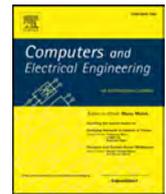


Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A location-based services and Google maps-based information master system for tour guiding[☆]

Sheng-Yuan Yang^{a,*}, Chun-Liang Hsu^b

^a Department of Information and Communication, St. John's University, 499, Sec. 4, Tam-King Rd., Tam-Shuei, New Taipei City 25135, Taiwan

^b Department of Electrical Engineering, St. John's University, 499, Sec. 4, Tam-King Rd., Tam-Shuei, New Taipei City 25135, Taiwan

ARTICLE INFO

Article history:

Received 8 May 2015

Revised 19 November 2015

Accepted 20 November 2015

Available online xxx

Keywords:

Google maps

GPS

Location-based services

Tour guiding systems

Multi-agent systems

ABSTRACT

This paper aims to develop a location-based services supported Dr.What-Info system, i.e. a master multi-agent system on what the information is, using Google maps and an image recognition technology as a tourism information provider and as a route planner for tourists. Users can have great fun during vacation travels through an easy-to-use interface, integrating smartphone GPS function, a QR/Bar code reader and easy access to a cloud database, to find all of the required web services. In particular, given an archeological site in New Taipei City, Taiwan, for testing purposes, the presented system is demonstrated not only as a provider of information on popular tourist attractions, but also as a high performance GPS navigation device to guide users toward their desired destinations. The complete system developments, displays, and corresponding experiments and comparisons show that the research results demonstrate performance superiority over a number of previous studies.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

The Taiwan Network Information Center (TWNIC), Taiwan, began to conduct an annual survey on the number of Internet users in Taiwan in the year 2002. Fig. 1 presents the numbers of various Internet users between the year 2006 and the year 2015. It is clearly seen that there is a high and steady growth of mobile users, including Wi-Fi LAN and Mobile Net, in Taiwan as a consequence of web service promotions and high quality network infrastructure. More importantly, up to half of Taiwan's population are mobile network users. Hence, it becomes an issue of immediate concern to provide users with a wide variety of mobile web services in a highly efficient and reliable manner, as stated in [1].

A statistical data analysis made by Taiwan Tourism Bureau (TTB) indicates that the number of visitors to Taiwan grew rapidly from 3.4 million in 2005 to 9.9 million in 2014, as illustrated in Fig. 2. Furthermore, as many as 72.6% of the visitors, i.e. 7.2 million, arrived for tourism, as shown in Fig. 3. In recent times, there has been a sharp rise in the demand for tourism information as a consequence of the policy for Chinese backpackers admitted to Taiwan. Accordingly, there is definitely a need to develop alternatives to conventional mobile tour guiding systems, such that an easy-to-use interface on mobile devices is developed which can take input data by a smart phone camera and a QR/Bar code reader; users can then have easy access to tour information of interest in a direct manner. As tabulated in Table 1, a survey conducted by the Nielsen Company at <http://www.nielsen.com>, points out that a high percentage of mobile shoppers take pictures of QR or bar codes when shopping online. This motivates the development of an easy-to-use tour guide system, referred to in [2].

[☆] Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. S. D. Prior.

* Corresponding author. Tel.: +886 2 28013131x 6300; fax +886 28012131.

E-mail addresses: ysy@mail.sju.edu.tw (S.-Y. Yang), liang@mail.sju.edu.tw (C.-L. Hsu).

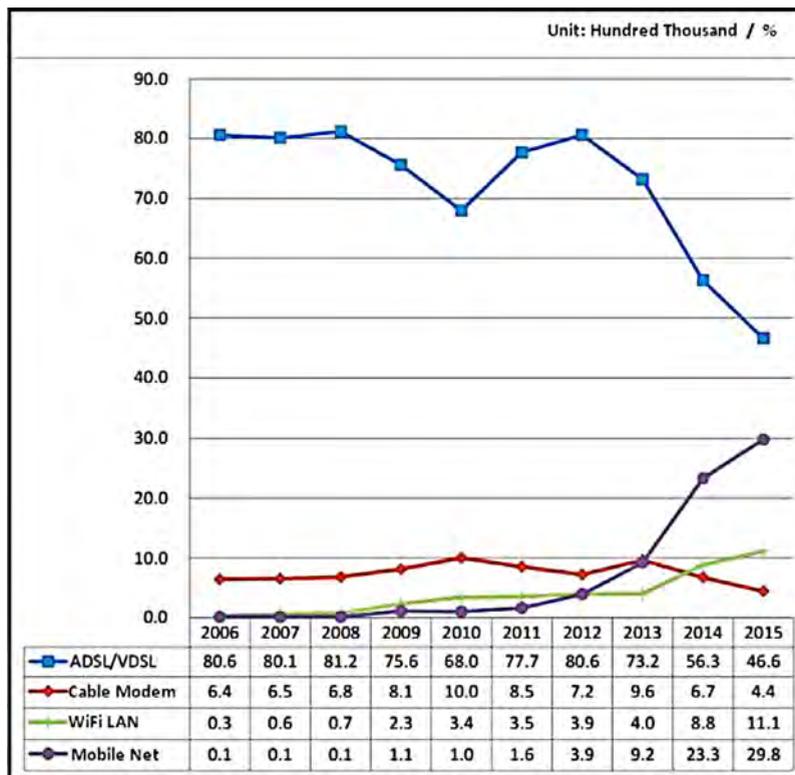


Fig. 1. Statistical data on Internet users in Taiwan (TWNIC, Aug. 2015).

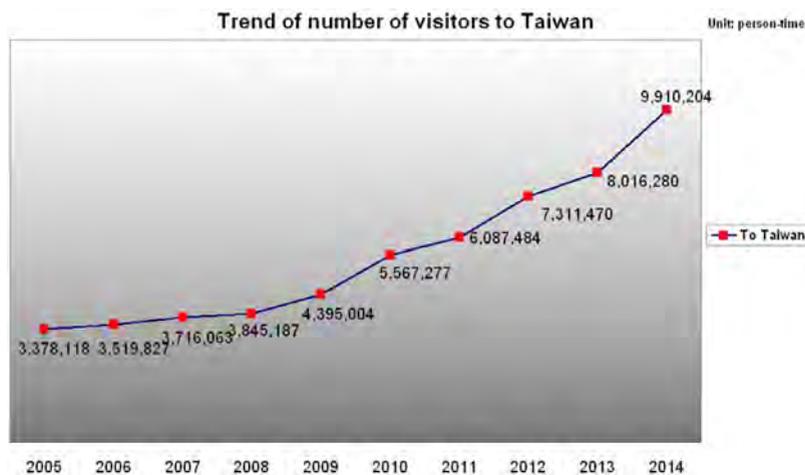


Fig. 2. Total visitors to Taiwan between 2005 and 2014 (TTB, Aug. 2015).

Google maps were firstly announced on Google Blog on Feb. 8, 2005, and the map coverage was extended from the USA, UK and Canada to the entire world on June 20th of the same year. Google Maps, previously referred to as Google Local, is a global online map service, which includes landmarks, path lines, area shapes, vector maps, satellite maps, topographic maps, etc. There have been a great number of Google map-based studies. For instance, an online map application using Google Maps APIs, SQL database and ASP.NET was created by Hu and Dai [3]; a travel guide teaching platform was developed using Google Street View by Kung [4]; an online location-based service was developed using Google Maps for Android mobile by Ibrahim and Mohsen [5]. There is no doubt that the integration of Google Maps into mobile devices will be a mainstream trend in the very near future, and it is the major reason behind this research as well. LBS refers to web service providing integrated services available nearby. Personalized LBS, once integrated with mobile devices, particularly smartphones, is gaining more attention than ever before. Optimized LBS is made accessible so as to meet users' need by integrating GPS navigation, mobile communication techniques, and more. An easy-to-use interface, developed herein, enables tourists to access desirable LBS, as mentioned in [6].

