Huang, H., 2011. Supporting Smart Mobile Navigation in a Smart Environment. In S. A. Ahson & M. Ilyas, eds. Location-Based Services Handbook: Applications, Technologies, and Security. CRC Press, pp 109-129.

Support smart mobile navigation in a smart environment

Abstract:

With more and more active or passive devices/sensors being augmented, our environment has become smarter. Also in the era of Web 2.0, the concept of "Web-as-participation-platform" has been fully adopted in the ICT society. This paper focuses on the question of how mobile navigation services can benefit from Smart Environment/Ambient Intelligence and Web 2.0/collective intelligence. After setting up a smart environment, a mobile navigation service is designed to support users' wayfinding, facilitate users' interaction and annotation with the smart environment, and collect user generated content (UGC). Based on UGC, this paper designs several "collective intelligence"-based route calculation algorithms to illustrate the benefits of combining mobile navigation services, smart environment, and Web 2.0, such as providing "the nicest route", "the least complex route", "the most popular route", and "the optimal route". Finally, this paper concludes that mobile navigation services in smart environment can help to explicitly and implicitly collect user generated content (collective intelligence), and thus provide users with a new experience and smart wayfinding support (e.g., "collective intelligence"-based route recommendations).

Keywords:

Smart Environment, user generated content, collective intelligence, route calculation, mobile navigation service, Location Based Services

1. Introduction

The ubiquity of mobile devices (such as cell phones and PDAs) has led to the introduction of Location Based Services (LBS), or Location-Aware Services. A system can be called a Location Based Service (LBS), when the position of a mobile device – and therefore the position of the user – is somehow part of an information system (Gartner 2007). LBS aim at providing information/services relevant to the current location and context of a mobile user.

In this paper, we will focus on one of the most important LBS applications - mobile navigation service, which provides wayfinding guidance in an unfamiliar environment. In our daily life, we always encounter wayfinding problems when arriving a new place, such as “what’s the way from Train satiation to City hall”. Usually, we ask people in the surrounding for advices, or plan our trip in advance on paper maps or web maps (such as Google map). With the help of mobile navigation services (with GPS or other positioning technologies), users can easily find their way in a new environment. One of the successful mobile navigation systems is car navigation which has been widely used and trusted by car drivers all over the world. Recently, the increasing ubiquity of personal mobile devices (such as cell phones and PDAs) triggers a move towards mobile pedestrian navigation systems. Comparing the information needs of car drivers and pedestrians Rehrl & Leitinger (2007) conclude that the main differences are in the degrees of freedom in movement, the velocity of movement and thus the opportunity for perceiving the environment and the resolution of space is larger.

Technology available today is rich. Currently, with the rapid advances in enabling technologies for ubiquitous computing, more and more active or passive devices/sensors are augmented in the physical environment, our environment has become smarter. This abundance of technologies has given place to the new notions of “Smart Environment (SmE)” and “Ambient Intelligent (AmI)”. The basic idea behind SmE and AmI is that “by enriching an environment with technology (sensor, processor, actuators, information terminals, and other devices interconnected through a network), a system can be built such that based on the real-time gathered and the historical data accumulated, decisions can be taken to benefit the users of that environment” (Augusto and Aghajan 2009). One of the most popular instantiations of these areas is the concept of smart home. With the increasing ubiquity of smart environments, the question of how mobile pedestrian navigation systems can benefit from SmE and AmI should be carefully investigated. However, to our knowledge, there is little work on that.

Another interesting event in the field of ICT (Information and Communication Technologies) is the gradual evolution of Web 1.0 to Web 2.0. Compared to “Web-as-information-source” in Web 1.0, Web 2.0 adopts the notion of “Web-as-participation-platform” (Wikipedia 2009a). In Web 2.0, users can actively contribute to the web. However, the concept of “Web 2.0” has not been introduced to mobile navigation services. Most of the current mobile navigation systems are limited to provide richer, just-in-time information (navigation instructions) for users. However, a lot of users are not satisfied with simply being passive consumers, but rather want to be active contributors (Kang et al. 2008). By encouraging users to annotate physical space with experiences, questions, and opinions during navigation, which reflect the perspective of the people who navigate in the space and the activities that occur there, the mobile navigation services can fulfill users’ intrinsic desire to share their experiences (with friends, or even with other people they don’t really know) and thus provide users with a new experience during wayfinding.

In the era of Web 2.0, users are encouraged to contribute to the web. As a result, the term user generated content (UGC) entered mainstream usage since 2005 (Wikipedia 2009c). It refers to “various kinds of media content, publicly available, that are produced by end users”. UGC on the web reflects users’ collective intelligence, and can be viewed as the “wisdom of the crowds” (James Surowiecki 2005). How can UGC be used to generate value/benefits for mobile navigation services? Recommendation system from the E-commerce field (such as Amazon.com) may be one of the most promising solutions for this question. Recommendation systems can help to make collective intelligence useful. However, little work has been done on applying recommendation technology to generate value from UGC for mobile navigation services.

This paper attempts to introduce the notions of SmE/AmI and Web 2.0/collective intelligence into mobile navigation services. We propose that mobile navigation systems in smart environment can help to collect (gather and accumulate) related information (information about users and system, user generated content, etc.), and thus provide users with a new experience and smart wayfinding support (e.g., context-awareness and “collective intelligenc”- based route recommendations).

The rest of this paper is structured as follows. Section 2 presents the related research. In section 3, we deploy some devices/sensors to our office building and set up a smart environment as a testbed for our mobile navigation service. Section 4 discusses the issue of users’ interaction and annotation, collecting user generated content in this smart environment. In section 5, we investigate how user generated content can be used to provide “collective intelligence”-based route recommendations, such as the nicest route, the least complex route, the most popular route, and the optimal route. Section 6 designs some issues on context-awareness of our mobile navigation service. Finally, section 7 draws the conclusions and presents the future work.

2. Related work

Our research concerns how mobile navigation services can benefit from Smart Environment/Ambient Intelligence and Web 2.0/ collective intelligence. This issue mixes several mainstream trends and concepts, such as Location Based Services, Smart Environment, Web 2.0, User Generated Content (collective intelligence, wisdom of the crowds), etc. From these aspects, we summarize the related works.

