Using Personal Schedule Information to Aid the Data Selection in LAS Applications

Yi-Min, Chiang [a] *and Jung-Hong Hong* [b]

[a] *Research Assistant, Intelligent Geographic Information System Lab (IGIS Lab), Dept. Geomatics National Cheng Kung University,*

No.1, University Road, Tainan City 701, Taiwan (R.O.C.); TEL: +886-6-2757575
E-mail: cs94276@gmail.com

[b] *Associate Professor, Intelligent Geographic Information System Lab (IGIS Lab), Dept. Geomatics National Cheng Kung University,*

No.1, University Road, Tainan City 701, Taiwan (R.O.C.); TEL: +886-6-2757575
E-mail: junghong@mail.ncku.edu.tw

KEY WORDS: GIS, Location-based service, Location-aware service, Schedule, Calendar

Abstract: The applications of location-based services (LBS) have received tremendous attention in recent years. With its "push" characteristics, one distinct feature of the location-aware service (LAS) is the servers can automatically supply information to users. Although this advantage enables information providers to actively locate candidate users and increase the value of their services, its success, however, heavily relies on the correct and effective selection of transferred information based on users' needs and conditions, otherwise users may be annoying about the garbage information that continuously presented to them and quickly lose interests of using the services. This paper proposes an integrated approach, namely, Personalized Location-Aware Service (PLAS), by adding personal profiles and schedule into the development of location-aware systems. As personal schedule records pre-scheduled events, it provides valuable information about when, where and what of persons' daily plan. The developed PLAS can thus precisely determine individual user's free time and only supply information about the events they have sufficient time to participate during their free time. According to users' spatial and temporal conditions, events stored in the database are automatically filtered in advance to reduce inappropriate information. As users move and time goes, this list of suggested events is continuously and dynamically updated according to users' new status and remaining time. College students' campus life is chosen for test in this paper because students have fixed daily class schedule and campus events are abundant and versatile. Although the test result is still preliminary, the developed system prototype has already demonstrated the feasibility and improvements by including personal schedule information as the spatial and temporal constraints for filtering event information in LAS-based applications. More contextual information will be added in the future to the proposed model to increase the intelligence for information selection.

1. INTRODUCTION

In 2012, the global smartphone shipment expects to gain a 32% increase and reaches 146 million units (Strategy Analytics). This revolution brought booming impacts to the wireless internet market. As most of the smartphones are equipped with built-in GPS chips, the availability of coordinate information makes location-enable search and advertisement possible. The IEMR report (IE Market Research) projected the global market for GPS navigation and location-based services (LBS) to grow 51.3% and increase to $13.4 billion in 2014. This implies a huge growth of the number of new users, professional or not, and fast expansion of business market in the future. LBS are services accessible with mobile devices through the mobile network and that have the ability to make use of the location of the terminals (Virrantaus et al. 2001). Mobile device, wireless network and positioning equipment are basic elements for developing LBSs. Based on its location-dependent nature, a LBS is capable of collecting user's location, and allows the data provider to supply information via wireless network according to his or her current location and application needs. The successful operation of LBS thus serves to bridge the communication between users and data providers via users' "continuously changing location."

Depending on how interaction is made between users and providers, the modes of data delivery of LBS are subdivided into two major approaches: the pull mode and the push mode. In the pull mode, users specify their requests and the LBS responses with information that fits users' demands. On the other hand, the push mode automatically feeds information the service deems useful to users without his or her request (Jiang & Yao, 2006). Location-Aware Services (LAS) are widely known as "push" services. To improve the quality of the distributed information, services based on push mode must take users' context into consideration. Kaasinen (2003) defines context as "the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user". Following this concept, a context-aware service can be regarded as a type of service that provides relevant information to the user based on their current task (Van Setten et al., 2004). Location-aware service is a special case of context-aware service using location information as the main context. In addition to the low-level context, such as location and time, high-level context like personal preference or social activity can also be considered (Chen & Kotz 2000). It is no doubt that the more context LAS has about individual users, the accurate its service content is. For example, ComMotion (Marmasse & Schmandt, 2000) tracks the places users frequently visit over time and periodically invite them to name or classify the coordinate information, such that when users arrive at certain places, he or she will receive reminder messages defined by themselves. Designed for travelers, COMPASS (Van Setten, Pokraev, & Koolwaaij, 2004) is a recommender system integrated with a context-aware application platform. The system framework of COMPASS consists of context manager, service registry and recommendation engine based on user's interests and his or her current context to provide adaptive information. The registry service allows third parties to offer their services. The context manager retrieves the information of user's context, such as location, time, weather and agenda, from the appropriate context services. Several prediction methods, e.g. social filtering, case-based reasoning, item-item filtering and category learning are developed in the recommendation engine according to the category of POI (Point of Interests).