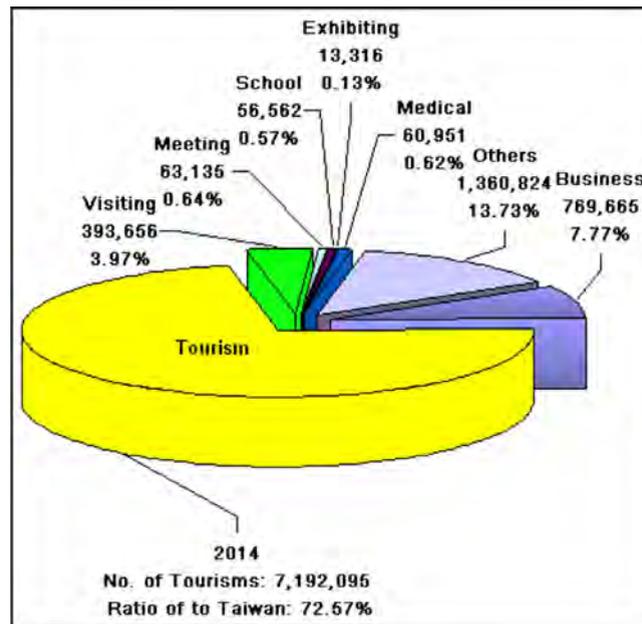


Fig. 3. Pie charts for visitors to Taiwan for various purposes in 2014 (TTB, Aug. 2015).

Table 1

Mobile shopping survey across countries.

Countries	Purposes			
	Bar/QR code scanning	Mobile shopping (%)	NFC/mobile wallet	Mobile bank (%)
South Korea	38%	43	15%	51
China	30%	43	20%	42
Australia	14%	25	4%	40
American	24%	30	3%	38
Russia	12%	17	11%	33
Brazil	22%	17	11%	28
Britain	18%	26	4%	28
Italy	14%	15	3%	22
India	–	7	–	7
Turkey	2%	3	1%	4

Source: Courtesy of Nielsen, Mar. 2013.

In short, Google Maps and an image processing technique are integrated in such a way that the aim of LBS Dr.What-Info system (i.e. a master multi-agent system on what the information is) development is reached for tour guiding. The presented system is designed to take pictures of QR/Bar codes as input data; subsequently, users can have direct and easy access to a cloud database for desired tourism information, and can navigate to specified attractions by smartphone GPS function and Google maps. For demonstration purposes, an archeological site in New Taipei City, Taiwan, is selected as a testing case to validate the superior performance of this work relative to a number of publications referred to herein.

The remainder of this paper is organized as follows: Section 2 introduces the motivation and technologies involved. Section 3 illustrates the system operating architecture, and how it is constructed. Section 4 presents the system displays and evaluations. Section 5 offers some conclusions and future works.

2. Motivation and technologies involved

App store opened following the release of iPhone by Apple in the year 2007. Since then, app is commonly known as any type of application software running on all versions of iPhones and tablets. Soon after, in the same year, Google announced Android, a Linux-based mobile operating system. Since it is an open-source platform, Android appeals to a large number of mobile device makers. Android Market, Google Music and eBookstore were subsequently introduced into the market, and are integrated into Google Play, as announced in March, 2012. Ever since, a multitude of Android apps have been developed at a rapid pace, such that they have become strong competitors against those contained in App store. As indicated in Chen [7], there are substantially two categories of mobile software design: mobile websites and mobile apps. In the former case, applications are executed in a browser. Tremendous progress has been made in mobile websites as a consequence of technology improvement in mobile de-

vices. In essence, a mobile app is a type of application software designed to run on a mobile device. Thus far, the app market is dominated by iOS and Android, while the market share of Microsoft Windows Phone has already been marginalized. Closed source iOS apps are mainly developed in objective-C, while open source Android apps are written in Java and are authorized to mobile app developers. Typical app development tools cover MIT App Inventor for Android at <http://appinventor.mit.edu/>, AppMakr at <http://www.appmakr.com/>, and Buztocuh for both iOS and Android at <http://www.buzztouch.com/>, etc., but the choice of which tool to use is made according to technology support. Accordingly, MIT App Inventor II is selected as the development tool since it features a near field communication (NFC) component and more importantly, supports traditional Chinese characters.

Optical character recognition (OCR) techniques are employed in most character processing cases; typically, it is a four-stage approach involving image processing, document analysis, character recognition and semantic processing. In recent years, tremendous progress has been made in the development of smart phones, particularly high end, giving rise to performance upgrade, e.g. in smartphone cameras and APIs. As suggested in [8], the proposed system is equipped with a number of features, e.g. noise filtration, font pruning, character segmentation, graph thinning, feature recognition, and more. Furthermore, it supports QR and bar code readers for easy use of mobile web services, as presented in Yi [9]. Besides, cloud computing is well defined by Google in Bloomberg Businessweek, 2007, and International Data Corporation clearly indicated that 2010 was the first year of the cloud computing age in Taiwan [10]. Since then, cloud infrastructure, cloud platforms and cloud applications have been developed and become commercially available in a successive way. Moreover, it has been pointed out that apps will become a mainstream product in the highly promising smart phone market. In this context, there have been a great number of cloud-based recognition systems, e.g. those in Cheng [11] and Shih [12]. Apparently, an integration of image recognition technology into smartphone cameras is believed to be a future trend of mobile web services, and has stimulated this study.

LBS are services offered through a mobile phone; they take into account the device's geographical location, and typically provide users with information of interest, either for entertainment or for business purposes. Since LBS are largely dependent on the mobile user's location, the primary objective of the service provider's system is to determine where the user is. There are many techniques to achieve this target. In recent times, numerous published studies on this issue have appeared due to the high demand growth of mobile devices and apps. For instance, user satisfaction with an LBS system was investigated in Wang [13], the business value and use efficiency of an LBS system were reported by Hong [14] and a novel concept of a mobile ad hoc network was proposed for LBS by means of circular data aggregation technique by Ramya and Prasad Babu [15]. One thing all the above have in common is that following an inquiry requested by a mobile device, web services are provided by a backend database through application software developed in PHP, .NET Solutions or Java solutions, and eXtensible Markup Language (XML) or lightweight JavaScript Object Notation (JSON) are employed to format the data transferred. In this manner, information security can be well maintained, and there is no point to redevelop app source codes running on various platforms, which is an advantage of cross platform. In most LBS systems, a small local database is built in such a way that websites of interest can be made accessible offline to users. This paper is presented with the advantages of all the above cited references, such as mentioned in [16], both adaptive features, supported by the underlying learning capability, however, provide yet another level of automation in information recommendation, and deserves more attention, as mentioned in [6].

Ontology refers to a branch of philosophy, dealing with questions concerning what entities exist or can be said to exist, and how such entities can be grouped, related within a hierarchy, and subdivided according to similarities and differences. In addition, ontology is able to provide complete semantic models. Through ontology, the core of a specific discipline can be fully presented, and new knowledge can hence be inferred; it is seen as a powerful tool to develop an information system, with respect to cost and maintenance. As illustrated in Fig. 4, in the proposed system, the content and link among all the concepts can be well depicted for the whole operation of a backend agent system. Referring to the website content at <http://protege.stanford.edu/>, an ontology-based cloud touring information system was accordingly built in St. John's University, Taiwan. The provided

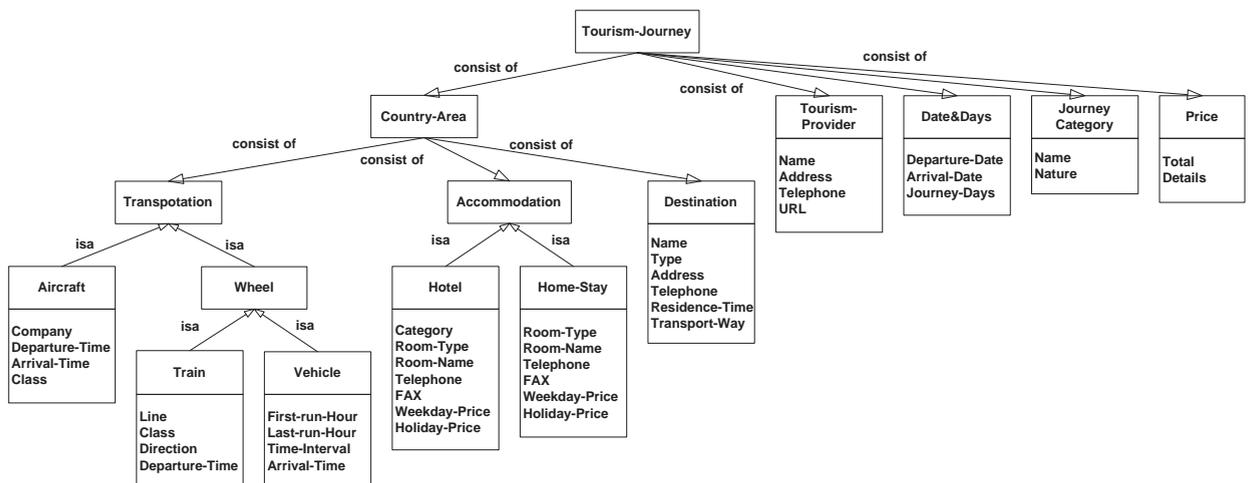


Fig. 4. Tourism ontology.