2.1 LBS in a smart environment

Computing has become increasingly mobile and pervasive, which demands applications that are capable of recognizing and adapting to highly dynamic environments while placing fewer demands on user's attention (Henricksen et al. 2002). It is widely acknowledged that context-awareness can meet these requirements. As one type of ubiquitous computing, in order to provide a good usability, LBS should be context-aware and adapt to dynamic environment.

Dey and Abowd (1999) defined context as "any information that can be used to characterize the situation of entities". From this understanding, location is a kind of context. Most of the outdoor LBS systems employ GPS for positioning. Unfortunately, these systems can only be used outside of buildings because the employed radio signals cannot penetrate solid walls. For positioning in indoor environment, additional installations (e.g., WLAN, sensor networks) are required.
Additionally, "There is more to context than location" (Schmidt et al. 1999). In order to gather other context data, different sensors (such as temperature sensors, noise sensors, etc.) are employed in LBS systems. Usually, the data gathered from different sensors has to be aggregated and analyzed to deduce some high level context information.

Currently, the abundance of technology in an environment has given place to the notion of "Smart Environments (SmE)", which refers to "environments that sense, perceive, interpret, project, react to and anticipate the events of interest and offer services to users accordingly" (Augusto and Aghajan 2009). Smart environment can help to gather real-time context information. And also by constantly observing the environment, and accumulating historical data, smart environment can deduce high level context information. To sum up, smart environment can help to enable context-awareness in LBS.

2.2 LBS in Web 2.0

Web 2.0 is a hot topic in the field of ICT (Information and Communication Technologies). It is characterized as facilitating communication, information sharing, interoperability, user-centered design and collaboration on the World Wide Web (Wikipedia 2009). Ovaska and Leino (2008) provided a survey of related issues in Web 2.0.

The philosophy of Web 2.0 is "Web-as-participation-platform". Web 2.0 allows users to do more than just retrieve information. Users are also encouraged to contribute their data. These "various kinds of media content" that "are produced by end users" are User Generated Content. Currently, with the impetus of Web 2.0 applications, such as Facebook, Flickr, and Twitter, huge amounts of UGC are being created every hour, even every second. Additionally, with the ubiquity of GPS and easily access of web maps such as Google Earth, Google Map, Yahoo! Map, and Microsoft Live Map, more and more UGCs are georeferenced/geotagged.

The highly available UGC brings some challenges: 1) the sheer volume of UGC makes it more and more difficult for users in general to find and access relevant information; 2) how can UGC be used to generate value? Recommendation system is one of the most promising solutions for these challenges. It is usually used in E-commerce. Some examples about this are “Customers who bought this item also bought” and “Best seller lists” at the Amazon website, “Most viewed” at YouTube, etc. In daily life, when people make decisions on different options which their have no prior experience, they always seek advice from others who have such experience (word-of-mouth). UGC reflects users’ experience, and can be viewed as “wisdom of the crowd”. From these aspects, users (especially other users) can benefit from these kinds of collective intelligence based recommendations.

The combination of LBS and Web 2.0 is a trend. Web 2.0 can enhance LBS with rich and real-time user generated content, which can be used to provide better services in LBS. There are some researches on exploring the idea of incorporating content created by users into LBS systems (Espinoza et al. 2001, Burrell et al. 2002). Some of the researchers used recommendation technology to make UGC useful, for example, event recommendations (de Spindler et al. 2006), tourist destination recommendations (Hinze and Junmanee 2006), restaurant recommendations (Dunlop et al. 2004), gas recommendations (Woerndl et al. 2009), etc.

2.3 Mobile navigation

Mobile navigation is one of the most important LBS applications. When coming to a new place, we always need some wayfinding support. Mobile navigation services are designed to provide wayfinding guidance in an unfamiliar environment.

According to Downs and Stea (1977), navigation (wayfinding) includes four processes: orientation (determining one’s position), planning the route, keeping on the right track, discovering the destination. The last two processes can be combined together as moving from origin to destination. They correspondingly relate to three modules in wayfinding services: positioning, route calculation, and route communication.

The positioning module tries to determinate the position of the user. For outdoor navigation, GPS are always employed for positioning. For positioning in an indoor environment, additional installations (e.g., WiFi or sensor networks) are required. The route calculation module focuses on computing the “best” route from the origin to destination. However, there are different kinds of “best” routes: fastest, shortest, least traffic, most scenic, etc. As a result, when calculating route for users, users’ context should be considered. Another important aspect of mobile navigation is how to communicate route information efficiently (Gartner & Uhlirz 2005). A good route presentation form (such as map, textual and verbal instruction, signs, etc) will enable way finders to easily find their way with little cognitive load.

There are several survey papers focusing on mobile navigation systems, such as Baus J, Cheverst K & Kray C (2005), Krueger et al. (2007), Raper et al. (2007), and Huang & Gartner (2009). These surveys concluded that mobile pedestrian navigation systems are still on an early development stage. Currently, mobile pedestrian navigation systems always employ GPS (outdoor) or radio signal (indoor), such as WiFi, Bluetooth, RFID, etc., for positioning, which may suffer from the problem of poor reliability and stability. How to provide reliable and stable position information in a complex and changing environment is a very challenging task. Sensor fusion may be an option for this question. For route calculation, shortest and fastest routes are also employed in current mobile pedestrian navigation systems. Most of the researches in route communication are focusing on evaluating the suitability and efficiency of varied presentation forms for mobile pedestrian navigation.

3. Smart environment

Smart Environment (SmE) can be viewed as "*a physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives and connected through a continuous network*" (Weiser 1991). From this perspective, a smart environment should at least include different kinds of sensors and a communication infrastructure (wireless or wired) interconnecting these sensors. Based on this understanding, we established a simple smart environment with a positioning module, which uses sensors to provide adequate positioning information, and a wireless infrastructure module, which interconnects mobile clients (such as cell phones and PDAs) and devices installed in the environment (such as servers, sensors, etc.). This section will focus on these two modules.