As the most ideal scenario for LAS is to always supply customized information that best fits users' continuously changing situation, we proposed to develop a Personalized Location-Aware Service (PLAS) in this paper. Compared with the current distance-driven LAS, one distinct feature for PLAS is the addition of personal schedule information, which helps to determine when users are free (time periods without scheduled events), where they are when they become free, and where they shall go for the next scheduled event. The PLAS aims to push only the information of events where users have sufficient time to participate during their free time, that is, users will be able to participate in the pushed event after his or her free time begins, and arrive at the destination of the next scheduled activity in time. The developed algorithm can adapt to the changes of time and users' location and continuously adjust the pushed information. A pushed event is deleted if the remaining time does not permit the users to participate, so that only the useful information is provided to users. The following of the paper is organized as follows: section 2 introduces the conceptual design of our push algorithm, section 3 discusses the system architecture and our test results. Finally, section 4 concludes our major findings and future research directions.

2. PLAS DATA SELECTION ALGORITHM DESIGN

In order to use personal schedule information to aid the data selection algorithm in LAS applications, we need to analyze the time composed of user's daily life and the characteristic of events. The events we are talking about in this paper are the activities which user can participate in. The scheduled events are the planned events in personal schedule .We know that people are always at a particular location in a moment, and their location may change over time. The event we discuss here General events and personal scheduled events comprise people's daily life, and these kinds of events both have its specific location and time. The assumption of PLAS algorithm is to determine individual user's free time from accessing his schedule and only provide information of the events he has sufficient time to participate during his free time. Time constrains and location restricts are the key of designing PLAS algorithm, so the factors relevant to them need to be parameterized in order to make flexible use of it. In the following, we list the considerations and declare specific variables and function to express the method of this algorithm, and illustrate the flow chart of the solution.

2.1 Related factors analysis

(1) Factors relevant to user

Every individual user is expressed as $P_i,\ i = 1, \dots, N$. i is the unique identifier to identify an individual user. $P_i.name$ and $P_i.loc$ represent user's name and current location, respectively. A person's daily schedule is composed of a series of scheduled events and free time, as figure 1shows.

(2) Factors of user's schedule

We assume that user's schedule is composed of his personal scheduled event and the free time between them, as figure 1 shows. The scheduled event is represented as $S_k,\ k = 1, \dots, P$. The free time is expressed as $F_l,\ l = 1, \dots, Q$. Variables k and l indicate the time order of scheduled event and free time, separately. A specific user's scheduled event and free time are expressed as $P_i.S_i$ and $P_i.F_l$, respectively. The description, location, begin and end time and the time interval are the features we concern about the scheduled event, which are represented as $S_k.description$, $S_k.loc$, $S_k.begin$, $S_k.end$, $S_k.interval$, respectively. Regarding to the free time, the time begin and end and it's time interval are then represented as $F_l.begin$, $F_l.end$ and $F_l.interval$.

Figure 1: Personal daily time axis

(3) Factors related to event

The events we discussed here contain lots of activity which can be classified into two categories depends on whether user needs to involve throughout the entire event or not, the following gives the detail description of the two classes of events:

- Non-interruptible events

  User is required to participate in the entire non-interruptible events such as speech, lecture, concert, etc. Midway participation or leave in the middle of this type of event are not recommended.

