right represent home and work. The long path is one trip to the far location and the way back.

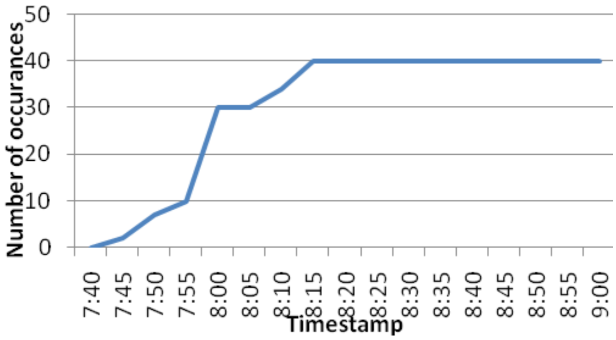

Figure 2: Histogram for short sequence (around 8:00 AM) and one location (Work).

5. Histograms for every location over timestamp are calculated
6. Histogram for every location is evaluated and score for intervals and locations is given
7. Set interval size to the one which lead us to better score or end
8. Jump back to step 4

Steps 1 through 4 are done for filtering unnecessary records and simplifying the analyses. We are filtering locations which were passed but those are not relevant for further knowledge discovery. The following cycle is then used to search for the best interval. Iterations approximate the correct result. The most important step is evaluation of histogram (step 6). Total number of locations and timestamps are put in the ratio with all discovered maximums (Fig. 2) and score is calculated. This partial evaluation approximates the best result. Result for every location is then used as discovered ritual. We can observe that user was always at work at 8:15 am. This is the time she has to be at work and it is considered to be start of the event.

When we know how to predict following events we also need information about how long does it take to reach the destination. This is done by retrieving current location and searching the shortest path to the predicted destination. As you can see on previous picture, our user has to be at work between 7:40 am and 8:15 am. Since we predicted work as following event we have to reserve time needed for transportation when we want to remind. This is also how our method adapts to the present location. To manage the problem of searching for shortest path between single points we use Dijkstra's algorithm.

Another fact we considered is time needed to prepare before user leaves. This is more complex problem which we currently solve only by comparing time she usually leaves and time she set on alarm clock. This is considered to be the maximal time needed to prepare before she leaves. It means that we use the same time for events which are happening in the morning and event which are happening later during day. Actually we should only remind in cases that our reminder discovered that user is very late on leaving. In this case we do not need to know the time she needs to prepare herself. Our reminder just warn that it is the time to leave (it will not bother the user if she is already on her way).

2.3 Conditions of transportation

Besides location and time there are even more contexts which should consider. We want to discuss type of transportation used and also weather. Both contexts are not easily retrieved. We have to analyze time needed to transport from one point to another to decide whether user went by walk, took a bus or drove a car. Actually this is necessary only if user changes her rituals and she alternates her type of transportation. In other cases, there is almost no chance to predict such thing and the only way is to ask user.

Other interesting condition is weather, which definitely affects the time needed to transport from one place to another. Snowing, raining or similar conditions slow down traffic or even cause jams what means that such information would help user to estimate time needed to transport. Users with internet connection have such an option. During estimating of time needed to transport are information about current location used to download current weather. This context is then used for better estimation.

Every context brings another dimension to our records. Not only type of transportation or weather conditions but also part of a day affects how long it takes to transport. Actually part of a day is considered to be one of the most useful contexts [9]. We use part of a day as other aspect that affects time needed to transport. To create suitable representation we work with records which are combinations of

$$(location) \times (weather) \times (transportation) \times (time). \quad (1)$$

It means that we use combinations of contexts which are associated with record. Time between locations is then remembered for more combinations. This leads to complex structure which could cause serious performance problem. Since we work with individual users only, our data is not so complex rather simple. Calculation itself is then simple so far. We also mentioned the map of transportation times. This map could be collaboratively generated by recording more users and maintaining the same database. Mentioned complexity problem would be an issue in this case.

3. Evaluation

In our work we want to experimentally evaluate two hypotheses. Both experiments we did were prepared with data we obtained by monitoring one user for 2 months. Our monitoring software was installed on Windows Mobile 6. Device was recording the location every 5 minutes. The location was recorded only if it was changed since last observation. We recorded information about GSM antenna and time. Our intention was to use this data to discover rituals and then observe hypotheses we placed.

3.1 Discovering rituals

First hypothesis is our presumption that periodical rituals are repeating in common time frames. It means that people are repeating their manners daily, weekly, monthly (etc.). Our method for discovering rituals is used to search for intervals of any length. We want to compare length of intervals discovered by our method and length of intervals which we presumed. In our experiment we discovered more unknown rituals. More of them were intensive enough to be discussed here. We noticed that discovering of rituals which are repeating in shorter intervals is more accurate. It is consequence of the number of repeats. In the table 1 we placed discovered rituals and their attributes.

As we can observe, weekdays are discovered as weekly rituals. It is because of free weekends and little differences between Mondays, Fridays and rest of the week. There are also few workdays when our user was not at work (business trip, holiday or illness). We also observed that leaving work earlier on Friday becomes also a ritual. In this experiment we observed that our method is able to discover rituals. Lengths of intervals are almost as we expect (mostly a week). We could claim that there is no need to search for intervals which are very rare basis. Methods which use heuristics on these common intervals would be as successful as method we proposed or even better.

