The Eighteenth International Symposium on Artificial Life and Robotics 2013 (AROB 18th '13),
Daejeon Convention Center, Daejeon, Korea, January 30-February 1, 2013

Design of a user support system for event addition based on route search using schedule information

Hiroki Imamura[1] and Hiroyuki Nishiyama[1]

[1] Tokyo University of Science, 2641 Yamazaki, Noda-shi, Chiba-ken, Japan
(Tel:+81-47122-1106; Fax:+81-47122-1106)
(j7408022@ed.tus.ac.jp)

Abstract: In recent years, the ability to manage schedules using a mobile device equipped with a touch panel has increased. However, when one wishes to add a new event to the schedule book, taking into account the transit time of an existing event, it is necessary to switch to another application or website. In this paper, we design a support system for additional events to automatically determine whether the new event can be added. If this is impossible, the system proposes a time range that can have events added.

Keywords: mobile device, schedule, context, route search, event addition

1 INTRODUCTION

The recent widespread use of smartphones has encouraged efforts to provide context-aware services (e.g., i concier[1] and Google Now[2]). Here, "context" encompasses various meanings: user context, physical context, computing context, and time context. A context-aware service provides a user with the desired results, depending on the changing context. To achieve this, you must know what, where, and when to understand the current context, but the spread of smartphones is resolving this issue. The smartphone is equipped with various sensor systems (e.g., a GPS sensor, an acceleration sensor, and a touch sensor) and a camera function. In addition, we can easily develop applications that can acquire, collect, and disseminate information obtained from each function.

How to manage time for the individual is another important issue. We have complex schedules, and storing them is difficult. We have to manage events in the calendar and schedule book. With the recent development of an environment using networked computers, we can check and change a schedule anytime and anywhere by using the schedule management system in the smartphone. We believe that this kind of demand is increasing and will continue to increase in the future. In addition, by using the existing scheduling information, we can understand what should be done where and when. When planning an additional appointment, it is necessary to consider the transit time. To determine the transit time, we may use empirical data or use a web route search service. The former approach is less accurate, and the latter is time consuming.

With this as background, the purpose of this study is to design a system that supports the addition of an event while considering existing events and that can quickly add events. To achieve this, our system includes schedule management, which automatically determines whether new events can be added. The system is more accurate than an empirical determination and less troublesome than using a web route search service. In addition, if the event cannot be added, our system searches for times at which it can be added and proposes candidate them.

2 MOTIVATION

Mobile phones are used in many different situations. Therefore, research has focused on gathering information that can be observed using mobile phones with added sensors, and on the use of the collected information to suit the user 's situation. For example, Kamisaka et al. observed actual daily mobile phone user operations with minimal interference for six months[3]. They then demonstrated that machine learning could be used to predict future operations based on the collected information[4]. Additionally, they proposed a system to recommend applications, according to the situation, based on collected information and application usage. In addition, Kozawa et al. proposed a method for extracting prior advice from the web[5]. The method first identifies whether or not a given sentence provides advice. If the sentence is identified as advice, the method then identifies whether or not the sentence constitutes prior advice. Matsuura et al. created a design for a scheduling support system[6]. Their method extracts Web information based on position information and information provided by schedule management. It is possible to provide relevant information after asking the user for information, such as providing a map of the vicinity of a specified destination and the travel route from the user 's current position to that destination. The user can obtain various information related to the schedule without having to search a Web site. In addition, acquiring position information provided by GPS enables providing information that reflects the current time and the user's location.

In this paper, we collect the history of a user using the system as a life-log. We have achieved a decision-support system for individuals to use when adding events, which takes into account the transit time for events on the schedule that have already been registered.

3 SYSTEM DESIGN AND IMPLEMENTATION

This section provides an overview of the system's additional support events, the function of each module, and the internal algorithms.

3.1 System Overview

In this study, we design a mobile application system with the function of schedule management, route searching, and support for adding events. For schedule management, we created a calendar that can be synchronized with Google Calendar. For route searching, the system performs a route search based on the event information that the user has registered and displays the information necessary to reach the destination and to assist the action of the user. In addition, it accumulates a life-log search history in a local database stored in the high-performance mobile phone that is designed to speed up the search. To support adding events, the system determines whether a new event can be added. If this is not possible, the system proposes a time when the event can be added.

Figure 1 illustrates using the system.