- Free-participation events

  In the entire free-participation events, users are free about when to participate in this kind of event, and they can leave right after finishing the task, this kind of events such as blood donation, user may spend 30 minutes donating their blood within its service hours.

Event is expressed as $E_j, j = 1, \ldots, M$, variable j is used to unique identify each event. The descriptions, location, type, begin and end time and the time interval of the event are factors important to obtain, which parameterized by$E_j.description$, $E_j.loc$, $E_j.type$, $E_j.begin$, $E_j.end$, $E_j.interval$, respectively.

(4) Travel cost

The travel cost function help the system determine the time cost of moving between different locations. The function is expressed as $TC(location1,\ location2)$, by inputting the coordinates values of two specific locations into the function, the output of time cost to travel between these places can thus be returned by travel cost function. For instance, the time cost of traveling from user's current location to a particular event can be calculated by the input of two sets of coordinates values into the function, which is expressed as $TC\left(P_i.loc\ ,\ E_j.loc\right)$ shown in figure 2, then the output of time variable will be returned.

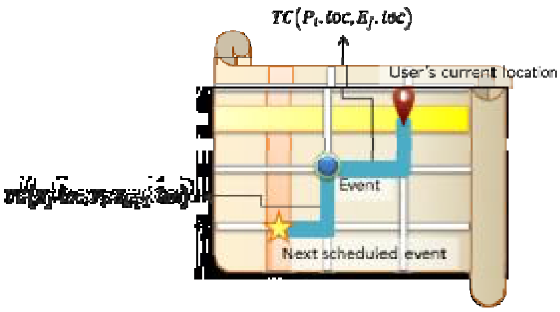

Figure 2: Travel cost

(5) Within function

The "Within function" is to determine whether a time point is within a time interval, it is represented as $within(time\ point, time\ interval\ )$, the output of this function is a boolean value. If the time point is within the time interval, the output will be true, else it will be false. This function is used to judge if user's current time is within the free time of his schedule.

2.2 Algorithm Development

As we mentioned before, the assumption of PLAS algorithm is to determine individual user's free time from accessing his schedule and only provide information of the events he has sufficient time to participate during his free time, this idea focus on the time constraints of user's free time and the travel cost of moving among the locations of user, event and his next scheduled event, which is implemented by the procedure illustrated as figure 2's flow diagram shows, the four steps will be described in-depth in the following:

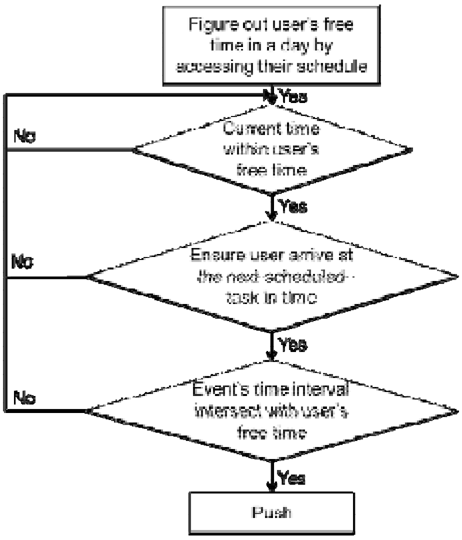

Figure 2: Flow diagram of the PLAS data selection algorithm

Step 1: Find out user's free time on that day from his personal schedule.

The information of scheduled events is recorded in user's personal schedule, figure 3 illustrates students' class schedule in that day. By equation 1, we can figure out the time interval of user's specific period of free time.