ontology-based web services cover semantic distance transformation between index terms, hypernyms and hyponyms of term retrieval mapping, conversion between synonyms and antonyms, etc. With the WordNet at <http://wordnet.princeton.edu/> and the Academia Sinica Bilingual Ontological Wordnet at <http://bow.sinica.edu.tw/>, the conversion between English and Chinese languages and connections between distinguishing meanings and relationships of words can be made as a basic operation of this presented system. Furthermore, the Jaccard similarity is employed as a measure of the consistency between ontological concepts. Subsequently, the domain concept can be indexed by the consistency between the concepts of the index terms in WordNet. In the end, with the identification codes Synset_ID given by WordNet, the domain concept can be made accessible for the whole system operation. The Jaccard Similarity, also known as the Jaccard index, is defined as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

and the Jaccard Dissimilarity, also known as the Jaccard distance, is defined as:

$$J_{\delta}(A, B) = 1 - J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \quad (2)$$

In Eq. (1), with the value of $J(A, B)$ in the range of 0 and 1, $J(A, B) = 0$ represents that there is no connection between A and B, and $J(A, B) = 1$ denotes that A is a synonym of B. Alternatively, as defined in Eq. (2), $J_{\delta}(A, B) = 1$, i.e. $J(A, B) = 0$, indicates a case with the maximum Jaccard distance, while $J_{\delta}(A, B) = 0$, i.e. $J(A, B) = 1$, represents a case with the minimum Jaccard distance. Taking OWL/RDF(s) as an instance, all the definitions can be found in “owl:ObjectProperty”, “owl:Functional Property” and “owl:InverseFunctionalProperty”. Hence, ontology can be illustrated with the relevant rules as follows:

```

TRIPLE
(SUBJECT ?P)
(PREDICATE "HTTP://WWW.W3.ORG/1999/02/22-RDF-SYNTAX-NS#TYPE")
(OBJECT "HTTP://WWW.W3.ORG/2002/07/OWL#OBJECTPROPERTY")
TRIPLE(SUBJECT ?S)(PREDICATE ?P)(OBJECT ?O)
=>RESULT : O IS DESCRIPTION FACTOR FOR S

```

As illustrated above, if a predicate P is the above vocabulary type and a subject S has any other concept of an object O , then S is the main concept of this domain. Subsequently, among a set of main concepts, the domain concept is given as:

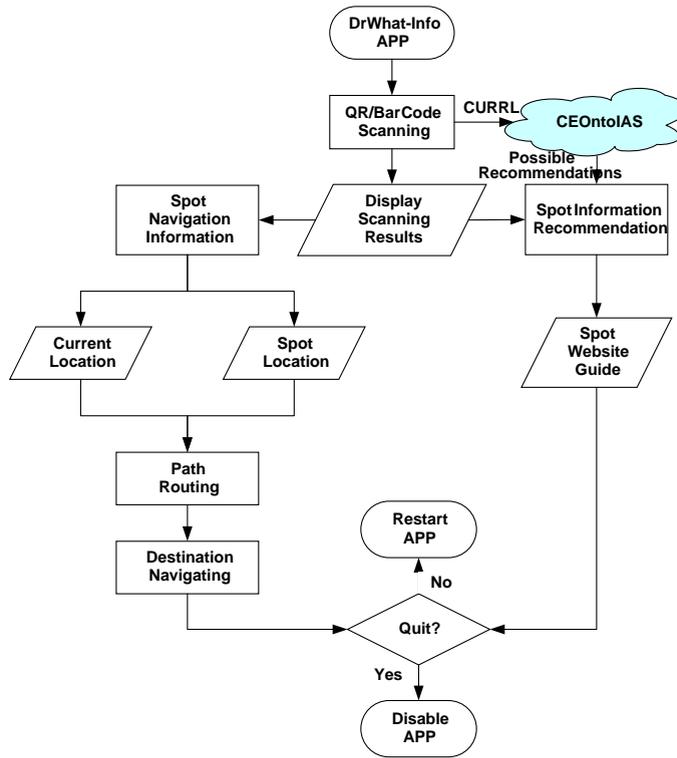
$$DomainConcept = \max_{c \in S(c_1, c_2)} [-\log P(c)] \quad (3)$$

Furthermore, hasURI and hasConsistency are the top 2 ranked attributes in terms of domain significance. By means of the former attribute, domain concept indexing can be made using WordNet synset; the domain concept can be represented by the Jaccard similarity and then stored in the latter attribute. In this manner, the domain concept is completed, and then an ontology-based indexing is made from a data base, according to the Synset_ID given by WordNet. In the end, by means of primitives and an application program interface given at <http://protege.stanford.edu/doc/dev.html>, a number of ontology-based web services were developed for an improved system robustness and efficient integration, e.g. transforming query terms into canonical ontology terms, finding definitions of specific terms in ontology, finding relationships among terms, finding compatible and/or conflicting terms against a specific term, etc., for supporting the whole system operation.

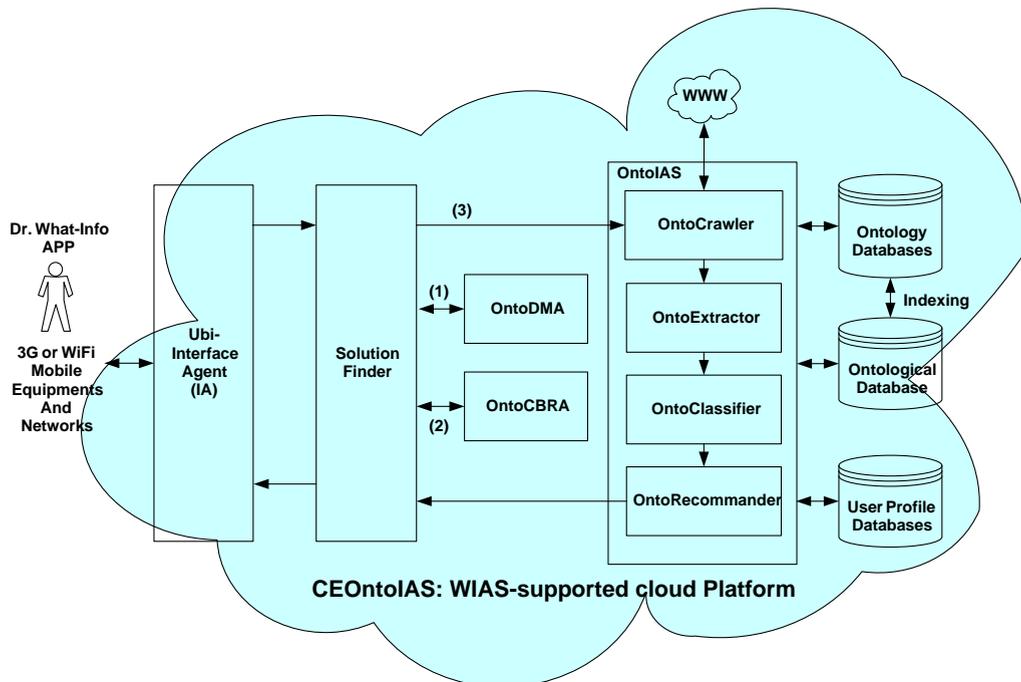
In simple terms, this paper presents an LBS Dr.What-Info system providing most popular tourist attractions and optimal routes toward destinations by means of Google Maps and image processing technique. This is done through smartphone GPS tracking, a QR/bar code reader and direct access to a cloud database indexed by the abovementioned system ontology. The development tool employed was MIT App Inventor II, while a Linux-Apache-MySQL-PHP (LAMP) server was built, with the web services developed in a script language. In comparison with Java/J2EE, LAMP has the advantage of a large amount of web resources, being lightweight, rapid application development, and the advantage of generic, cross platform, high performance and low cost over Microsoft .NET framework. Hence, LAMP is considered as the first platform choice when a website is constructed by business concerns in all respects. At the end of this work, the performance superiority of this presented LBS system is demonstrated with an archaeological site in New Taipei city, Taiwan, all the way from an easy-to-use interface operation and tour information presentation, to route planning and navigation by a joint operation between Google Maps and smart phone GPS technology.

