



Personalised Tourism Mobile Application using Semantic Web

Rukmani. G, Kartik Sridhar, Mohammed Refath, Sailusha Reddy, Hussain. F, Archanaa Rajendran

Department of Information Technology, AmritaVishwaVidyapeetham, Coimbatore, India.

Abstract: Owing to the varied interests of tourists visiting a place, listing out all possible places of interests in a particular city can be very tedious. As a result, providing assistance to tourists to visit their places of interest in an efficient manner has been highly sought after. Besides, the number of people who use smart phones is also on the rise. As a way of taking advantage of the same, this project focuses on creating a personalized Android mobile tourism application using Semantic Web. This is achieved by creating a personalised profile for the user. This profile is used to suggest places that he/she can visit based on his current location and interest as opposed to the current systems which display all possible locations based on his previous search histories.

Keywords: Semantic Web; Android; Resource Description Framework (RDF); Location Based Services (LBS); Ontology; SPARQL;

I. Introduction

Tourism is a fast growing industry which contributes enormously to the growth of a country's economy and making the tourists feel at home is an integral part of making the industry grow. In such a fast moving industry keeping track of the various places of interest in a particular city can be highly cumbersome. The most plausible way to meet such an aim would be to tap the enormous resources available in the form of smart phones and the Internet. This would be the easiest way to provide assistance to tourists, thereby helping them know their way around in foreign places. As a direct consequence, there are many versions of such applications, each having its own pros and cons. Popular examples include AA Tourism iPhone Mobile App [1] and Visit Apps [10]. However, most of these applications only help in planning trips in advance or provide dedicated tourism services for a particular city but don't provide assistance to tourists who are in the place that they are visiting. Moreover, they provide suggestions by deciphering a pattern from the user's previous search history and not necessarily by analysing what the user actually likes. Hence, the user has to spend considerable time personalizing the application to meet his needs. This implies that applications incorporating personalisation as part of its core functionalities would serve the purpose better than those that don't. One upcoming technology that can be employed to achieve personalization is making use of the semantic web. Semantic Web has been widely used to bring in personalisation to the applications in many scenarios. For example, personalising relevance in information seeking activities by analysing the user's information need from their social network is a scenario where semantic web can be used [19]. Semantic web makes the application more intelligent in terms of how the data is stored, retrieved and presented to the user [9], [23]. Along with semantic web technologies, usage of RDF graphs for data storage and the VCard ontology for association makes the application both easy to use and easy to personalize [8]. While search results are populated, a two level filtering is carried out based on the location of the user and his interests previously recorded to provide accurate results to the user which has a high chance of satisfying the user's interest. The feedback mechanism helps the application learn from user patterns and keeps the application up to date at all times. Making use of Android for the front end ensures a wide audience who can download the application onto their Android mobile phone and get personalized location based notifications about the places nearby that they would like to visit. At any point in time, the user can review all the inferences which have been made by the application. These reviews have a two pronged effect. Through the reviews, the user can give feedback about the inferences drawn by the application. This helps the application gather better feedback in the future. Also, through the user reviews, information can be gathered about the various interests of the user. This also helps in the recommendation of places of potential interest to the user. The implementation techniques followed in the application are largely based on the methods discussed in [23]. The following sections describe in detail how we use Semantic Web and Android to create the application we have just described above.

II. Related Work

While considering the domain of mobile applications that provide recommendations to tourists visiting a particular city, a prominent example is GUIDE [29]. GUIDE is an application developed for visitors of the city of Lancaster. It has varied features like flexibility, context awareness and a support for dynamic information. It is equipped with a browser based interface that caters to the information need and a map interface to display the current and the suggested locations [14]. In addition, GUIDE supports composition of and requests for recommended tours. It

operates on a client server model architecture. In this model, communication between the modules is brought about by using cell based WLAN to enable the PDA client to sense geographical position and interact with the server. The server stores all characteristic information about all the places of interest in Lancaster. Content based filtering before sending the resulting subset to the PDA. Filtering of information in the GUIDE architecture is based on a model of context distinguishing between personal and environmental context.