$$P_i.F_l.interval = P_i.S_{k+1}.begin - P_i.S_k.end \quad (1)$$

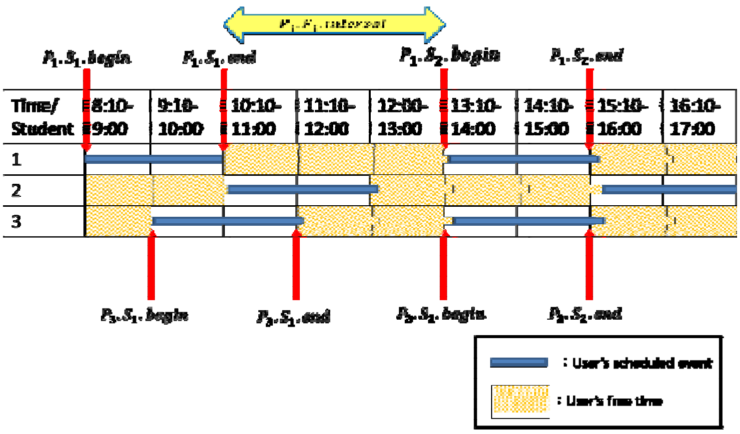

Figure 3: Personal schedule

Step 2: User's current time is compared with the time interval of user's free time to check if it is within user's free time or not. It is the threshold of the pushing mechanism which decides whether system push the appropriate POIs to the user, and it can be expressed as equation 2. Once user's current time match this equation, the system will start offering service.

$$within(CurrentTime\,,\, P_i.F_l.interval) = true \quad (2)$$

Step 3: Determine whether user will arrive at the next scheduled task in time. In order to achieve this target, the restriction that user's current time ($CurrentTime$) plus the travel cost from user's current location to the taken place of event ($TC(P_i.loc,\, E_j.loc)$), time required for the event ($E_j.interval$), plus the travel cost from position of the event to the next scheduled event ($TC(E_j.loc,\, P_i.S_{k+1}.loc)$) be smaller than the beginning time of next scheduled event must be satisfied, as equation 3 shows. In particular, if the time user arrives at the event is earlier than beginning of the event, indicating user will have to wait until the event start. In this case, we just need to determine if the beginning of the event plus the time required for the event and the travel cost between the locations of the event and next scheduled event is earlier than the beginning of next scheduled event. The time axis of above-mentioned time is illustrated in figure 4.

The definition of $E_j.interval$ will be differed depends on different categories of the events. For non-interruptible events, $E_j.interval$ is the time required to participate in the entire event. In regard to free-join events, $E_j.interval$ is the estimated time required to finish that kind of events. For the sales activities, the estimated time of purchase may be 10 minutes, so the period of $E_j.interval$ may vary from event to event.

$$\textbf{\textit{If}} \quad E_j.begin \geq CurrentTime + TC(P_i.loc\ ,\ E_j.loc)$$
$$\textbf{\textit{Then}} \quad E_j.begin + E_j.interval + TC(E_j.loc,\ P_i.S_{k+1}.loc) < P_A.S_{k+1}.begin$$
$$\textbf{\textit{Add to the push list}}$$

$$\textbf{\textit{If}} \quad E_j.begin < CurrentTime + TC(P_i.loc\ ,\ E_j.loc)$$
$$\textbf{\textit{Then}} \quad CurrentTime + TC(P_i.loc,\ E_j.loc)\ + E_j.interval + TC(E_j.loc,\ P_i.S_{k+1}.loc) < P_A.S_{k+1}.begin$$
$$\textbf{\textit{Add to the push list}} \tag{3}$$

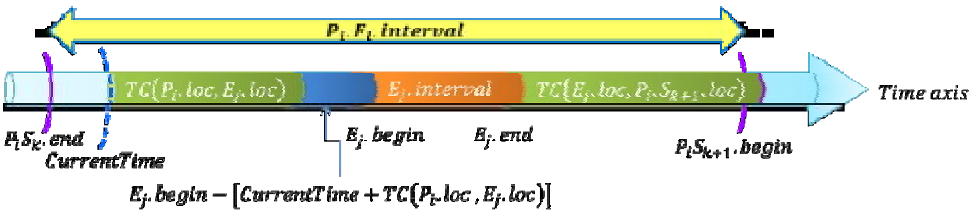

Figure 4: Time axis

Step 4: After confirming that user will arrive at the next class in time, we will further test if the time interval of the filtered events be intersect with user's free time in the fourth step. This is intended for preventing users from going to the event that out of its business hours. As we mentioned before, events can be divided to "non-interruptible events" and "free-join events", the following determine the restrictions based on these categories, respectively.