3. System operating architectures and technologies involved

As stated previously, an LBS Dr.What-Info system was developed in such a way that the most popular Web information and optimal routes can be recommended to tourists in an easy, efficient and precise manner. This is done through APIs integrating Google Maps, and QR/bar codes are employed as an easy way to get input data. Furthermore, Web information is ranked by the number of times that the corresponding hyperlink is clicked, and is stored in a back-end cloud database for integrated services. As sketched in Fig. 5(a), a complete system operation involves (1) a way to get started with the presented system, (2) destination, (3) recommended sightseeing spots and route navigations, and (4) finally a way to leave or restart the presented system. To begin



(a) Operation flow chart of this presented system



(b) A complete backend system architecture.

Fig. 5. A complete architecture of the Dr.What-Info system.

Table 2
Frame structure of CURRL.

Slots	Description
Theme	Topic of user's command
aTheme	Description of topic
tTime	Related time of topic
tSpace	Related space of topic
object	Topic objects
oldentity	Object description
oCardinality	Number of objects
oTime	Related time of object
oSpace	Related space of object

Table 3
Examples of user commands with CURRL.

Command type	Explanation examples	CURRL
Query	What is the URL of St. John's university, SJU?	Query [Theme = +URL, aTheme = SJU, tSpace = At (WWW)]
Simple command	Anything else?	Command [Theme = +Anymore, tTime = Now, object = Related, oSpace = At (Last-one)]
Conditional command	Retrieving the university webpages with the exception of St. John's university, SJU	ConditionalCommand [Condition [Theme = -SJU, tTime = Now, tSpace = At (WWW)], Command [Theme = +University, tTime = Now, tSpace = At (WWW)]]

with, the presented Dr.What-Info system is enabled, and then a destination is entered either manually by users or automatically by a QR/bar code reader. In the latter case, a decoded message can be displayed on a smart phone, and recommended tour information, supported by a cloud multi-agent system as shown in Fig. 5(b) (i.e. CEOntoIAS, described later), is then available for tourist visits, just by a click on the scan button. A site map appears, and the websites of interest can be clicked for viewing any detailed information on recommended tourist attractions, or users can be routed to the destination by means of GPS through a click on the navigation button. The technologies involved will be discussed as follows.

Illustrated in Fig. 5(b) is a complete architecture of the Dr.What-Info system. A request for relevant information is sent to a backend server by means of Ubi-IA (Ubiquitous Interface Agent) using the CURRL (Canonical User Request Representation Language) format [17], and a solution finder in this work serves as a control center providing domain information solutions. CURRL is a frame-based command representation that makes it easy to map users' command intentions, objects and goals into corresponding frame slots. Table 2 shows the frame slots and their descriptions, while Table 3 illustrates some related explanation examples of three types of user commands for fast processing. First, the proposed system initiates the OntoDMA (Ontological Data Mining Agent) [18], which uses data mining and cluster classification techniques and provides support for the backend operation mode of the system beforehand, meaning that it actively provides profound, up-to-date and fast domain operational knowledge of the proposed system, or finds the predicted rules; furthermore, through the OntoCBRA (Ontological Case-Based Reasoning Agent) [19] service and case similarity computing for inducing the Case-Based Reasoning mechanism, the proposed system actively acquires related intelligent domain knowledge to enhance management capability for providing user-oriented and domain-related information solutions; finally, if the two abovementioned items cannot provide the solutions for the domain information, then the OntoIAS (Ontological Information Agent Shell) [17] searches for the solutions from external mechanisms through information crawling by the OntoCrawler, data extracting by the OntoExtractor, data classification by the OntoClassifier and data recommending by the OntoRecommender. Such solutions not only provide the front and back-end with the Solution Finder through the Ubi-IA, but also become the basic information materials for further prediction and case learning by the OntoDMA and OntoCBRA, respectively. Finally, the proposed system can correspond to the research subject concerning optimized solution seeking of domain information constructed by the three-stage intelligent decision-making strategy, supported by the OntoDMA, OntoCBRA, and OntoIAS. In other words, the solution sought firstly is delivered to a front-end information processing system, i.e. Dr.What-Info App, as well as a back-end one through an Ubi-IA by a solution finder, and secondly is treated as a basic information material for future expansion of learning predictions and case-based reasoning. In this context, an improved

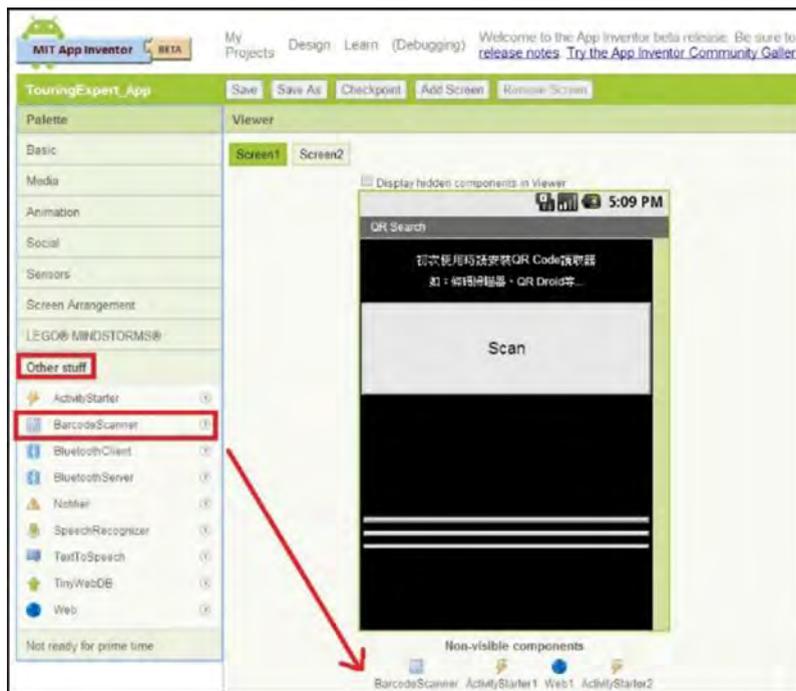


Fig. 6. A barcode scanner API.

Table 4

Detailed methods of the barcode scanner.

Block name	Function
BarcodeScanner.DoScan	Triggering BarcodeScanner to do scanning
BarcodeScanner.AfterScan	Action after BarcodeScanner scanning
BarcodeScanner.Result	Results of BarcodeScanner scanning
component BarcodeScanner	Components ordered by BarcodeScanner

robustness is seen for a backend domain information processing and decision-making system. Hence, the goal of the optimal domain solutions can be reached through a three-stage intelligent decision-making strategy, involving OntoDMA, OntoCBRA and OntoIAS, supported by the ontological WIAS-supported cloud platform (called CEOntoIAS- Cloud Extension of OntoIAS supported by WIAS- Web-service based Information Agent System) [20]. The design concept of Cloud WIAS is the notion of service-oriented architecture of Web services. WIAS employs the concept of SQL IC (Structured Query Language Integrated Circuit) [20] to construct the operational interface of cloud database as a data warehouse. A SQL IC is a SQL access template, such as a hardware IC, with different binding parameters responsible for communicating with related Web services and corresponding databases to easily access and return the corresponding query results, and then for providing related information services. Ontological Database is a stored structure designed according to the Ontology Database/Structure, serving as an ontology directed canonical format for storing webpage information processed by OntoIAS. User profile databases are responsible for recording relevant user models and providing relatively personal information services supported by OntoRecommender.

As demonstrated in Fig. 6, a barcode scanner can be directly enabled in the “Other stuff”, to scan a QR or a barcode in an APP inventor. A QR code comprises specific data, e.g. URL, phone numbers, text messages, address books, and many more. A preload barcode reader app is hence enabled, and then a scanning task is completed automatically. Table 4 presents the descriptions of each button; a complete one is presented in Fig. 7.

It is requested that the GPS function in a smart phone be enabled ahead of the Dr.What-Info system development. The GPS feature comprises a location sensor and web view, provided by an APP inventor, integrating Google Maps. As exhibited in Fig. 8, the latitude and longitude of a user’s current location is first found by the location sensor, and subsequently a destination and the starting point are specified in web view for GPS navigation. As presented in Fig. 9, saddr and daddr represent the starting point and destination, respectively.