Another application along the same lines is CRUMPET [26], [29]. CRUMPET provides information and recommendations about tourist attractions, places to eat and tours by getting updated information from external content providers and also by taking into account the user's information need and his/her context. Although CRUMPET does not feature the tour element of GUIDE, it extends GUIDE's tourist service and improvises on it by providing personalized pro-active recommendations. CRUMPET also features an interactive map interface for user interaction. CRUMPET works on a three level architecture with a client tier, middle tier and data access tier. The data access tier performs the integration of several external sources of tourism information and makes these services homogenous with the help of interfaces accessible from the middle tier. The middle tier reads information from external providers and applies geo-coding and adapts it. The adaption process makes use of individual user models to provide content-based filtering. These models are populated by asking the user to specify his/her preferences and the system enhances the model by analysing the browsing data. The client tier contains a software agent that pools GPS information by monitoring the user's movement and automatically contacts the middle tier for proactive recommendations.

Another such application is the Tourist Information Provider (TIP) [20], [29]. TIP is a research project aimed at developing a mobile infrastructure for cooperating information services. The core system consists of an event-based architecture. The system uses the user's context in terms of location, personal preferences and his/her travel history to deliver information about tourist places of interest. Although the system serves its purpose, the biggest disadvantage is that the visualisations of the various locations provided by the third party library in the map service supports only basic rendering which makes it difficult for the designers to make distinctions between categories of sights [30].

While the above systems have their own undoubted advantages, research done in [23], [30], [29], [21], [31],[28] has shown the need for an inherent intelligence in the application. Based on the theoretical knowledge derived from these papers, we believe that we have created an application that employs the necessary intelligence along with the features of the previously existing tourism applications. The details of our implementation are discussed in the following sections.

III. Architecture

As described in the previous sections, the goal of the system is to create a personalized tourism mobile application using Semantic Web. In order to achieve the same, we propose to set up a system whose architecture is shown in Fig. 1. The main modules of the system are as follows:

- The front-end is an Android application which navigates to two different windows- Map window and User profile window. The user profile window allows the user to add more details to his profile and the Map window displays a map using Google Maps. The Map window, as shown in Fig. 2 is used to indicate the user's current location and also provide suggestions to the user on locations that the user might like to visit within k metres vicinity of his current location.