3.1 indoor positioning

For outdoor LBS, satellite positioning, such as GPS, provides comfortable, sufficient accuracy, and from the end user's point of view, economical positioning (Roth 2004). As a result, most of the outdoor navigation systems employ GPS for positioning. Unfortunately, GPS cannot be used in the indoor environment because the employed radio signals cannot penetrate solid walls. For positioning in an indoor environment, additional installations (e.g., WiFi or sensor networks) are required.

There exist numerous different positioning techniques that vary greatly in terms of accuracy, costs and used technology. Huang and Gartner (2009) provided a survey on different positioning techniques. All of them have advantages and disadvantages. When selecting a positioning approach, several questions have to be considered:

1) Which positioning signal is suitable for the application? Infrared, Ultrasound, Radio, or Visual Light?

2)Which type of sensors is suitable for the application? Infrared, Ultrasound, WLAN

(WiFi, IEEE 802.11), Bluetooth, Zigbee, UWB or RFID?

3)Which signal metric is suitable for the application? Cell of Origin (CoO), Received Signal Strength (RSS), Angle of Arrival (AOA), Time of Arrival (TOA), or Time Difference of Arrival (TDOA)?

4）Which positioning algorithm is suitable for the application? Proximity, Triangulation (lateration and angulation), or Location fingerprinting?

5）Which operation mode is suitable for the application? Active client or Passive client?

6)Which position calculation mode is suitable for the application? Server-side or Client-side?

7) Is it cost-effective?

After comparing different positioning techniques, a *Bluetooth-based beacon positioning solution* is adopted, which uses Cell of Original (CoO) as signal metric, proximity as positioning algorithm, and adopts passive position calculation. In the smart environment, we use BlueLon BodyTag BT-002 (Bluelon 2009) as Bluetooth beacon because we can adjust the range of BodyTag BT-002 by changing its transmit power. Bluetooth beacons are placed in different places actively broadcasting their unique IDs. Mobile devices passively receive the broadcast message when they are within the range of a beacon. After receiving a beacon ID, mobile devices look up the current position from a mapping table. This mapping table can be cached in the mobile devices or accessed from a server.

After choosing the positioning technique, the sensor placement which tries to optimize the placement to balance the signal coverage and development cost has to be considered. Different applications may have different coverage requirements. Most practical applications do not need complete coverage at all times. As a result, the optimized placement is application-dependent. There are different methods which handle the arrangement of digital signs such as experimental approaches and some probabilistic methods like Monte-Carlo localization. Most of them have tried to cover the entire indoor environment to avoid disconnection between the users and positioning sensors (Hahnel et al. 2004). For indoor navigation, complete coverage is not necessary. As decision points (areas where the navigator must make a wayfinding decision, such as whether to continue along the current route or to change direction) are essential for wayfinding (Golledge 1999), we adopt a simple placement solution: beacons are placed at every decision point. The methods suggested in Brunner-Friedrich and Radoczky (2005) are used to derive the positions of decision points. And then, in order to avoid overlapping, the range for every beacon is adjusted.

3.2 infrastructure

The wireless infrastructure module interconnects mobile clients and devices installed in the environment. To establish a wireless infrastructure, several technological solutions are possible: IrDA, WiFi, Bluetooth, UWB, ZigBee, etc. They differ in operating frequency, range, data transfer rate, connection type, etc. Table 1 shows a general overview of the different techniques regards their operating range, data transfer rate and used carrier frequency, etc.

Table 1: overview of connection possibility

Name	Frequency Spectrum	Data rate (bps)	Range (m)	Connection Type, Direction	Application
Bluetooth	2.4-2.485 GHz	1M-3M	1-10-100	Multipoint, Omni-directional	Cellphone, PDA
UWB	3.1-10.6 GHz	70M-1G	10	Multipoint, Omni-directional	Family multimedia
ZigBee	2.4-2.485 GHz	250K	50	Multipoint, Omni-directional	Sensor network,
IrDA	Infrared	115K-4M	1-3	Point-to-Point, Line of sight	Cellphone, PDA
Wireless LAN	2.4-2.485 GHz, 5GHz	11M-54M	300	Multipoint, Omni-directional	Mobile devices, Internet services

For a specific application, data rate, range and connection type maybe the most important criteria. After carefully analyzing and comparing different technologies, we establish a wireless infrastructure based on WiFi technology because of its highly availability, its high data rate, and its wide coverage range.

A central server is also introduced to the smart environment. It is responsible for providing indoor navigation services, gathering and recording real time messages (such as users' moving track, user generated content, etc).

Figure 1 depicts the layout of the proposed smart environment. This smart environment is very simple, but it is enough as a testbed for effectively supporting the entire indoor navigation process, including indoor positioning, route selection, and route presentation. Also this smart environment enables collecting (gathering and accumulating) different user generated content explicitly and implicitly. For other applications, different other sensors such as temperature sensors and noise sensors may be also integrated into the smart environment to facilitate context gathering.

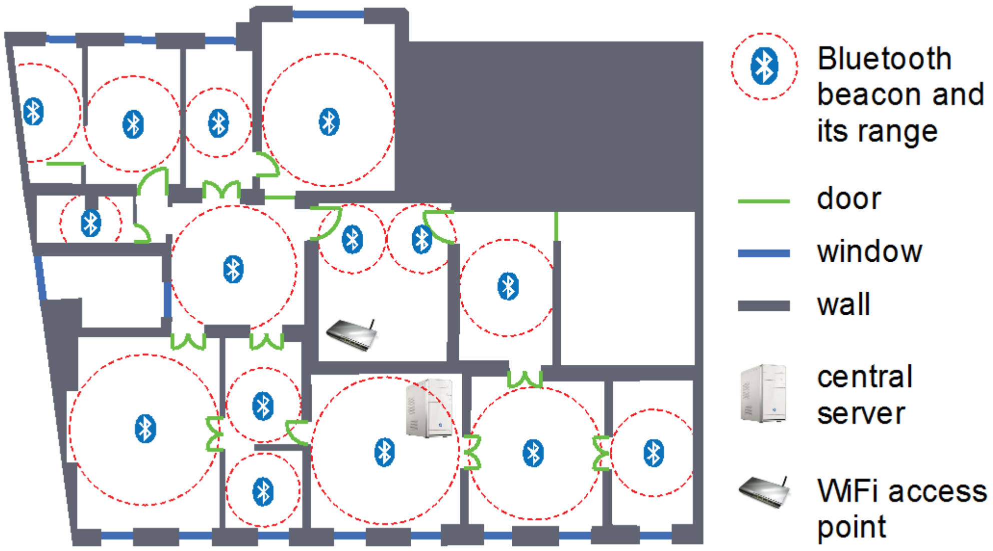

Figure 1. Layout of the proposed smart environment

4. User interaction and annotation

One of the great advantages of ubiquitous systems is the potentiality to directly interact with the environment. This functionality provides a basis for collecting different user generated content explicitly and implicitly. The proposed smart environment also supports this functionality.