Non-interruptible events:

These kinds of events must be attended when the activities begin and leave at the end of it. Due to that, the beginning time of the event should be later than the time user arrive at it, and the event must be finished before the end of user's free time. This concept is implemented by equation 4.

$$\textbf{\textit{If}}\ \ E_j.begin > CurrentTime + TC(P_i.loc,\ E_j.loc)\ \ and\ E_j.end + TC(E_j.loc,\ P_i.S_{k+1}.loc) < P_i.S_{k+1}.begin$$
$$\textbf{\textit{Push the event}} \tag{4}$$

Free-participation events:

Users are not necessary to involve throughout the entire free-join event, it indicates that the whole activity time need not to fall entirely within user's free time but needs to intersect with the time period from the time user arrive at the event to the end of user's free time. To meet this idea, equation 7 needs to be satisfied.

$$\textbf{\textit{If}}\ \ E_j.end > CurrentTime + TC(P_i.loc,\ E_j.loc)\ \ and\ E_j.begin < P_i.S_{k+1}.begin$$
$$\textbf{\textit{Push the event}} \tag{5}$$

If steps 1 to 4 are all satisfied, the system will start pushing information of those events, otherwise, the LAS data selection mechanism will keep searching for the events until it fulfilled these conditions within user's free time.

3. SYSTEM ARCHETECTURE

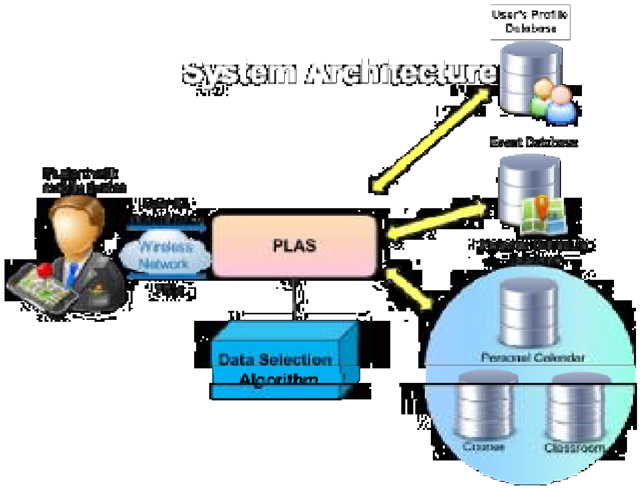

Figure 5: System Architecture

The system architecture is as the figure 5 shows. The mobile device held by user automatically sends user's id and coordinates of his current location to PLAS, through its data selection algorithm, user's personal schedule can be accessed and further determine their free time of a day. User's free time will be compared with event's database to decide appropriate events to recommend, finally return the information of filtered events to the user. College student's campus life is chosen for the test in this paper, so the databases are created based on student's curriculum and campus events. The components of the system architecture will be described in detail below:

3.1 User's profile database

Focusing on the students in college, there are some fields must be recorded in order to uniquely identify every student. Student's ID, name, major, etc., this information can be retrieved from the database of college's computer center. This information must be held by the system in order to access every student's personal profile.

3.2 Event database

The ID, description, type, location of the event and the time event begins and ends are necessary to be recorded in event database. With event database, the system is able to offer the information of nearby event information by processing PLAS data selection algorithm.

3.3 Personal schedule database

Personal calendar, course and classroom databases are created to constitute a complete personal schedule database.

3.3.1 Personal course table

In personal course table, the student ID from student profile table, the ID of courses taken by the student and the class time order in a day is the necessary information for the system discovers the courses taken by an individual student and further determines his free time.

3.3.2 Course table

The college's course enrollment system utilizes department code together with course number to uniquely identify each course. Following this rule, the course ID, department code, course number, course name, classroom ID and the beginning and end of time of each course will be recorded in course table. The PLAS data selection algorithm is able to link personal course table with course table by joining foreign key attribute "course ID" to learn the details of course information taken by a specific student.