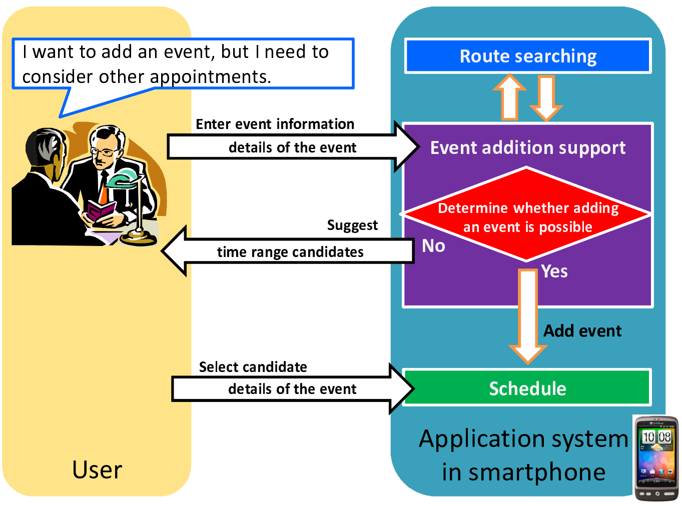

Fig. 1. Use of the system

3.2 Implementation environment

This system is implemented on an HTC Desire Soft-Bank X06HT equipped with Android 2.2 using Java language. The development environments used Eclipse3.6 (Helios) with Eclipse plug-ins and the Android SDK distributed by Google. For the local database we used SQLite, which is a standard feature on Android devices.

3.3 System configuration

Figure 2 illustrates the overall system configuration.

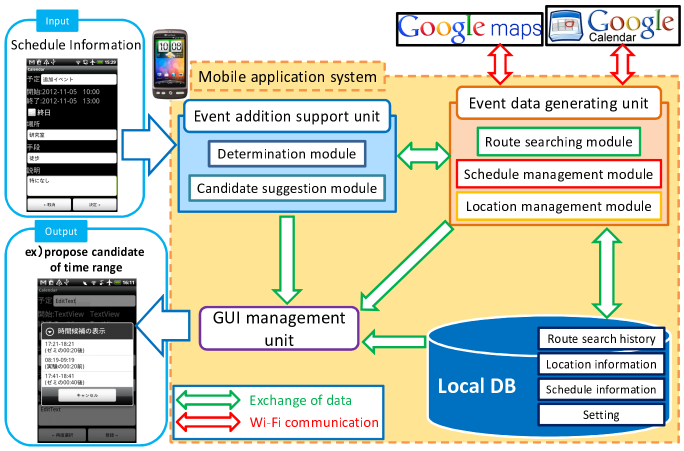

Fig. 2. System configuration

3.4 Schedule management module

The schedule management module manages event information including the title of the event, the location, the start time, the end time, and the means of transportation.

3.5 Location management module

The module is implemented to enable the user to register and manage location information while looking at a map. This module was created in order to reduce the need for typing. Thus, when entering position information into the schedule management module, it is unnecessary to describe the details of the address and buildings. Getting location information uses the location package (android.location) provided with the Android SDK.

3.6 Route searching module

The route searching module displays the daily schedule along with routing information based on the time afforded that takes into account the time margin from the current position, the start time, the end time, the location, and the transportation. Route searching uses the web service provided by Google. Analyzing the html source code of the route search results provides the necessary time, the time required, the departure time, and the arrival time. The time margin has the same meaning as in the Osako et al. study[7], and indicates that there is enough time to prepare the schedule. In addition, the starting point can be selected by the user from the following three.

1. Current position in the GPS information
2. Location that the user has previously registered
3. Position obtained from the schedule information

The destination point uses position information obtained from the schedule information. Transportation may include car, bicycle, walking, or train. The arrival time is before the scheduled time margin.

3.7 Module to determine whether adding an event is possible

This module checks to determine whether a temporal overlap exists using the route searching module and event information obtained from the schedule management module. If there is no overlap,it is possible to determine additional events from event information stored in the local database. If there is an overlap, the hours of the additional event are calculated using the overlap time and a candidate is proposed.

3.8 Module to propose candidate for time range that can accept an event

When the module determines that it is not possible to add an event, this module proposes a candidate time. Here, a collection of one-day events is referred to as an event group.

Upon calculating the time range in which an event can be added, we perform a two-way case analysis of the location where the new event is to be added.

(a) Pattern to be inserted at the beginning or end of an event group

(b) Pattern to be inserted between events

Case (a) is covered by Eq. (1), which determines the stagger time of the last or first event. We choose the larger of the overlap time or time margin to determine the stagger time. The system proposes the time the user can leave or arrive with a time margin of $T_{Stagger}$.

$$T_{Stagger} = \begin{cases} T_{Overlap} & (T_{Overlap} \geq T_{Margin}) \\ T_{Margin} & (T_{Overlap} < T_{Margin}) \end{cases} \tag{1}$$

Next, we describe Case (b). As an example, assume that you are adding a new event (event z) to existing events (events x and y), as depicted in Fig. 3. The necessary time is defined as the time that you need to consider when you want to add event z. The necessary time is the total of the running time of event z plus the time margin and the travel time between events x and z and the travel time between events z and y. In addition, the time slot between events x and y indicates the time between the start time of event y and the end time of event x.

The system proposes up to three candidate time ranges when the event can be added. Thus, there is a possibility that when the proposals between the same events, the system propose time range biased. To exclude this possibility, it is necessary to consider where new events can be inserted between existing events. Equation 2 gives the maximum number of events that can be added.