We design a mobile navigation system to provide navigation guidance in this smart environment. During navigation in the smart environment, users receive wayfinding support which guides them to their destination. Currently, we calculate the shortest (distance) route for users according to the current context (mainly “traffic information”). For example, sometimes, some of the corridors are blocked, as a result, the system avoids guiding the user (navigator) crossing those corridors. As maps are proved to be one of the most useful presentation forms which can communicate route information efficiently, we employ schematic maps as the route presentation form. In order to enable users (navigators) to easily find their way with little cognitive load, we derive landmarks and visualize them in the route map. During navigation, if the user strays from the calculated route, the system will warn the user, and ask the user to go back. In order to protect their privacy, users can use the system anonymously.

However, the proposed mobile navigation system allows users to do more than just receive navigation guidance. They are also encouraged to contribute their own user generated content (e.g., ratings, comments, feedbacks, etc.) while using the smart environment.

4.1 User Generated Content

Currently, two kinds of user interaction and annotation are supported in the proposed smart environment: explicitly and implicitly.

Explicitly means that users have to interact with the system (e.g., providing information) actively, for example, giving ratings, writing comments, adding feedbacks, etc. During navigation in the smart environment, users are encouraged to annotate their personal preferences, comments or experiences to this environment. We adopt the “note category” described by Burrell et al. (2002) to classify different kinds of user generated content: factual, opinion/advice, snapshot, humor, and question/answer. As the smart environment is georeferenced by the Bluetooth beacons (every beacon has an address), the user generated content posted by users can be viewed as *user generated georeferenced content*. Currently, the proposed system only supports text UGC. Multimedia UGC will be supported in the next version of the system. In default case, user generated content is dedicated for everyone (public) and has a permanent availability. Users can also specify the target person and the duration of it, for instance, this user generated content is only showed to Mary, and is only available on April Fools’ Day. In order to protect the privacy, users can post their comments anonymously.

Currently, computers are hard to measure and process text information automatically. As a result, we also encourage users to give ratings. For navigation, the route users need to follow can be viewed as route segments connected by different decision points (areas). Users can give ratings for these two elements: decision point and route segment. In the smart environment, every decision point is georeferenced by a Bluetooth beacon, while every route segment is georeferenced by two Bluetooth beacons (two decision points). At every decision point, users can give rating to identify the *level of complexity* (cost of effort) of making right decision (choosing the right road to follow) at this point. The rating value scales from 1 to 5. The more complexity, the higher the rating value. Rating for a route segment reflects users’ *level of interest* for the route segment. For example, users may like route segment SA very much because of the nice view along it. The rating value scales from 1 to 5. The more the interest, the lower the rating value. When submitting the UGC, users only need to write their comments or give the rating values. The smart environment figures out the related positions from the positioning module (section 3.1), and stores the comments or ratings to the central server via the wireless infrastructure module (section 3.2).

These kinds of information created by users simultaneously represent their navigation experiences in the environment, and can be used to generate value (such as recommendations) for other users (Ovaska and Leino 2008). Also, Espinoza et al. (2001) and Burrell et al. (2002) noted that the “social, expressive, and subversive,” qualities of content created by users may be more interesting than content created by administrators which “tends to be ‘serious’ and ‘utility oriented’”.

Implicit interaction and annotation means that users don’t have to do anything other than using the system (Ovaska and Leino 2008). The system constantly tracks users’ actions and behaviors to detect their preference. During navigation in the smart environment, a user’s current position is recorded by the system every second, such as (userA, 2009-6-20 15:23:40, placeA), (userA, 2009-6-20 15:23:41, placeB). This sequential position information forms the user’s moving track during her/his current navigation. In order to protect her/his privacy, the system uses a pseudo name (e.g., randomly generated by computer) to represent the user.

4.2 Motivation and Data Quality of UGC

One important issue related to users' interaction and annotation is what motivates users to contribute. Kang et al. (2008) developed a system for sharing tourism experience and concluded that "Tourists not only want to see and feel" the environment, "but they also want to learn more about its history (other people's experiences) and make an impact on its future (contributing their own experiences)." Burrell et al. (2002) noted that users are motivated to contribute "when they thought themselves experts, when there is a pay off or when it is very easy to do"; Users "also seemed to have benefited from feelings of altruism and expertise resulting from contributing notes to help out others". Nov (2007) made a survey on people who contribute to Wikipedia, and identified some main factors which motivate people to contribute, such as fun, ideology, values, understanding, enhancements, protective, career, and social. We also propose that the motivation to contribute also includes the improvement of services we receive and the possibility of reaching much more relevant information (e.g., systems can learn our preferences from our user generated content).

Data quality is also a big problem of user generated content in Web 2.0. While many notes were correct, relevant, interesting, and useful, others were not. It is difficult to automatically determine whether the content users post are of high quality. As Burrell et al. (2002) suggested, allowing users to vote on the usefulness of contents themselves is a possible solution to this problem. We adopt this suggestion. However, further research has to be done on this issue.

5. Collective intelligence based route selection

As mentioned in section 2.2, recommendation system can help to make user generated content useful. It is also a good approach to show the "wisdom of the crowds". Some examples about recommendation are "Customers who bought this item also bought…" and "Best seller lists" in the Amazon website, "Most viewed" and "Most discussed" in YouTube, "Most popular tags" in Flickr, "The most popular bookmarks" in Del.icio.us, etc. These kinds of "collective intelligence"-based recommendations can be very useful for the users of these services. Also these kinds of recommendation methods can help to achieve the center goal of Web 2.0 services: *the more they are used, the better they get* (Musser et al. 2006).