Information on local and hot spots is provided to the system users. In the former case, the top n local spots around a destination on Google Maps in a ranking in terms of the number of times that the corresponding hyperlink is clicked are expressed as:

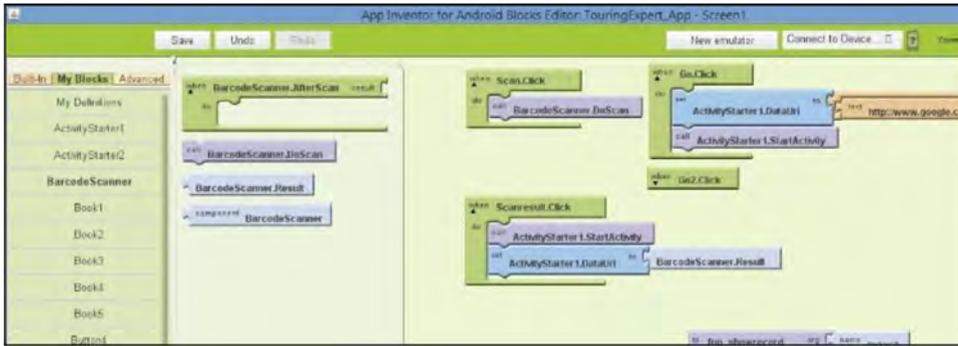


Fig. 7. Detailed features of the barcode scanner.

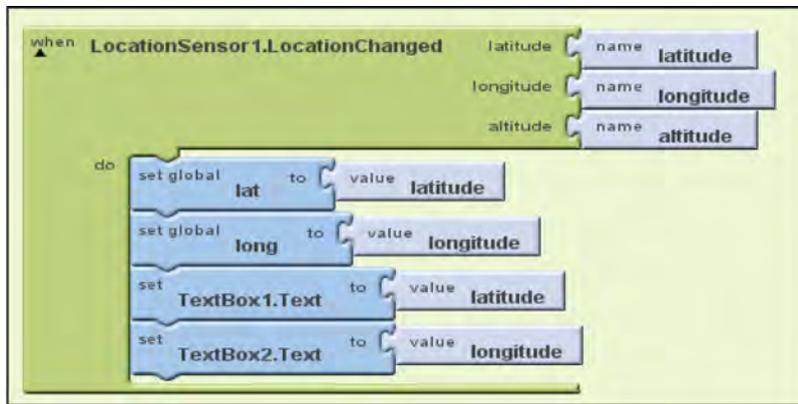


Fig. 8. Acquisition of current user location through a location sensor.

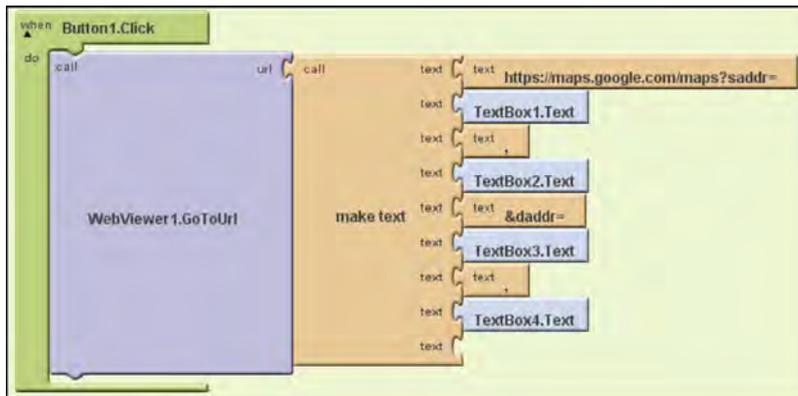


Fig. 9. Configure GPS settings for a starting point and destinations using a web view component.

$$Loc_n = \text{Max}_{i=1}^n \left\{ \sum_{P_i}^n Click_{p_i} \right\} \tag{4}$$

where i represents the number of information, and P_i the i th information page. Likewise, the top n hot spots are defined as:

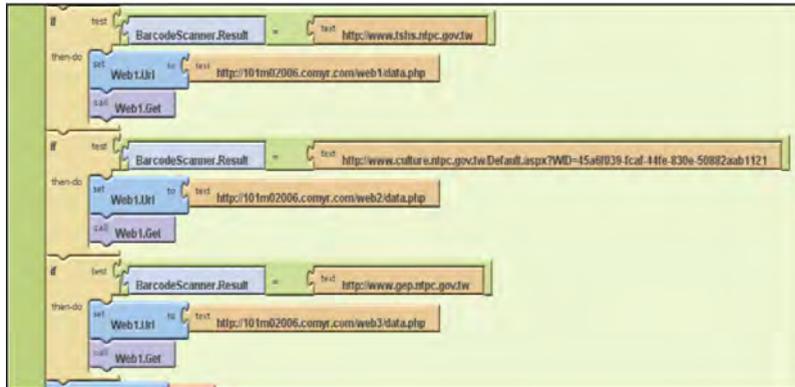
$$Hot_n = \text{Max}_{i=1}^n \left\{ \sum_{P_i}^n Click_{p_i} \right\} \tag{5}$$

In an analogy to the local spot case, it is the first n elements in a sorted sequence by the number of times that the corresponding hyperlink is clicked that are recorded offline in a periodic manner. Subsequently, the ratio between Loc_n and Hot_n is treated as a criterion to tune the weights for both hot and local spots for re-sorting the sequences. In the end, the first five elements (detailed in Section 4.3) in the resorted sequence are selected as the recommended tourist attractions.

Table 5

Respective weights for local and hot spots in various cases.

No.	Local	Hot	G_{normal} (%)
1	0	1	69.16
2	0.2	0.8	84.23
3	0.4	0.6	91.20
4	0.6	0.4	90.00
5	0.8	0.2	89.35
6	1	0	86.83

**Fig. 10.** Data lists called by decoded messages.

The tourist attractions are ranked by a ranking algorithm as suggested in [21]. Suppose that there are a total of n ranked tourist attractions (S_1, S_2, \dots, S_n), including local and hot spots, while (E_1, E_2, \dots, E_n) is a document sequence provided by domain experts. The significance of the i th element in a sorted sequence is defined as:

$$R_i = 10 \text{EXP} \left(\frac{n - i + 1}{n} \right) \quad (6)$$

For discussion purposes, the significance of R_i to the presented system is represented as SR_i , and that provided by the domain experts is represented as ER_i . For the consistency comparison between these two rankings, the ranking gain G is defined as:

$$G = \sum_{i=1}^n SR_i \times ER_i \quad (7)$$

Through such vector-like operations, a high value of G reflects a high consistency accordingly. In the end, the maximum and minimum values of G , G_{max} and G_{min} , are evaluated in advance, and a normalized ranking gain G_{normal} is defined as:

$$G_{normal} = \frac{G - G_{min}}{G_{max} - G_{min}} \times 100\% \quad (8)$$

Table 5 gives respective weightings for local and hot spots in each testing case, which are specified as 0.4 and 0.6 in case No. 3.

A backend database is constructed in such a way that information on a wide variety of tourist attractions is periodically made accessible for subsequent offline data update. In other words, corresponding data lists will be called in case there is a match between the decoded message and default strings, as illustrated in Fig. 10. Yet, column contents in a data list are accessed according to the split of strings acquired from a database, as illustrated in Fig. 11, and then stored in the corresponding data list, as illustrated in Fig. 12. In the end, the column contents are called by the system. Demonstrated in Fig. 13 are the data columns of the back-end database, and Fig. 14 is a final system presentation to users, i.e. a recommended site map for the top popular tourist attraction and for popularity rankings. In this manner, the rankings can be periodically updated offline, a way to improve the robustness of a back-end database.

4. System displays, comparisons and experiments

4.1. System displays

As presented in Fig. 15, enabling this presented tour guide app leads to a reminder message saying "Please install a QR code reader for the first time use" (an intelligent system guide) at the very top of a mobile screen. Subsequently, a QR code, as

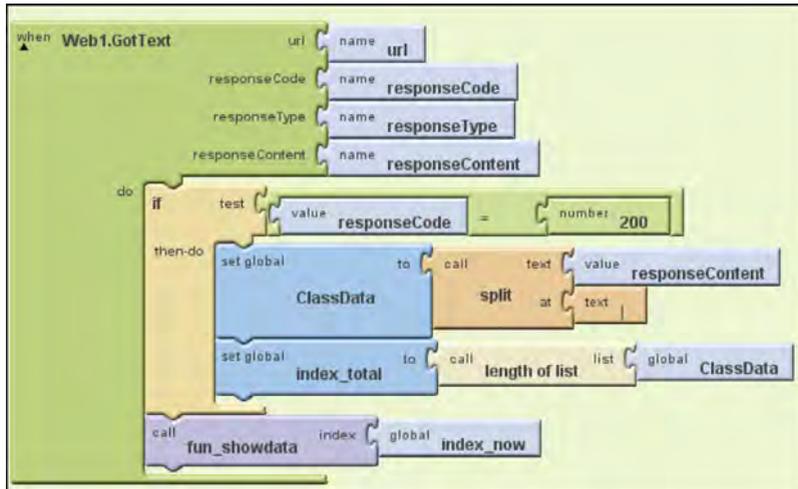


Fig. 11. Split string processing.