3.3.3 Classroom table

Classroom is the place student learning courses. The classroom table plays the role of indicating spatial information in personal schedule database. The attribute must be recorded contains classroom ID, school name, district of campus, name of building, room number and its coordinates. The course database is feasible to join classroom database on the foreign key "classroom ID".

The relations between tables of personal schedule database are shown in figure 6. The student ID from student profile table and the course ID from course table are linked to the personal course table to figure out the courses taken by a specific student. In figure 7, we illustrate the concept of personal schedule database of an individual student. Student's class schedule can be recorded as the table in the left-hand side, the classroom ID is linked to the classroom table in order to indicate the location of classroom in personal schedule database.

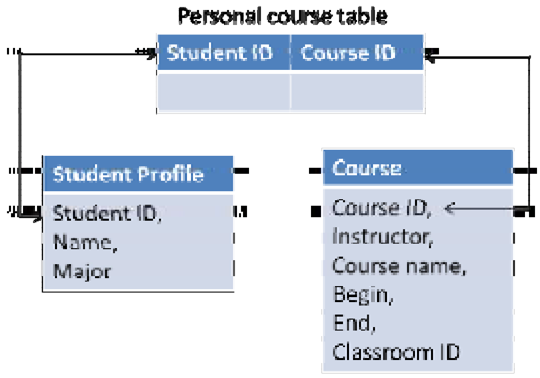

Figure 6: Relations between tables of personal schedule database

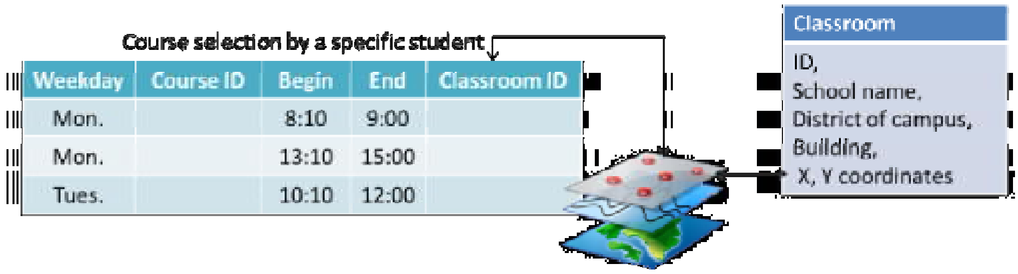

Figure 7: Personal Schedule database

By querying from the personal schedule database, the system can thus retrieve the information about the courses taken by an individual student.

4. Test and Analysis

To test the proposed PLAS push algorithm, we choose campus event information for test in this paper. For each semester, students are required to select a number of courses. Since courses are offered at specific time slots and location on a weekly basis, we assume every student has a fixed personal schedule. Since courses are data with spatial and temporal component, the PLAS can analyze the free time for a particular student, where he or she is when the free time begins and where and when his or her next scheduled event (next course) is. Figure 8 shows a personal schedule of a student named Steve. At this particular day, his first course starts at 8:10 and ends at 9:00. He then has 70 minutes of free time. His next course is at a different building from 10:10 to noon. In the afternoon, he has a 2-hr course. After 16:00, he is free and can go home or participate in any event he likes. As mentioned earlier, his schedule is composed of a series of scheduled events (courses he need to go) and free time.