In this section, we focus on applying recommendation technology to generate value from user generated content for mobile navigation.

5.1 data modeling

As described in section 4.1, user interaction and annotation are supported in the proposed smart environment explicitly and implicitly. For explicit interaction and annotation, we encourage users to give ratings for different decision points (level of complexity) and different route segments (level of interest).

Rating for a decision point reflects the *level of complexity* (cost of effort) of making right decision (choosing the right road to follow) at this point It is always involved with *a pair of connected route segments* (the route segment which the user just visited, and the route segment which the user is going to visit). The current decision point is the junction of these two route segments. As a result, rating for a decision point is modeled as a *4-tuple (previous, current, next, value)* containing the previous decision point, the current point, the next decision point, and a rating value.

Rating for a route segment reflects users' *level of interest* for the route segment. It is a *3-tuple (start, end, value)* containing the start and end decision point of the route segment, and a rating value. For example, a user like the route segment SA very much, and give the following rating: (S, A, 1).

We can also use the data collected in the implicit interaction and annotation. For every moving track, some statistical data about the current navigation can be obtained: *moving duration at every decision point*, and *error point*. Similar to ratings for decision points, these two parameters may also reflect the complexity of decision points. For example, if the user stays more time at decision point A (duration is too long), and doesn't do anything (e.g, posting comments, giving ratings), it is reasonable to consider that the user has troubles in choosing which way to follow, as a result, the complexity of this decision point is high. Durations are recorded as *4-tuple (previous, current, next, value)* containing the previous decision point, the current point, the next decision point, and a duration (measured by second). Currently, we assume that, most of the time users spend during navigation is either on walking or on posting comments/ratings. Durations have to be *standardized* to a cost scale (similar to ratings for decision points). In order to make this standardization, some field experiments should be carried out to find out some referenced durations.

The proposed navigation system continually checks whether the user is on the right route to her/his destination. If the user strays from the calculated route, the system will warn the user, and ask the user to go back. We adopt the following method to identify the error points where a user makes wrong decisions: if the user moves in the sequence of A-B-A-C, point A will be the error point for this user because s/he made some wrong decision at A. For error point A, the following rating (previous, A, C, 5) is assigned.

5.2 collective intelligence based route selection

In this section, we focus on the issue of how these ratings can be used to generate value for mobile navigation.

Inspired by the "most popular (viewed, discussed)…" like recommendations, we design several algorithms to illustrate *how our mobile navigation service can benefit from the user generated* *content* (ratings). We name these algorithms as "collective intelligence"-based algorithm because they are based on users' collective intelligence. These algorithms use UGC (collective intelligence) to calculate different routes, such as route with minimal route segment rating (the nicest route), the least complex route, the most popular route, and the optimal route. As a result, we can provide "collective intelligence"-based navigation service.

The nicest route: route with minimal route segment rating

The goal of this algorithm is to compute the route with minimal route segment rating between origin and destination. As described in section 3.1, rating for a route segment reflects users' level of interest on the route segment. The route with minimal route segment rating can be viewed as "the nicest route".

Generally, graphs are a standard data structure for representing road and transportation networks. A graph G consists of a set of vertices V and edges $E \subset V \times V$ connecting the vertices. In a route network, every intersection is represented as a vertex, and each road (route segment) is represented as an edge (Duckham and Kulik 2003). Edges can be assigned with weights (cost), for example, Euclidean distance of this edge, travel time, or travel fares. For our case, G is an Undirected Graph. The shortest (cost) route from origin A to destination B can be viewed as the path in graph G with least cost. *Dijkstra's algorithm* can be used to solve this problem (Dijkstra 1959). The basic idea of Dijkstra's algorithm is to assign some initial distance values and try to improve them step-by-step.

For calculating the route with minimal route segment rating from origin to destination, the rating for each route segment (road) is assigned to its corresponding edge in graph G. The rating for route segment (s,e) based on collective intelligence is calculated as:

$$R\_E(s,e) = \begin{cases} 3, \ if \ no \ ratings, use \ default \ value \\ \dfrac{\sum R_i(s,e)}{n}, \ others \end{cases} \qquad (f-1)$$

Where $R_i(s,e)$ is user *i*'s rating for route segment (s,e), and n is the total number of ratings for (s,e).

The least complex route

The goal of this algorithm is to compute route with least complexity between origin and destination. As described in section 3.1, ratings for decision points reflect the complexity of each decision point. A route from origin to destination includes a series of decision points. As a result, the least complex route can be viewed as route with the minimal ratings for decision points.

Ratings for decision points are modeled as 4-tuple (previous, current, next, value). The rating value can be viewed as a cost assigning for a pair of connected route segments. For example, rating (S, A, B, 4) in Figure 2 can be viewed as the cost of negotiating the path from S to B through decision point A. The cost of navigating from node $previous$ to node $next$ through node $current$ based on collective intelligence is calculated as:

$$R\_DP(previous, current, next) = \begin{cases} 3, \ if \ no \ ratings, \ use \ default \ value \\ \dfrac{\sum R_i(previous, current, next)}{n}, \ others \end{cases} \qquad (f-2)$$

Where $R_i(previous, current, next)$ is user i's rating for route segments $(previous, current)$ to $(current, next)$, and n is the total number of ratings on this pair of route segments.

In order to solve this problem, we use the *restricted pseudo-dual graph* proposed by Winter (2002). The pseudo-dual graph D of the original graph G is defined as: 1) each edge e_i of G is represented as a node v_i in D, 2) each pair of connected edges (e_i, e_j) in G is represented as edge ε which connects nodes v_i and v_j in D. For example, Figure 5 depicts the pseudo-dual graph of the graph in Figure 2. Note that the pseudo-dual graph D is a Directed Graph.