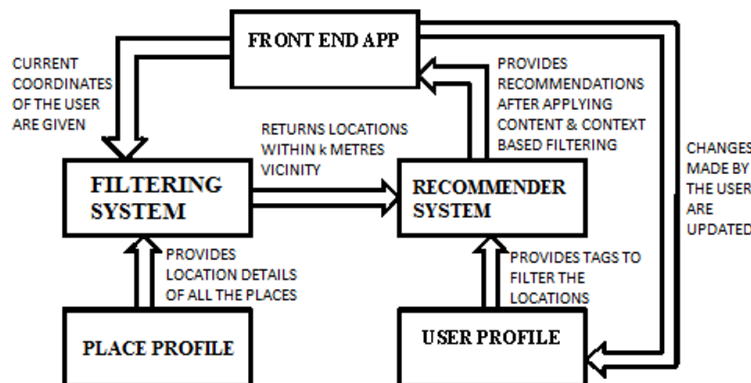


Fig. 1 – System architecture

- The user profile is stored in the form of an RDF graph. All the activities of the user's interest are stored in the form of tags and recommendations are provided to the user based on the tags stored.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#" >
  <rdf:Description rdf:nodeID="A0">
    <vcard:CATEGORIES>CINEMA</vcard:CATEGORIES>
    <vcard:GEO>11.0084N, 76.9598E</vcard:GEO>
    <vcard:CATEGORIES>MALL</vcard:CATEGORIES>
  </rdf:Description>
  <rdf:Description rdf:about="http://TamilNadu/Coimbatore">
    <vcard:N rdf:nodeID="A0"/>
    <vcard:FN>Brookfield Mall</vcard:FN>
  </rdf:Description>
</rdf:RDF>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#" >
  <rdf:Description rdf:about="http://TamilNadu/Coimbatore">
    <vcard:N rdf:nodeID="A0"/>
    <vcard:FN>Fun Republic Mall</vcard:FN>
  </rdf:Description>
  <rdf:Description rdf:nodeID="A0">
    <vcard:CATEGORIES>CINEMA</vcard:CATEGORIES>
    <vcard:GEO>11.00N, 77.01E</vcard:GEO>
    <vcard:CATEGORIES>MALL</vcard:CATEGORIES>
  </rdf:Description>
</rdf:RDF>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#" >
  <rdf:Description rdf:nodeID="A0">
    <vcard:GEO>11.0441N, 76.85194E</vcard:GEO>
    <vcard:CATEGORIES>TEMPLE</vcard:CATEGORIES>
  </rdf:Description>
  <rdf:Description rdf:about="http://TamilNadu/Coimbatore">
    <vcard:N rdf:nodeID="A0"/>

```

Fig. 2 – Sample RDF graph

- The place profile stores details about all the places in the given city in the form of an RDF graph. Each location, depicted as a model within the RDF graph, has certain tags that are associated with it. These tags are essentially keywords that are used to categorise the places according to their features. For example, the tags associated with a typical mall would be - mall, shopping, movie, food. These tags are used in order to make suggestions to the user. Apart from the characteristic tags, the geographic coordinates of each place is also stored as part of the graph in order to approximate the vicinity of the location from the user's current location. Fig. 3 depicts a sample place profile using RDF graphs for the city of Coimbatore.

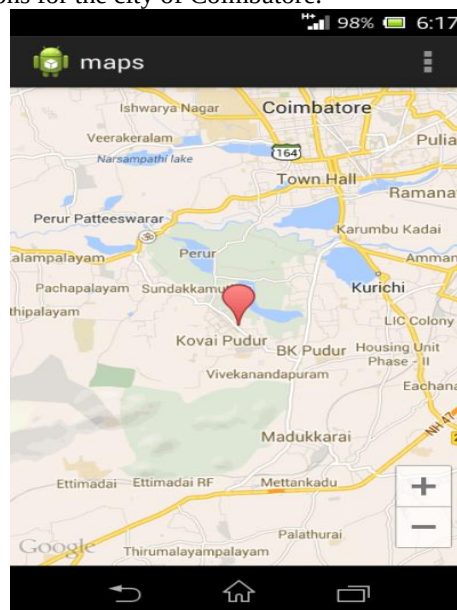


Fig. 3 – Screenshot of Map Interface

- The filtering system takes in all the locations from the item profile and filters the locations that are within the k metres radius from the user's current location.

- The recommender system takes all the locations provided by the filtering system and filters only those locations that have tags matching the ones in the user profile. This way only the places relevant to the user are displayed to him/her.

IV. Implementation

As discussed previously, the mobile application which we have created does the following:

- Create a profile for the user wherein his interests are stored.
- Create a profile for the various tourist locations in the city under consideration.
- Obtain current location of the user using Google Maps API.
- Provide recommendations to the user based on two factors - proximity and user interests.
- Gather feedback data from the user interactions with the application and use it to further build the user profile.

In order to create the application described above, we consider the architecture and modelling described in section 3. With such a system setup, the above mentioned functions of the application are described below.

Creating a User Profile

The application recommends tourist places to a user based on his interests. These interests are obtained after profiling the user's actions with the application. The user, in our case, is assumed to be a tourist in the city of Coimbatore, India. Initially, the user profile is empty. In order to populate information about the user, the first time the user uses the application, he/she is requested to answer a few questions. These questions provide a valuable insight into the users' interests and are stored as a part of the user profile, which will be discussed later. While answering the questionnaire is optional, it is advisable not to skip it, as the information gathered helps the application filter and provide recommendations which are better suited to the user's interests. However, even if the user profile is empty at a particular point, recommendations based on the proximity to the user's current location are provided.

In order to achieve the above functionality, the user profile is implemented as an Resource Description Framework (RDF) graph. In this RDF graph, the user is represented as a model and his interests are stored as the properties of the model. These properties also store the current location of the user obtained dynamically. The RDF graph was chosen to model the user profile because it is a language predominantly used for representing information about resources in the World Wide Web. RDF is intended for situations in which the information represented is used by the application for processing, than for just display purposes. The common framework provided by RDF for expressing this information helps in the exchange of information between applications without loss of meaning. Since it is a common framework, application designers can make use of common ontologies. The ability to exchange information between different applications makes the information available in a readable and interpretable format to other applications apart from those for which it was originally created. RDF identifies things using web identifiers and describes resources as a property name value pair. This enables RDF to represent simple statements about resources as a graph of nodes and arcs representing the resources [8], [17].