3.2 Contexts affecting rituals

In this experiment we want to show that context we mentioned before affects rituals. Especially we want to observe time needed to transport from one location to another and how it is affected by weather and time. Our hypothesis is that e.g. rain affects traffic what should affect time needed to transport from one place to another. Part of day also affects traffic. In the morning there are lots of cars on the roads. We used our data to show that these contexts affects time needed to transport.

In the following table (Tab. 2) we show how long it takes to move from home location to work location. We work with the same data as in the previous experiment. We used timestamps to retrieve weather conditions. We use scale 1 to 3 (good) to rate weather condition like light rain, rain, strong rain, fog (`http://www.wunderground.com`). Our observation is that both time and weather context affect duration of transportation.

4. Conclusions and future work

To sum up we presented method for adaptive event reminding. Main idea is in reminding when to leave one

Table 1: Rituals discovered by our method (ID 1 to 5 are grouped).

ID	Length of period	Number of occurrences	Identification of ritual (user)
1-5	∼7 days	6 or 7 (each)	Going to work (Mon-Fri)
6	∼7 days	6	Going home (Fridays)
7	∼30 days	2	Regular visit

Table 2: Duration of transport is affected by different conditions.

time / weather	1	2	3
7:30 - 8:00	21 min.	23 min.	15 min.
8:00 - 8:30	10 min.	9 min.	11 min.
13:00 - 13:30	9 min.	10 min.	8 min.

location and prevent our user of missing other events. We considered many contexts which affects our daily rituals and designed this method to learn and adapt. We use records which are made by monitoring our users to discover, model and predict their behavior. Our solution is for mobile devices which are nowadays almost always with us and ready to operate to help us. Our application uses telestrial antennas to monitor locations which user passes. Further analyses are then applied to discover upcoming events. We also use these records to predict time needed to move from one point to another. It means that we incrementally build a map which tells us how long it takes to transport while contexts such as weather, type of transportation, and part of day are considered.

We have developed simple application which monitors user through her mobile device. Our intention was to propose an idea and evaluate analyses which were made using this data. Our aim is to move these analyses, prediction model to mobile devices as an application which works as we designed. In this case we would evaluate our solution in real life with real users. We would potentially discover other drawbacks.

In the future we plan to consider even more contexts (e.g. cultural or demographic information [2]). We mentioned how we use contexts now and how simple is it to add other contexts which also affect time needed to transport from one location to another. We want to analyze different context and enhance our model to accept also more contexts. This is associated with the representation of the map of transportation times which is currently not ready to work with amount of users. We want to provide service or even integrate into existing solution [1] to help manage user profile in the field of context aware recommending.

Acknowledgements.
This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0675/11, and it is the partial result of the Research & Development Operational Programme for the project Research of methods for acquisition, analysis and personalized conveying of information and knowledge, ITMS 26240220039, co-funded by the ERDF.

References

[1] S. Abbar, M. Bouzeghoub, and S. Lopez. Context-Aware Recommender Systems: A Service-Oriented Approach. *Personalized Access, Profile Management, and Context Awareness in Databases*, pages 1–6, 2009.

[2] F. Antonelli, G. Francini, M. Geymonat, and S. Lepsø y. DynamicTV: a culture-aware recommender. In *Proceedings of the third ACM conference on Recommender systems*, pages 257–260. ACM, 2009.

[3] L. Baltrunas and X. Amatriain. Towards Time-Dependant Recommendation based on Implicit Feedback. In *Context-aware Recommender Systems Workshop at Recsys09*, 2009.

[4] A. Bamis, J. Fang, and A. Savvides. A method for discovering components of human rituals from streams of sensor data. pages 779–788, 2010.

[5] T. Cebrián, M. Planagumà, P. Villegas, and X. Amatriain. Music recommendations with temporal context awareness. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 349–352, New York, NY, USA, 2010. ACM.

[6] M. Ciglan and K. Nørvåg. Wikipop: personalized event detection system based on wikipedia page view statistics. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1931–1932, New York, NY, USA, 2010. ACM.

[7] S. Clarke and C. Driver. Context-aware trails. *Computer*, 37:97–99, August 2004.

[8] P. Coppola, V. D. Mea, L. D. Gaspero, D. Menegon, D. Mischis, S. Mizzaro, I. Scagnetto, and L. Vassena. Context-aware browser. 99(PrePrints), 2009.

[9] M. Halvey, M. Keane, and B. Smyth. Predicting navigation patterns on the mobile-internet using time of the week. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 958–959. ACM, 2005.

[10] S. H. Hsu, M.-H. Wen, H.-C. Lin, C.-C. Lee, and C.-H. Lee. Aimed: a personalized tv recommendation system. pages 166–174, 2007.

[11] Q. Mei, D. Zhou, and K. Church. Query suggestion using hitting time. pages 469–478, 2008.

[12] K. Oku, S. Nakajima, J. Miyazaki, S. Uemura, H. Kato, and F. Hattori. A Recommendation System Considering Users' Past/Current/Future Contexts. pages 3–7, 2010.

[13] C. Vo, T. Torabi, and S. Loke. Towards context-aware task recommendation. In *Joint Conferences on Pervasive Computing (JCPC)*, pages 289–292, Tamsui, Taipei, 2010. IEEE.