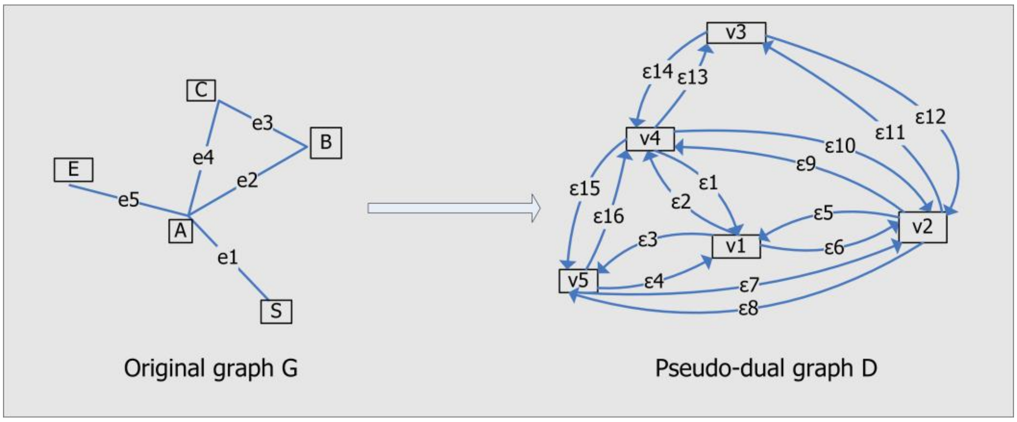

Figure 5. Original graph and its pseudo-dual graph

Winter (2002) proved that the shortest (cost) route (single-source/single target) problem in original graph G can be transformed to multi-sources/multi-targets problem in D. He reduced this problem to a single-source/single-target problem by adding a virtual source node and a virtual target node to D. In this new graph D', the shortest route can be computed by using the classical Dijkstra's algorithm.

For our case, the "collective intelligence"-based costs for decision points (in formula 2) can be easily assigned to edges in the pseudo-dual graph D:

$$Cost(v1, v2) = R\_DP(previous, current, next) \qquad (f-3)$$

Where $v1$ in D is the edge $(previous, current)$ in G, and $v2$ in D is the edge $(current, next)$ in G.

It is also interesting to note that these ratings for decision points can also be used to *quantitatively evaluate* the mobile navigation service. For example, for a long period of time (e.g., two months), if the average rating of a specific decision point is too high (e.g., >4.5), it is reasonable to consider that the navigation support provided at this decision point is very poor. As a result, information services provided by the navigation and smart environment at that place should be enhanced. This can be viewed as a new method for quantitative evaluation of software in software engineering.

The most popular route

The goal of this algorithm is to compute the most popular route between origin and destination. The most popular route is the *optimal trade-off* between the route with minimal route segment rating ("the nicest route" in section 4.1) and the route with least complexity (section 4.2).

In order to calculate the most popular route, we assign an optimum cost to each decision point, which depends on both the ratings for route segments, and ratings for decision points. This optimum cost is given by:

$$\begin{aligned} &R\_DP_{popular}(previous, current, next) = \\ &\quad \lambda_d \cdot R\_DP(previous, current, next) + (1-\lambda_d) \cdot R\_E(current, next) \qquad (f-4) \end{aligned}$$

Where λ_d determines the weight of the impact for the ratings for decision points, $R\_E(current, next)$ and $R\_DP(previous, current, next)$ can be calculated by formula f-1 and f-2.

Similar to the algorithm in section 4.2, the most popular route can be calculated by the classical Dijkstra's algorithm based on the pseudo-dual graph.

In order to achieve better result, λ_d has to be calibrated. λ_d may be different for different environments. The method proposed by Haque et al. (2007) can be used to find out the optimum value for λ_d. It compares the results for different λ_d values with those obtained from the separate algorithms (e.g., route with minimal route segment rating in section 4.1 and route with least complexity in section 4.2).

The optimal route

Compared to the shortest (distance) route, the most popular route in section 4.3 may lead to longer distance between origin and destination. As a result, we calculate the optimal route, which takes ratings for route segments, ratings for decision points, and the Euclidean length of route segments into account. In order to calculate the optimal route, we assign an optimum cost to each decision point, which depends on the three parameters mentioned above. This optimum cost is given by:

$$R\_DP_{optimal}(previous, current, next) = \\ (1-\lambda_o)\cdot R\_DP_{popular}(previous, current, next) + \lambda_0 \cdot Dist(current, next) \qquad (f-5)$$

Where λ_o determines the weight of the impact for the Euclidean length of route segments, $Dist(current, next)$ is the Euclidean length of route segments, and $R\_DP_{popular}(previous, current, next)$ can be calculated by formula f-4.

The calculation of the optimal route and the calibration of λ_o can use the same methods as in section 4.3.

5.3 discussions

There are some similar papers focusing on calculating different routes for users. For example, the route with minimal number of turns, the route with minimal angle by Winter (2002); the route with least instruction complexity by Duckham and Kulik (2003); the reliable route which minimizes the number of complex intersections with turn ambiguities by Haque et al. (2007), etc. However, all of the above routes are mainly based on the geometrical characteristics of the road network. The proposed “collective intelligence”-based algorithms are based on all users’ UGC, which r*eflects users' navigation experiences* in the environment. As a result, compared to other route algorithms, our algorithms will provide results which are more suitable to the users.

From the mathematical perspective, the nicest route algorithm uses the cost (weight) of edges to compute the shortest (cost) route. The least complex route algorithm uses the cost (weight) of connected edges pair. Both of the most popular route algorithm and the optimal route algorithm combine cost of edges and cost of connected edges pair. In this paper, we use level of interest and complexity as cost. For some other applications, the cost function may differ, such as travel time, travel expenses, etc. However, the proposed algorithms can be also used to solve this kind of problems.

It is also important to note that the above algorithms of “collective intelligence”-based route calculation make community-at-large recommendations to individual users. It is not especially made for any particular user but all get the same recommendation (Ovaska and Leino 2008). In our daily life, *these kinds of popularity-based recommendations have been proved to be very useful*. However, some of the users may have particular interests. In order to make more relevant recommendations for them, *collaborative filtering* should be introduced. The most known examples of collaborative filtering are Amazon-like “Customers who bought this item also bought” recommendations. Collaborative filtering includes two steps: 1) find out similar users (this step can be viewed as assigning the current user to a group), 2) carry out the “popularity-based recommendations” on this group of users. As a result, the proposed algorithms can also be used in the second step of collaborative filtering.

In this paper, we use indoor navigation as a testbed. However, the proposed algorithms can be also applied to outdoor pedestrian navigation services and car navigation services.