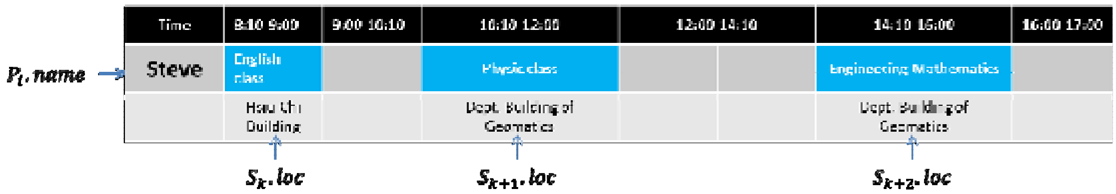

Figure 8: Steve's schedule in a day

Five events are simulated, event A is a charity bazaar held by a student association, event B is a blood donation activity, event C is the volleyball match in the campus, event D is an English lecture offered by the library and event E is the NCKU museum trail. Events A, B, C and E are free-participation events, and event D is non-interruptible event. When Steve's English course ends at 9:00, his position is at the Hsiu-Chi building. Following Equation 2, the PLAS detects he is free and will be free for 70 minutes until next scheduled course begins. All of the five events are tested. For free-join events, only those meet the condition listed in equation 5 are selected. For non-interrupted events, the condition listed in equation 4 must be tested true. Table 1 shows the test results at 9:00. Although all of the five events are pushed, the arrival time arriving at the classroom of the next scheduled course is different from each other. Figure 9 shows the test map interface, the two classrooms are represented by a star symbol and users' current location is represented by the pin symbol. If Steven chooses to go to event B for donating blood, he would spend 11minutes and 32seconds to move from his current location to the place where the blood donation activity is held. After the blood donation, he will spend 13 minutes and 12 seconds to go to the Dept. of Geomatics, where his next course is held. The reason this information is pushed is because the PLAS determine that Steve has sufficient time to participate in the event and arrive his next course in time (before 10:10).

Event's name	$E_j.begi$	$E_j.begin$	$E_j.end$	$E_j.time$	$TC(P_i.loc, E_j.loc$	$TC(E_j.loc, P_i.S_{k+1}.loc$	$CurrentTime + TC(P_i.loc, E_j.loc) + E_j.time + TC(E_j.loc, P_i.S_{k+1}.loc$	Display or not
A	Free-participation	09:00:00	17:00:00	10 minutes	12 min 8 sec	01 min 04 sec	09:23:13	Yes
B	Free-participation	09:30:00	12:00:00	20 minutes	11 min 32 sec	13 min 12 sec	09:44:45	Yes
C	Free-participation	08:00:00	10:00:00	20 minutes	13 min 54 sec	11 min 22 sec	09:45:17	Yes
D	Non-interruptible	09:30:00	10:00:00	30 minutes	7 min 59 sec	4 min 48 sec	09:42:50	Yes
E	Free-participation	09:00:00	16:00:00	40 minutes	15 min 55 sec	7 min 17 sec	09:53:14	Yes

Table 1: Parameters at the time of 9:00

The time comes to 9:20, Steve is now at the place of event B, and the parameters in this situation are listed in table 2. The arrival time at the next class is represent as the penultimate column, $CurrentTime + TC(P_i.loc,\ E_j.loc)\ + E_j.time + TC(E_j.loc,\ P_i.S_{k+1}.loc)$, and we can observe that the arrival time at event E is out of Steve's free time, so it won't be display at the map view. Event D also won't be appeared on the screen, although its arrival time, 10:09:12 is earlier than the start time of next class, but it must to fulfill equation 4 due to the non-interruptible characteristic, therefore, the reason why it won't be pushed is that the time arrives at the event which is calculated by $CurrentTime + TC(P_i.loc,\ E_j.loc)$, 9:34:22, is later than the beginning time of the entirely-participated event D. The map view at 9:20 is shown in figure 10.

Event's name	$E_j.type$	$E_j.begin$	$E_j.end$	$E_j.time$	$TC(P_i.loc, E_j.loc)$	$TC(E_j.loc, P_i.S_{k+1}.loc)$	$CurrentTime + TC(P_i.loc, E_j.loc) + E_j.time + TC(E_j.loc, P_i.S_{k+1}.loc)$	Display or not
A	Free-participation	09:00:00	17:00:00	10 minutes	13 min 08 sec	1 min 4 sec	09:44:23	Yes
B	Free-participation	09:30:00	12:00:00	20 minutes	0 min	13 min 12 sec	09:53:12	Yes
C	Free-participation	08:00:00	10:00:00	20 minutes	4 min 42 sec	11 min 22 sec	09:56:04	No
D	Non-interruptible	09:30:00	10:00:00	30 minutes	14 min 22 sec	04 min 49 sec	10:09:12	No
E	Free-participation	09:00:00	16:00:00	40 minutes	11 min 40 sec	07 min 17 sec	10:18:58	No

Table 2: Parameters at the time of 9:20

If Steve decides to continuously wait for blood donation. When the time reaches 9:36, the remaining time is not enough for him to donate blood and arrive at the next scheduled place in time, so event B is excluded from the pushed list. At this time, only the information of event is pushed, which demonstrates the information PLAS pushes can dynamically adapt to the continuously changing status of the spatial and temporal situation. Figure 11 shows the displayed events.