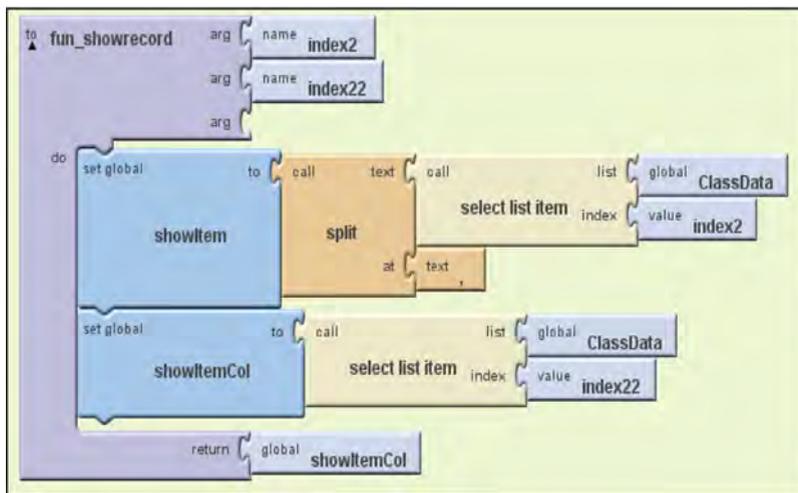


Fig. 12. Processing on item contents.

Field	Type	Collation	Attributes	Null	Default	Extra	Action
num	tinyint(3)		UNSIGNED	No		auto_increment	[Icons]
address	varchar(255)	latin1_general_ci		No			[Icons]

Check All / Uncheck All With selected: [Icons]

Fig. 13. Data fields contained in a backend data base.

十三行博物館	十三行博物館	5 MyISAM latin1_general_ci
淡水紅毛城	淡水紅毛城	5 MyISAM latin1_general_ci
黃金博物館	黃金博物館	5 MyISAM latin1_general_ci

Fig. 14. Presentation of tourism spots selected from a backend data base.

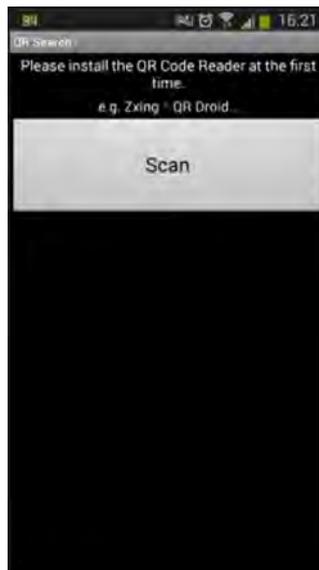


Fig. 15. An operation window of a QR code scanner.



Fig. 16. An illustration of QR codes.

demonstrated in Fig. 16, is scanned following a click on the scan button. As presented in Fig. 17, the decoded message content immediately appears on the screen, and a site map is provided by a backend database for any detailed information on this spot. Any website of interest can be clicked; furthermore, a user can be routed toward the intended destination by means of GPS technology just by a click on the navigation button, as exhibited in Fig. 18.

Due to security concern, a prompt message appears immediately for permission that Google can access the user's location, as presented in Fig. 19. In case of an access permission, three ways to reach the intended destinations: by driving, by walking and by public transportation, including route planning, trip distance and travel time, are suggested to users, as demonstrated in Fig. 20. Fig. 21 is an illustration of optimal routes by drive on Google Maps.

4.2. System comparisons

Two campus guides were developed by Kuo et al. [22] and by Tseng [23], and the latter is employed as benchmarks for this presented system. As exhibited in Fig. 22, step by step instructions are provided so as to operate this presented system; all the function buttons are displayed on the mobile screen for a complete view of the campus guide features. Accordingly, a major advantage of this work over the other two is an easy to operate user interface. Furthermore, a more user friendly QR code scanner, as opposed to that provided by a campus guide, is presented in Fig. 23.



Fig. 17. Decoded messages and websites recommended to tourists.



Fig. 18. A window on route planner operation.

Once a QR code is scanned, the decoded information on a tourist attraction and all the relevant websites of interest, including the official websites, the overview and the history at the attraction, are displayed together on the mobile screen. In contrast, a campus guide cannot provide integrated information to match visitors' needs, as shown in Fig. 24. Yet, as demonstrated in Fig. 25, the campus guide, integrated with Google Maps and GPS technology, is designed to function as a superior navigation device relative to this study. With reference to route planning, there is little difference between both, except that the campus guide is equipped with a video presentation feature, as shown in Fig. 26. The last two differences will be investigated in successive studies.

A way must be found to measure user satisfaction for the comparison of the above-stated user interfaces, e.g. the usability of interface contents and the interface usability. There are three quantities: efficiency, performance and satisfaction, stipulated by the international organization of standardization as the measures of usability. Furthermore, the so-called 5Es, i.e. Efficient, Effective, Engaging, Error Tolerant and Easy to Learn, were proposed by Quesenbery [24] as a way to understand web service users, and they are employed herein for an objective comparison on usability. In addition, as a collective measure of 'easy-to-use', up to 10 quantities, suggested by Nielsen [25], are adopted: (1) visibility of system status; (2) match between system and the real world; (3) user control and freedom; (4) consistency and standards; (5) error prevention; (6) recognition rather than recall; (7) flexibility and efficiency of use; (8) aesthetic and minimalist design; (9) help users recognize, diagnose and recover from errors; and (10) help and documentation. As tabulated in Table 6, the presented system user interface receives a collective satisfaction score of 80%, while a campus tour guide counterpart receives a score of 53%. This tremendous difference in satisfaction score is



Fig. 19. Permission request for access to user location.



Fig. 20. An illustration of resultant suggested route to destinations.

caused by superior engaging, including (1)–(8) of the 10 quantities by Nielsen as a consequence of an intelligent user friendly interface design. However, it must be pointed out that there is still room for improvement regarding ‘error tolerant’, (9) and (10) in both user interfaces.

4.3. System experiments

The authors also employ an accuracy metric to evaluate the system. As accuracy metrics, recall and precision are the most popular measures. Recall is the fraction of clicked items that are recommended in a recommendation processing and can be expressed as follows:

$$recall = \frac{|{\text{recommended_items}} \cap {\text{clicked_items}}|}{|{\text{clicked_items}}|} \quad (9)$$

Precision is the fraction of recommended items that are clicked in a recommendation processing and can be expressed as follows:

$$precision = \frac{|{\text{recommended_items}} \cap {\text{clicked_items}}|}{|{\text{recommended_items}}|} \quad (10)$$

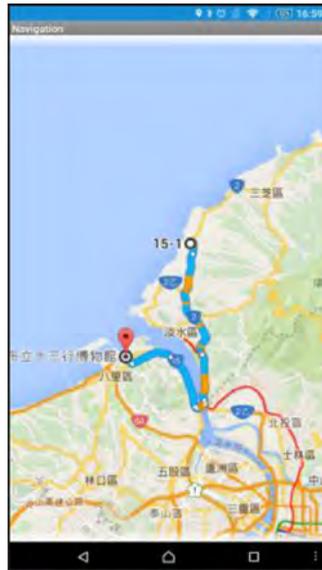


Fig. 21. An illustration of optimal routes by drive on Google maps.

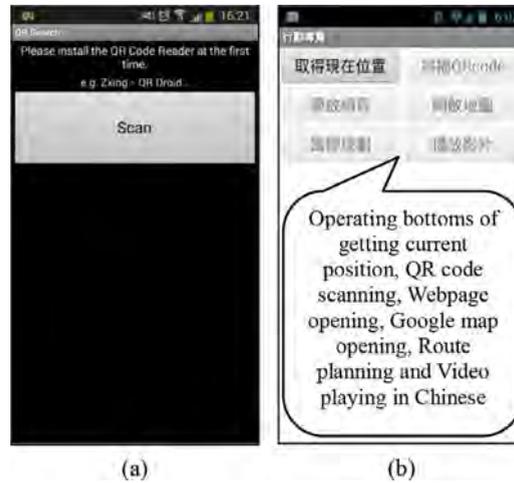


Fig. 22. Comparison on the first web page of the user interfaces between (a) the presented system and (b) a campus tour guide.