6. Selected context-awareness

Mobile navigation should be context-aware, and adapt to the dynamic changing environment. Before discussing the context-awareness providing by our navigation system, we want to introduce the notion of context used in this paper. We adopt the definition provided by Huang and Gartner (2009): “1) Something is context because it is used for adapting the interaction between the human and the current system. 2) Activity is central to context. 3) Context differs in each occasion of the activity.”

Based on the above two modules and the smart environment, our navigation system provides the following context-aware adaptations.

6.1 software architecture

Software architecture is very important when designing navigation systems. While not being directly apparent to the user, it has a serious impact on the system’s extensibility and adaptability (Baus et al. 2005).

For software architecture, we can classify navigation systems into services-side (connecting) and client-side (local caching) solutions according to where the data (spatial data and route instructions) is stored and the calculation (mainly route selection) is executed. These two solutions have different requirements in CPU’s processing performance, memory capability, battery consumption, network availability, etc. In fact, it is not suitable to simply assign the calculation and data to the server side or the client side. In order to have an extensible and adaptable system, the decisions on where the calculation is executed and data is stored should depend on the current context, such as mobile devices’ processing performance, memory level, power (battery) level, network availability etc.

In our navigation system, we provide a context-aware adaptation for software architecture. Some of the context parameters we used are: mobile devices' processing performance, memory level, power (battery) level, and network availability. Where to execute the calculation and where to store the data are adapted based on these context parameters. We develop an empirical function for determining the distributions of data (spatial data and route instructions) storing and calculation (route selection) executing. This context-aware adaptation will start (by invoking the empirical function) when users enter the smart environment.

Figure 5 depicts the server-side solution. The basic steps are:

① The Bluetooth beacon constantly and actively broadcasts its unique ID.

② When the mobile device (PDA or smart phone, held by the user) is within the range of the Bluetooth beacon placing at the entrance, it receives a unique ID. The user types his/her destination (such as a member of our group). Then the mobile device forwards this message (the unique ID, the destination, user profile, device profile) to the central server.

③ After receiving the message, the central server looks up the associated position information in the Mapping table, and calculates the route for the given origin and destination according to the current context, and then forwards the route guidance (maps or information in other communication forms) to the mobile device. If the destination is a person, the central server requests the person for his or her current position. The central server may connect to the Internet to obtain some context parameters.

④ The user walks along the suggested path.

⑤ When the mobile device receives a new beacon ID, it forwards the ID to the central server.

⑥ The central server checks the user's current position and verifies if he or she is still along the right route. If the user lost the suggested route, a new path is calculated and sent to the mobile device automatically. If the user is on the right route, a new guidance corresponding to the current position is forwarded to the mobile device.

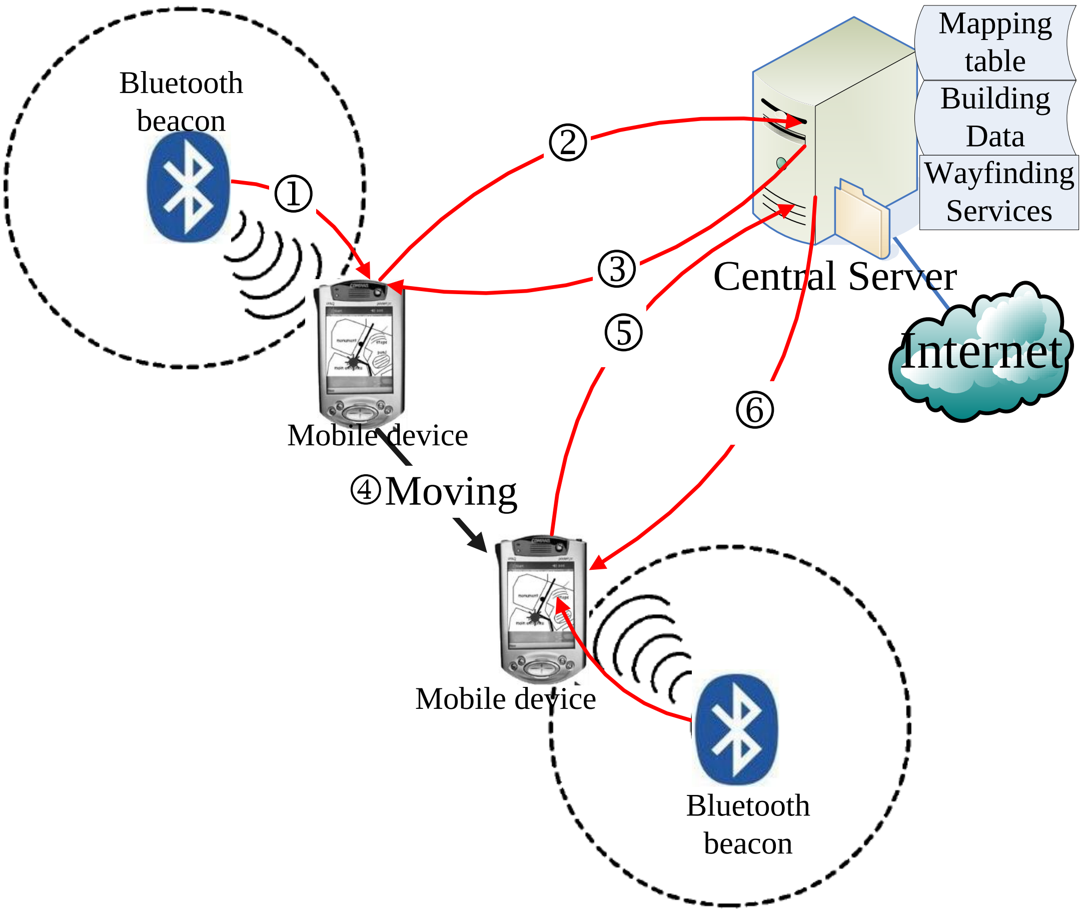

Figure 5: Work flow of Server-side solution

The navigation services in the users’ mobile devices can also operate in the client side. Figure 6 depicts the work flow of client-side solution.