Event's name	$E_j.type$	$E_j.begin$	$E_j.end$	$E_j.time$	$TC(P_i.loc, E_j.loc)$	$TC(E_j.loc, P_i.S_{k+1}.loc)$	$CurrentTime + TC(P_i.loc, E_j.loc) + E_j.time + TC(E_j.loc, P_i.S_{k+1}.loc)$	Display or not
A	Free-participation	09:00:00	17:00:00	10 minutes	13 min 08 sec	1 min 4 sec	10:01:00	Yes
B	Free-participation	09:30:00	12:00:00	20 minutes	0 min	13 min 12 sec	10:10:00	No
C	Free-participation	08:00:00	10:00:00	20 minutes	4 min 42 sec	11 min 22 sec	10:12:52	No
D	Non-interruptible	09:30:00	10:00:00	30 minutes	14 min 22 sec	04 min 49 sec	10:25:59	No
E	Free-participation	09:00:00	18:00:00	40 minutes	11 min 40 sec	07 min 17 sec	10:35:45	No

Table 3: Parameters at the time of 9:36

Figure 9: The time of 9:00

Figure 10: The time of 9:20

Figure 11: The time of 9:36

5. CONCLUSIONS

In this paper, we focus on the time constrains in LAS application, considering events available time and user's personal schedule to design our data selection algorithm. User will receive information of events only at its available time and arrive at the next destination on time. Thus, registration mechanism is necessary to let the third parties fill in their business time and the average time spending on it. With additionally consider personal schedule, the situation that arriving POIs at its rest time won't happen, also student doesn't need to be embarrassed when attending to class late since our system won't recommend events that its total costing time larger than user's free time. Our system not only use personal schedule information to aid the data selection in LAS applications, but also provides a platform for campus information flow between school and student based on the context about student's curriculum data, it offers a new way of communication in campus.

6. REFERENCES

IE Market Research: 3Q.2010 Global GPS Navigation and Location Based Services Forecast, 2010 - 2014: Global market for GPS navigation and location based mobile services to rise to $13.4 billion in 2014, a CAGR of 51.3% (Jul 19, 2010)
https://www.iemarketresearch.com/Members/Reports/3Q-2010-Global-GPS-Navigation-and-Location-Based-Services-Forecast-2010--2014-Global-market-for-GPS-navigation-and-location-based-mobile-services-to-rise-to-13-4-billion-in-2014-a-CAGR-of-51-3--RID1480-1.aspx （search on 2012/10/8)

Strategy Analytics: Global Smartphone Shipment Growth Slows to 32 Percent in Q2 2012
http://www.businesswire.com/news/home/20120726006911/en/Strategy-Analytics-Global-Smartphone-Shipment-Growth-Slows

Chen, G., & Kotz, D. (2000). A survey of context-aware mobile computing research.

Jiang, Bin, & Yao, Xiaobai. (2006). Location-based services and GIS in perspective. *Computers, Environment and Urban Systems, 30*(6), 712-725. doi: 10.1016/j.compenvurbsys.2006.02.003

Kaasinen, Eija. (2003). User needs for location-aware mobile services. *Personal and Ubiquitous Computing, 7*(1), 70-79. doi: 10.1007/s00779-002-0214-7

Marmasse, Natalia, & Schmandt, Chris. (2000). Location-Aware Information Delivery with ComMotion.

Van Setten, M., Pokraev, S., & Koolwaaij, J. (2004). *Context-aware recommendations in the mobile tourist application COMPASS*.

Virrantaus, K., Markkula, J., Garmash, A., Terziyan, V., Veijalainen, J., Katanosov, A., & Tirri, H. (2001). *Developing GIS-supported location-based services*.