Ontology is a data model used to represent knowledge that can range from ideas to facts as a set of concepts in a particular domain and to derive relationships between these concepts. It is used to deduce knowledge about the domain rather than to know about the particulars within the domain. Ontology is represented in the web using OWL which is the W3C standard [11]. In knowledge representation, ontology is used to describe the concepts and relationships in the application domain. It is used to provide meta-information, i.e. information about information. In the context of semantic web, ontology is used to denote the relationships between various terms within the information. It helps in conveying background information, thus enriching the description of the data hence making the context of the information more explicit. VCard ontology is used to represent tourism domain for the framework [12], [25], [22], [15], [27].

Jena, an open source Semantic Web framework for Java provides an API to read and write data in RDF graphs using IDEs like Eclipse, NetBeans. A model, which is an abstract representation of the graph, can be populated with data from varied sources like files, databases, URLs [2], [24], [13]. In the case of this application, the user profile consists of a model for the user where tags are generated based on the user's answer to the questions asked and stored as properties in the model. Then, the user model can be queried using SPARQL queries to provide recommendations.

Creating a Place Profile

In order to map the user interest to places which might be of possible interest, we need to model all the possible tourist destinations appropriately. These places should have an associated description which is necessary to help the application decide on the places of interest to recommend. The features of each place are stored as a set of tags.

Upon a simple existential comparison with the interests of the user, user specific tourist destinations can be extracted from the list of all possible tourist spots.

To implement the above, we model the profile of each place as an RDF graph. In an approach similar to the one followed for the building of an user profile, each place is represented as a model, and the defining features associated with that place are stored as properties of that particular model. Along with this data, the latitude and longitude combination is also stored, to help uniquely identify the tourist spot. Here we make use of the VCard ontology in order to both store the latitude and longitude co-ordinates as well as help make inferences between the RDF graphs for the user profile as well as the RDF graph for the Place profile.

VCARD is a standard file format for electronic business cards. VCARD, which is often attached to email messages, can also be exchanged on the World Wide Web or instant messaging. Apart from supporting the regular information like name, address, phone numbers, email addresses, it can also store the latitude and longitude details using the geo tag [18].

Obtaining Current Location

The current location of the user is a primary factor in deciding the recommendations which are provided. The underlying logic for such an implementation is that, if the tourist is not specific about any particular place he wants to visit, then it is better to provide him with a list of all places close by and make him choose. Such an approach will not only ensure that the user visits all the places in a particular area which could be of possible interest to him, but also groups the various places of interest in an attempt to cut down on the distance travelled and the net travel time. In order to do so, the map of the user's region obtained using the Google Maps API [4], [5], [6], [7]. Using this map, and the GPS API of Android, the current location of the user is extracted in the form of latitude and longitude. This, in essence, uniquely describes the user's current location, both to the application as well as the user himself by visually projecting it on the map.

Recommending Tourist Spots