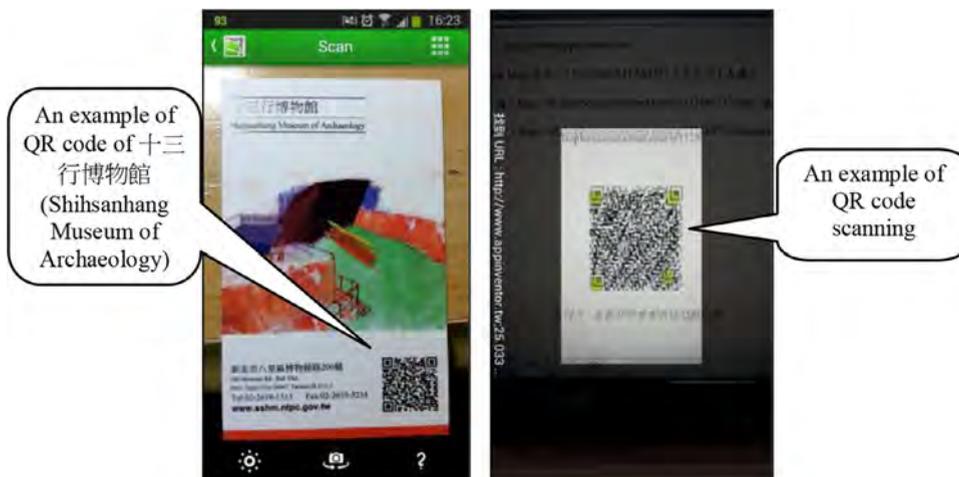


Fig. 23. Comparison on the web page of QR code reader operation between (a) the presented system and (b) a campus tour guide.

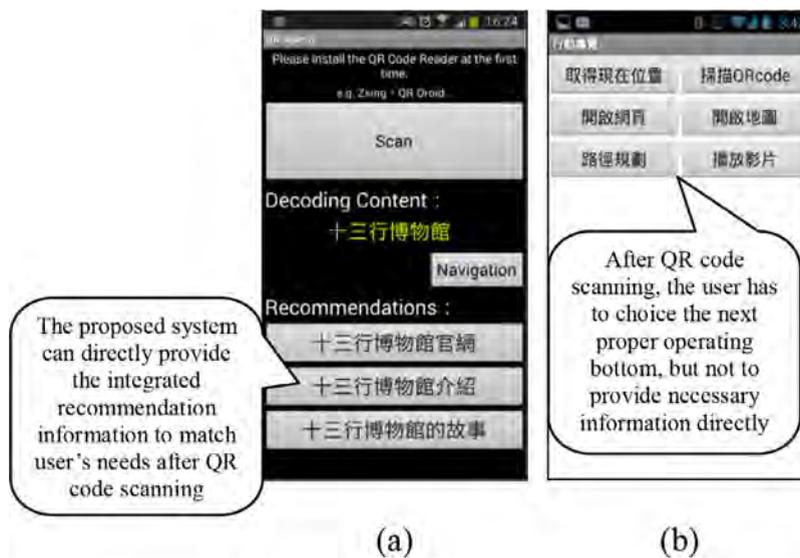


Fig. 24. Comparison of the tourism information provided by (a) the presented system and by (b) a campus tour guide.

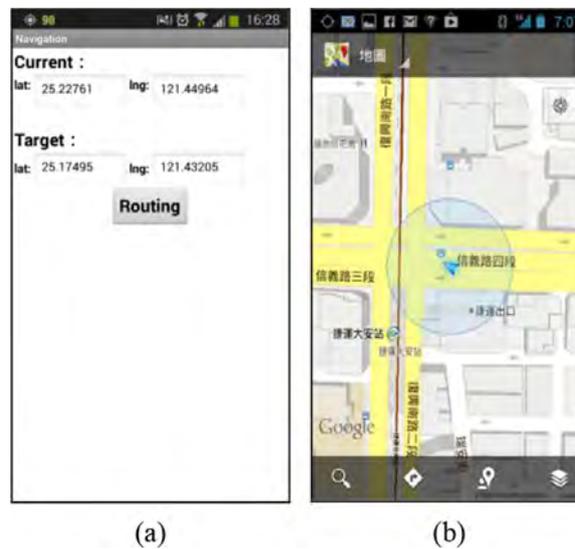


Fig. 25. GPS navigation comparison between (a) the presented app and (b) a campus tour guide.

In view of the system's high reliability and validity, the authors assume in the experiment that every recommending process contains at least one recommended item and gets at least one clicked item for avoiding the division of error by zero in the aforesaid two equations. The top- n recommendation set size is another factor affecting recommendation accuracy. The authors also discuss a reasonable size for the recommendation set. To sufficiently provide a basis for good recommendation which is big enough to provide comparisons with different parameters, i.e. recommender set size, the authors performed preliminary experiments at St. John's University in Taiwan. The experiments adopted top-3–5 recommendations, 10–20 recommender set sizes and 20–40 candidate information set sizes; the total number of unit websites of the school is 62 and, at most, five in a building with the exception of the Administration Building are used for measuring the accuracy of the recommendation. The test users include all of the freshmen students of this current year. The scenario is created to simulate students searching websites using a mobile device, with the system automatically providing related touring website information. The clicked items refer to when the student clicks on any recommended item and browses its corresponding website content. Note, that we examined only the precision rate without calculating the recall rate; in a general recommendation system, we are mainly concerned with the correct recommendation. Fig. 27 illustrates the experimental results. With an increase in recommendation set size, the average precision decreases at a diminishing rate. Fig. 27 (a) and (b) also shows that the average precision of the top-3 and top-4 recommendations are slightly distorted at Point 15 on the recommendation set size axis, while the decrease of top-5 recommendations is relatively smooth. In short, the larger the recommendation set size, the better the recommendation quality will be. However, larger



Fig. 26. Route planner comparison between (a) the presented app and (b) a campus tour guide.

Table 6

A Collective comparisons on the satisfaction of user interface.

	Standard	Items	The presented system	A campus guide
Usability	Quesenbery 5E	Efficient	O	O
		Effective	O	O
		Engaging	O	X
		Error Tolerant	X	X
		Easy-to-Learn	O	O
Easy-to-use	Nielsen	Visibility of system status	O	X
		Match between system and the real world	O	O
		User control and freedom	O	O
		Consistency and standards	O	O
		Error prevention	O	X
		Recognition rather than recall	O	X
		Flexibility and efficiency of use	O	O
		Aesthetic and minimalist design	O	O
		Help users recognize, diagnose, and recover from errors	X	X
Total comparisons			80%	53%

Note: "O" means to have this function; while "X" means none.

recommendation set sizes also means that more computation is involved. Thus, there is a traditional tradeoff between output quality and processing time. The experiment shows that the top-5 recommendation would be better.

5. Conclusion and future work

An intelligent information master system for tour guiding is proposed. Through a web service multi-agent system with an ontological system database, travel and tour information is made accessible to users in a highly efficient manner. Moreover, this Dr.What-Info app receives a collective satisfaction score of 80% in terms of Quesenbery's 5Es and Nielsen ratings. In addition, the top-n experiment shows that the top-5 recommendations would be better for solving the traditional tradeoff between output quality and processing time. However, it must be pointed out that there is still room for interface performance improvement in successive studies, including Google Map navigation in a route planner; a video presentation feature; error tolerance in interface design; Help users recognize, diagnose, and recover from errors; and Help and documentation.

For easy demonstration of the techniques in our system, the current implementation runs on a specific "Tourism" information domain. However, the authors believe that even if the domain is scaled up, our techniques are still applicable. The idea is that we are not directly scaling up our ontology; instead, we can create a complex system by integrating a set of simple systems through a multi-agent design, i.e. OntoIAS, as shown in Table 7. In addition, by exploiting the capability of Protégé, which supports the creation, extension and cooperation of a set of domain ontologies, we do not need to change our system much to transform it into a more complex system. We only need to collect and reconstruct the related ontological databases for supporting the system operation, rebuild the regular expression rule base for supporting the operations of the webpage crawler and information extractor, and devise a mechanism to ensure that a set of ontology-supported systems can cooperate effectively under our investigation.