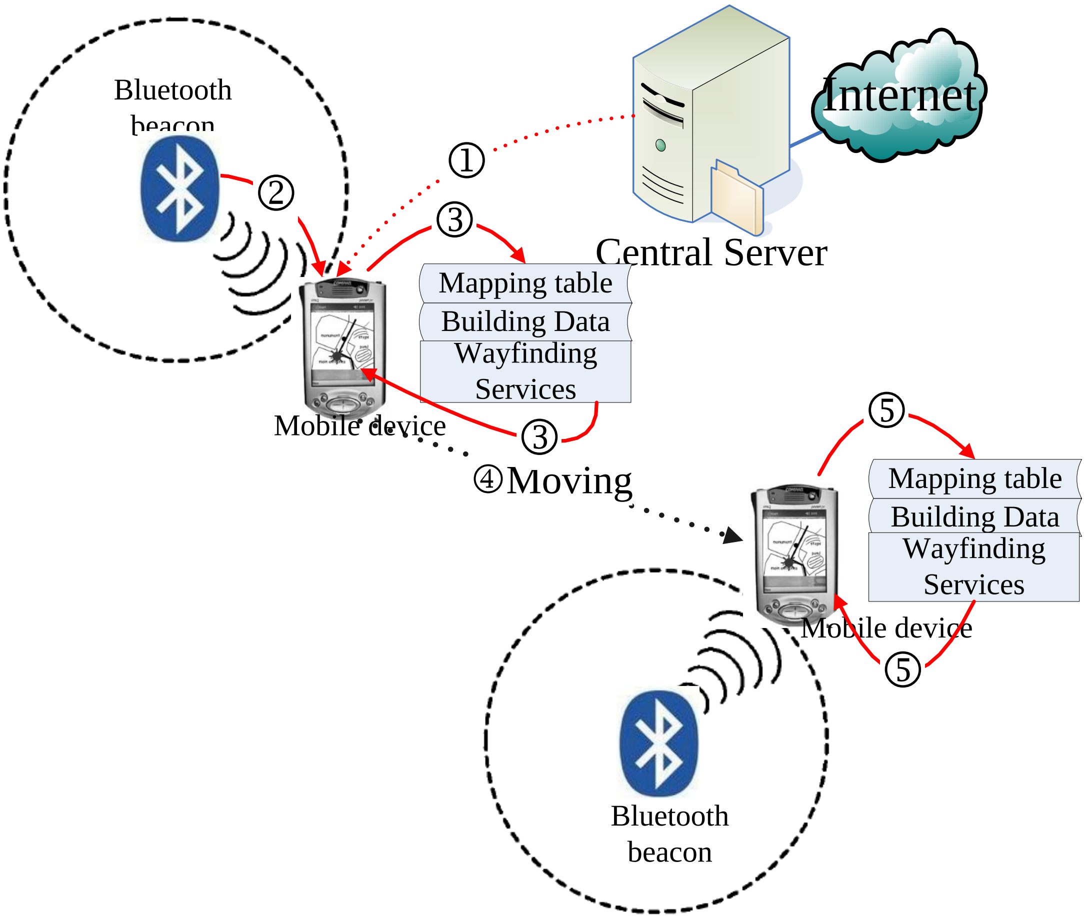

Figure 6: Work flow of Client-side solution

① The mapping table, building data, and other related information (such as context parameters) are downloaded from a server and cached on the mobile device in advance. Also wayfinding services are installed on the mobile device.

② The Bluetooth beacon constantly and actively broadcasts its unique ID.

③ When the mobile device (PDA or smart phone, held by the user) is within the range of the Bluetooth beacon placing at the entrance, it receives a unique ID. The user types his/her destination. Then the mobile device looks up the associated position information in the Mapping table, and calculate the route for the given origin and destination according to the current context, and then presents the route guidance (maps or information in other communication forms) to the user.

④ The user walks along the path.

⑤ When the mobile device receives a new beacon ID, it checks the user's current position and verifies if he or she is still along the right route. If the user lost the suggested route, a new path is calculated and presented to the user automatically. If the user is on the right route, a new guidance corresponding to the current position is presented to the user.

6.2 destination selection

At this moment, most of the navigation systems always guide users to a destination, which is always a place. However, for navigation, especially indoor navigation, users' destination may also be a person. We provide this function in our indoor navigation system. Usually, people don't always stay in one place (for example, at their desks in the office), e.g., they may move to another room for a meeting. Based on the tracking module, we can get the current position of the target person from the smart environment, and guide the user to the target person's current position. If the target person's current position can not be provided by the smart environment (for some privacy reason), the indoor navigation system will guide the user to the usual place (for example office).

7. Conclusions and future work

Technology available today is rich. With the rapid advances in enabling technologies for ubiquitous computing, more and more active or passive devices/sensors are augmented in the physical environment, our environment has become smarter. Also, currently the ICT (Information and Communication Technologies) society has fully adopted the concept of "Web-as-participation-platform" in Web 2.0. As a result, *the combination of Location Based Services, smart environment and Web 2.0 is a trend.* This paper addressed this concern. In this paper, a smart environment with a positioning module and a wireless communication module was set up to support users' wayfinding, facilitate users' interaction and annotation with the smart environment, and collect user generated content. In order to illustrate the benefits of introducing smart environment and Web 2.0 into mobile navigation services, this paper designed several "collective intelligence"-based route calculation algorithms to provide smart wayfinding support to users, such as "the nicest route", "the least complex route", "the most popular route", and "the optimal route".

From the above discussions, the following conclusions can be drawn: *smart environment can help to collect user generated content explicitly and implicitly (collective intelligence) during navigation, such as ratings, comments, feedbacks, moving tracks, durations at decision points, etc. By enabling UGC, mobile navigation services can provide users with a new experience and smart wayfinding support (such as, "collective intelligence"-based route recommendations).*

According to Svensson et al. (2005), the central idea of Web 2.0 services is that: the more they are used, the better they get. As a result, our next step is to recruit more people to use our "collective intelligence"-based navigation service. Furthermore, in the other part of our UCPNavi project, some researches are focusing on finding patterns of human wayfinding behaviors (Millonig and Gartner 2008). These patterns can help to divide users into different groups, and then be used for collaborative filtering. We will combine the result to provide more relevant recommendations.