Once the user's current location has been extracted, the application has to recommend spots which are both similar to the user interest as well as within close proximity of the user's current location. This two pronged filtering is carried out by the filtering module and the recommendation modules. The first level of short listing happens in the filter module. The functionality of this module is to return only those tourist places which are within a k km radius of the user's current location. In order to facilitate this, one hashing function and one hash map is used. At the hashing function, the geographic co-ordinates (latitude and longitude) of each of the tourist destinations are hashed to get a unique value for each location. This is done to facilitate the usage of both the latitude and longitude as prospective keys to the hash map. This unique hash value is then used as a hash key for the hash map. This hash map contains a key-value mapping, with the hash value from the previous function acting as a key and the corresponding place as value. The hashing function is implemented to be tolerant to k kilometres. As a result, when the geographic co-ordinates of the user's current location is given to the hash functions, only those models from the place profile which are sufficiently close by are returned. This acts as the initial filtering mechanism.

Having obtained the smaller set of locations that are within k metres vicinity, it is now required to filter these places based on the user's interests. This is done by selecting only the places that have similar tags as those in the user profile. This provides a means to eliminate all the places near the user's location that don't interest him/her. The tags, that are stored as properties in the user model, are obtained by querying the user model. This is done by using SPARQL querying language [3], [16]. Similar queries are written to obtain the properties of all the models in the subset of place profile. Once the tags for both the user profile and the subset of the place profile are returned, an existential comparison is carried out. Only those places in the subset that have at least one matching tag as those in the user profile are returned as recommendations to the user. These recommendations are sorted based on the number of matches of tags, i.e., those places in the place profile which have a higher number of tags matching to the user profile are ranked higher. In the event that the query to the user profile returns no values (the user has not had any profiling done as of yet), all the places that are present in the subset of the place profile are provided as recommendation to the user.

Gathering Feedback Data

In an attempt to enhance the future results which are to be provided, the user choices are monitored and feedback data is collected. This helps profile the user better. For every location that the user selects from those recommended to him, all the tags associated with that location are added to the user profile. As a result, the number of properties in the user profile model keep increasing. This way, the user profile keeps growing and reflects the choices made by the user over his/her interactions with the application. However, to keep a check on the number of tags being associated to the user profile, and to decrease the possibility of too many matching places, we provide the user the option to review all the tags that are present in his/her profile and edit them as required. We believe such a learning

capability to the system will help in enhancing the results based on both better recommendations and newer recommendations.

V. Merits

The application has been developed in the manner as described in the above sections. In this section, we consider the merits and the demerits of our application.

Merits

- This application gives the user a level of independence and sophistication. He/she can visit places of interest even if he/she is not a native of the region and/or does not know the local language.
- Tourists don't have to depend on external entities to guide them. This could help prevent the tourists from being duped.
- By using the Semantic Web, which is expected to be in use for the next few decades, we have given the application room for development and maturity in the coming years.

Demerits and Possible Improvisations

- The end target for this application is the accumulation of all the data from all the tourist spots from around the world. In such a case, the amount of data to be stored will be extremely high. To overcome this drawback, we propose to use a cloud based system, which shall download the place profiles of only those tourist destinations which are in the same region as the user.
- Right now, in order to test the feasibility of this system, place profiles have been manually added to the database. However, this will not be feasible for a large data set. So, we hope to solve this problem by extracting both prospective tourist locations from the Google Maps API. We also hope to associate relevant metadata from various sources in the internet, thus automating the place profile population process.
- As of now, the application does only a basic existential search. In the future versions of this application, we hope to bring in context based matching of the tags based on factors like weather, time a place is open, and any dangerous news regarding the area under consideration. This we believe will help refine the search result produced.

VI. Conclusion