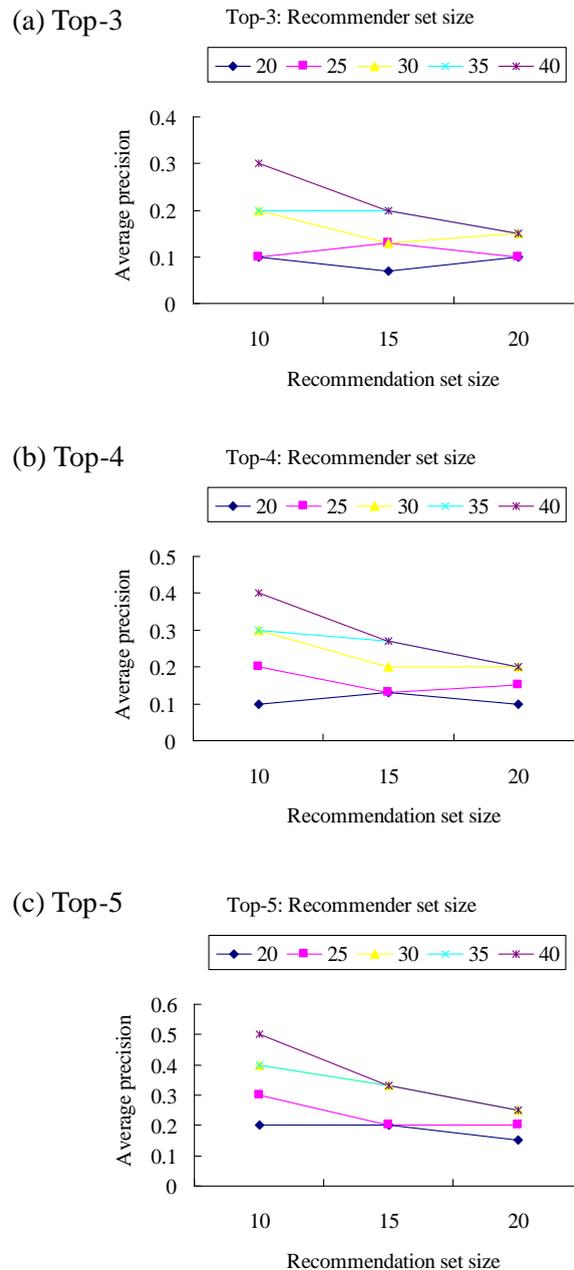


Fig. 27. Precision measures of recommendation set sizes over various candidate information set sizes.

Table 7

Assimilation and integration processes of the domain ontologies of the OntoIAS.

OntoCrawler	<ol style="list-style-type: none"> 1. Retrieving related webpages from Internet 2. Parsing contents of returned webpages 3. Storing related webpage contents after parsing
OntoExtractor	<ol style="list-style-type: none"> 1. Parsing specific ontology formats (for example, Owl/Rdf) 2. Extracting the set of main concepts of related domain ontology
OntoClassifier	<ol style="list-style-type: none"> 1. Utilizing WordNet and Eq. (3) to decide which main domain concepts are in the set of main domain concepts 2. Storing ontology information and constructing their related indexing
OntoRecommender	<ol style="list-style-type: none"> 1. Ranking ontology consistency according to Jaccard similarity 2. Recommending ontology webpages based on those ranking results

Acknowledgment

The authors would like to thank Tung-Han Tsai and Ji-Ling Gu for their assistance in system implementation and preliminary experiments. This research is sponsored under Grants 103-2221-E-129-009, 103-2632-E-129-001 and 104-2221-E-129-009 by the Ministry of Science and Technology, Taiwan. The authors feel deeply indebted to the Department of information and Communication, St. John's University, Taiwan, for all aspects of assistance provided.

References

- [1] Elgazzar K, Martin P, Hassanein H. Mobile web services: state of the art and challenges. *Int J Adv Comput Sci Appl* 2014;5(3):173–88.
- [2] Hara T, Iwamoto H. Tourism policy analysis on an eco-tour guide certification system at a UNESCO World heritage site: the Shirakami Mountain Range in Japan. *J Tour Econ Policy Hosp Manag* 2014;2(1):1–18.
- [3] Hu S, Dai T. Online map application development using Google maps API, SQL database, and ASP.NET. *Int J Inf Commun Technol Res* 2013;3(3):102–10.
- [4] Kung PC. Constructing a tour guide authoring system incorporated with Google map street-view function, Taichung, Taiwan: Department of Computer Science and Information Engineering, Asia University; 2014. Master thesis.
- [5] OA Ibrahim, Mohsen KJ. Design and implementation an online location based services using Google maps for android mobile. *Int J Comput Netw Commun Secur* 2014;2(3):113–18.
- [6] Tyagi AK, Sreenath N. Future challenging issues in location based services. *Int J Comput Appl* 2015;114(5):51–6.
- [7] Chen MF. Briefly talk about the direction of App design with mobile phones. Taipei, Taiwan: Information Technology Center, National Taiwan normal University; 2013.
- [8] Hsiao WH, Fu CC. Developing a vehicle license plate recognition system based on the android platform. *J Wufeng Univ* 2012;20(1) Wufeng University, Chiayi, Taiwan.
- [9] Yi Y. The design of 2D bar code recognition software on android. *Adv Mater Res: Smart Mater Intell Syst* 2012;442:453–7.
- [10] IDC. Taiwan IT Marketing, 2010 Top 10 predictions. Available at <http://www.idc.com.tw/about/85.html>, 2010 (Visited on May 07, 2015).
- [11] Chen Z, Xia J, Caulfield C. A survey of a personalized location-based service architecture for property hunting. *J Spat Sci* 2014;59(1):63–78.
- [12] Shih NY, Wang YP. Applying QR code in cloud mobile tour system – taking the mobile tour system of changhua confucius temple as an example. In: *Proceedings of 2012 conference on digital content and virtual learning*; 2012. p. 55–60.
- [13] Wang CH. Usability testing on the interface of the location-based m-tourism application, Kaohsiung, Taiwan: Information Management Department, National Kaohsiung First University of Science and Technology; 2012. Master thesis.
- [14] Hong SW. A study of the utility of ubiquitous mobile location based services, Taichung, Taiwan: Graduate Institute of Technology Management, National Chung Hsing University; 2013. Master thesis.
- [15] Ramya AR, Prasad Babu BR. A novel concept of MANET architecture for location based service using circular data aggregation technique. *Int J Innov Res Dev* 2014;3(1):252–8.
- [16] Soo SH, Tan MTKY, Ho JY, Low WL, Yahya MM, Ho JSY. Consumer personality, privacy concerns and usage of location-based services. *J Econ Bus Manag* 2015;3(10):961–6.
- [17] Yang SY. OntoIAS: an ontology-supported information agent shell for ubiquitous services. *Expert Syst Appl* 2011;38(6):7803–16.
- [18] Yang SY. Developing a cloud energy-saving and data-mining information agent with web service and ontology techniques. Submitted to *J Internet Technol* 2014 Paper-ID: 826.
- [19] Yang SY. Developing an energy-saving and case-based reasoning information agent with web service and ontology techniques. *Expert Syst Appl* 2013;40(9):3351–69.
- [20] Yang SY. A novel cloud information agent system with web service techniques: example of an energy-saving multi-agent system. *Expert Syst Appl* 2013;40(5):1758–85.
- [21] Yang SY, Chuang FC, Ho CS. Ontology-supported FAQ processing and ranking techniques. *J Intell Inf Syst* 2007;28(3):233–51.
- [22] Kuo YC, Wang SY, Chen PY, Fang CP. A study and application on campus guiding system, New Taipei City, Taiwan: Hsing Wu University; 2013. Technical report.
- [23] Tseng CH. A campus guiding system. Available at: http://www.appinventor.tw/camp_guide, 2013 (Visited on May 07, 2015).
- [24] Quesenberry W. Using the 5Es to understand users. Available at <http://www.wqusability.com/articles/getting-started.html>, 2011. (Visited on May 07, 2015)
- [25] Nielsen J, Pernice K. *Eyetracking web usability*. 1st ed. San Francisco, CA, USA: Peachpit Press; 2010. ISBN: 0-321-49836-4, Chapter 5 p. 53–457.

Sheng-Yuan Yang works in St. John's University, Taiwan, as Dean of computer and library, graduating from National Taiwan University of Science and Technology and major in Computer Science and Information Engineering. His research interests include agent technology, semantic web, hybrid architecture, and AI-enabled Web services.



Chun-Liang Hsu works in St. John's University as Dean of academic affairs, graduating from Industry Technology and Education Department in Taiwan Normal University and major in Electrical Engineering. Recently Hsu worked on intelligent system, embedded system, and wireless sensor network and communication.