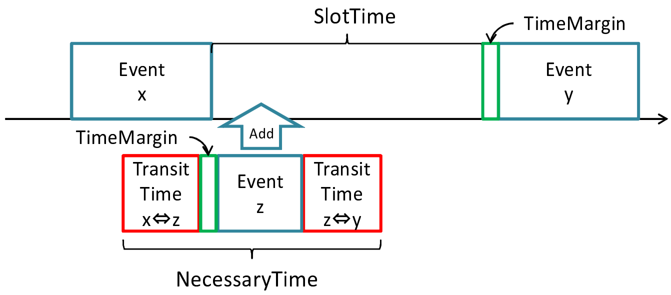

Fig. 3. Adding an event

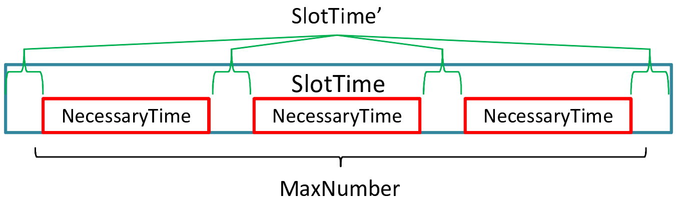

Fig. 4. Maximum number of events that can be added

$$\begin{aligned} N_{max}(x,y,z) &= \lfloor \frac{T_{Slot}(x,y)}{T_{Necessary}(x,y,z)} \rfloor \\ T_{Necessary}(x,y,z) &= T_{Transit}(x,z) + T_{Transit}(y,z) \\ &\quad + T_{Event}(z) + T_{Margin} \\ T_{Slot}(x,y) &= T_{Start}(y) \quad T_{End}(x) \quad T_{Margin} \end{aligned} \tag{2}$$

To calculate the time range in which an event can be added, we obtain the gap time between the candidates (SlotTime' in Fig. 4). Correcting the value calculated by Eq. (2) so as not to exceed the number of candidates proposed, we obtain the formula for the gap time between the candidates in Eq. (3).

$$\begin{aligned} T'_{Slot}(x,y,z) &= \frac{T_{Slot}(x,y) \quad T_{Necessary}(x,y,z) \times N(x,y,z)}{N(x,y,z)+1} \\ N(x,y,z) &= \begin{cases} N_{Candidate} \\ \quad (N_{Candidate} < N_{max}(x,y,z)) \\ N_{max}(x,y,z) \\ \quad (N_{Candidate} \geq N_{max}(x,y,z) > 0) \\ 0 \\ \quad (otherwise) \end{cases} \end{aligned} \tag{3}$$

Finally, Eq. 4 calculates the start time and end time for candidate w-th.

$$\begin{aligned} T'_{Start}(x,y,z,w) &= T_{End}(x) + wT'_{Slot}(x,y,z) \\ &\quad +(w \quad 1)T_{Necessary}(x,y,z) \\ T'_{End}(x,y,z,w) &= T_{Event}(z) + T'_{Start}(x,y,z,w) \end{aligned} \quad (4)$$

4 CONCLUSION

In this paper, we designed a user support system for additional events, taking into account the existing events. It determines the additional events and proposes a time range in which events can be added. This system enables users to schedule new events more smoothly than by conventional decision-making systems. In addition, problems related to determining transit time (such as low precision, or being time-consuming) were addressed in order to facilitate using a life-log. Future work will demonstrate the effectiveness of this system by evaluating the accuracy of the proposed candidates.

REFERENCES

[1] i concier, http://www.nttdocomo.co.jp/service/customize/iconcier/

[2] Google Now, http://www.google.com/landing/now/

[3] Daisuke Kamisaka, Shigeki Muramatsu and Hiroyuki Yokoyama (2008), A Study on Situation-Adaptive Operation-Assisting Method for Mobile Phone. Information Processing Society of Japan SIG Technical Report 2008-UBI-20(110):33-38

[4] Daisuke Kamisaka, Takeshi Iwamoto, Shigeki Muramatsu and Hiroyuki Yokoyama (2009), Long-term Data Analysis and Operation Prediction for Context-Aware UI of Mobile Phones. FIT2009 8(4):23-29

[5] Shunsuke Kozawa, Masayuki Okamoto, Shinichi Nagano, Kenta Cho and Shigeki Matsubara (2011), Advice Extraction from Web for Providing Prior Information Concerning Outdoor Activities. Proceedings of the 4th International Conference on Intelligent Interactive Multimedia Systems and Services (IIMSS 2011):251-260

[6] Hiroshi Matsuura, Tetsuya Tsukamoto and Hiroyuki Nishiyama (2009), Schedule management support system using Web information. IPSJ National Convention 71st

[7] Takashi Osako, Itaru Kuramoto, Yu Shibuya, Yoshihiro Tsujino and Mitsuru Minakuchi (2009), Prior notification technique of tasks/schedules taking account of time margin. Information Processing Society of Japan SIG Technical Report, 1009-HCI-131(4):23-30