In the above sections of the paper, we have described the method followed to develop an application which we believe will help tourists by recommending tourist spots tailor made for their interests. While there is scope for future work to be done, we believe this application will help reduce the troubles of many tourists, at least initially in the city of Coimbatore. We also believe that once some of the demerits of the application are addressed, this application will be really useful for tourists all over the world, thus reducing their dependence on third parties and blogs.

References

- [1] Aa tourism mobile application.
- [2] Apache jena - home.
- [3] Apache jena - sparql tutorial.
- [4] Emil's design - strive for perfection. » blog archive » guide: Googlemaps v2 for android : Creating your google map application:.
- [5] Emil's design - strive for perfection. » blog archive » guide: Googlemaps v2 for android : Getting the api key.
- [6] Google maps android api v2 — google developers.
- [7] Latest "rdf primer" versions.
- [8] Semanticweb.com - the business of big data, linked data, smart data.
- [9] Visitapps | mobile apps for tourism.
- [10] World wide web consortium (w3c).
- [11] CAO, T.-D., PHAN, T.-H., AND NGUYEN, A.-D. An ontology based approach to data representation and information search in smart touristguide system. In Knowledge and Systems Engineering (KSE), Third International Conference on (2011), IEEE, pp. 171–175.
- [12] CAO, T.-D., PHAN, T.-H., AND NGUYEN, A.-D. An ontology based approach to data representation and information search in smart touristguide system. In Knowledge and Systems Engineering (KSE), 2011 Third International Conference on (2011), IEEE, pp. 171–175.
- [13] CARROLL, J. J., DICKINSON, I., DOLLIN, C., REYNOLDS, D., SEABORNE, A., AND WILKINSON, K. Jena: implementing the semanticweb recommendations. In Proceedings of the 13th international WorldWide Web conference on Alternate track papers & posters (2004), ACM, pp. 74–83.
- [14] CHEVERST, K., DAVIES, N., MITCHELL, K., FRIDAY, A., AND EFSTRATIOU, C. Developing a context-aware electronic tourist guide: some issues and experiences. In Proceedings of the SIGCHI conference on Human factors in computing systems (2000), ACM, pp. 17–24.
- [15] COB, Z. C., AND ABDULLAH, R. Ontology-based semantic webservices framework for knowledge management system. In Information Technology, 2008. ITSIM 2008. International Symposium on (2008), vol. 2, IEEE, pp. 1–8.
- [16] EUZENAT, J., POLLERES, A., AND SCHARFFE, F. Processing ontology alignments with sparql. In Complex, Intelligent and Software Intensive Systems, 2008. CISIS 2008. International Conference on (2008), IEEE, pp. 913–917.
- [17] GROBE, M. Rdf, jena, sparql and the 'semantic web'. In Proceedings of the 37th annual ACM SIGUCCS fall conference (2009), ACM, pp. 131–138.
- [18] HALPIN, H., IANNELLA, R., SUDA, B., AND WALSH, N. Representing vcard objects in rdf. W3C Member Submission 20 (2010).

- [19] HEATH, M. T., ET AL. Personalizing relevance on the semantic web through trusted recommendations from a social network.
- [20] HINZE, A., AND VOISARD, A. Location-and time-based information delivery in tourism. In *Advances in Spatial and Temporal Databases*. Springer, 2003, pp. 489–507.
- [21] HUSAIN, W., AND DIH, L. Y. A framework of a personalized location based traveler recommendation system in mobile application. *International Journal of Multimedia & Ubiquitous Engineering* 7, 3 (2012).
- [22] JEPSEN, T. C. Just what is an ontology, anyway? *IT professional* 11, 5(2009), 22–27.
- [23] MAHMOOD, F., AND BIN ABDUL SALAM, Z. A conceptual framework for personalized location-based services (lbs) tourism mobile application leveraging semantic web to enhance tourism experience. In *Advance Computing Conference (IACC)*, 2013 IEEE 3rd International (Feb 2013), pp. 287–291.
- [24] MCBRIDE, B. Jena: Implementing the rdf model and syntax specification. In *SemWeb* (2001).
- [25] MOUHIM, S., AOUI, A., CHERKAoui, C., MEGDER, E., AND MAMMASS, D. Towards a knowledge management system for tourism based on the semantic web technology. In *Multimedia Computing and Systems (ICMCS)*, 2011 International Conference on (2011), IEEE, pp. 1–6.
- [26] POSLAD, S., LAAMANEN, H., MALAKA, R., NICK, A., BUCKLE, P., AND ZIPL, A. Crumppet: Creation of user-friendly mobile services personalised for tourism.
- [27] ŠAŠA, A., AND KRISPER, M. Ontology-based knowledge management in service-oriented systems. *Journal of Information and Organizational Sciences* 35, 1 (2011), 105–118.
- [28] SHETH, A., ILARRI, S., MENA, E., ET AL. Semantics in location-based services. *IEEE Internet Computing* 15, 6 (2011), 0010–14.
- [29] WIUM, M. Design and evaluation of a personalized mobile tourist application.
- [30] YANG, F., AND WANG, Z. A mobile location-based information recommendation system based on gps and web 2.0 services. *Database* 7 (2009), 8.
- [31] YU, C.-C., AND CHANG, H.-P. Personalized location-based recommendation services for tour planning in mobile tourism applications. In *E-Commerce and Web Technologies*. Springer, 2009, pp. 38–49.