Collective Intelligence in Mobile Consumer Social Applications

S. Diplaris, I. Kompatsiaris
Informatics and Telematics Instititute
CERTH
Thessaloniki, Greece
[diplaris, ikom]@iti.gr

A. Flores, M. Escriche
Telefonica R+D
Valladolid, Spain
[anafc, mev]@tid.es

B. Sigurbjornsson, L. Garcia,
R. van Zwol
Yahoo! Research
Yahoo! Iberia
Barcelona, Spain
[borkur, lluis, roelof]@yahoo-inc.com

Abstract – **This paper presents a mobile software application for the provision of mobile guidance, supporting functionalities, which are based on automatically extracted Collective Intelligence. Collective Intelligence is the intelligence which emerges from the collaboration, competition and coordination among individuals and can be extracted by the analysis of mass amount of user-contributed data currently available in Web 2.0 applications. More specifically, services including automatic Point of Interest (POI) detection, raking, search and aggregation with semi-structured sources (e.g. Wikipedia) are developed, which are based on lexical and statistical analysis of mass data coming from Wikipedia, Yahoo! Geoplanet, query logs and flickr tags. These services together with personalization functionalities are integrated in a travel mobile application, enabling their efficient usage exploiting on the same time user location information. Evaluation with real users depicts the application's potential for providing a higher degree of satisfaction compared to existing travel information management solutions and also directions for future enhancements.**

Collective Intelligence; mobile guidance; mobile social applications; mass user feedback analysis; text mining; POIs extraction

I. Introduction

Web 2.0 technologies have unveiled the role of user generated content (UGC). The large-scale manual annotation of media objects, known as tagging, enabled what for long was deemed impossible: effective retrieval of media assets on the Internet through annotations provided by users. User generated content together with the additional information provided (e.g. tags, comments, rankings, votes, favorites etc) does not only allow efficient retrieval, but in addition it provides an ample opportunity to extract collective knowledge from social media sharing sites, which can be used to enhance the user (media) experience across various application domains. For example, clustering of tags assigned to photos according to the geographical location can lead to "user generated maps", which reflect the users views of geographical areas extension and boundaries [1]. Other examples include support for emergency response [2], patterns, trends and facts extraction by analysing the patterns of user activities emerging in such systems, e.g. the evolution of popularity for online stories [3].

The extraction of such hidden knowledge, also known as Collective Intelligence, can emerge from two different aspects: large amount of input data and combination of different intelligence layers, i.e. personal, media, mass, social and organisational intelligence. A number of techniques exploiting the availability of large amount of user contributed content and analysis results coming from different layers, provide higher level information and allow the linking of user-generated content with structured and semi-structured sources. These techniques include statistical analysis, machine learning, data mining, pattern recognition, social network analysis, etc. exploiting information such as joint groups of related tags and social data sources and co-occurrences of tags attached by users [4] [5].

In this paper, we present a mobile application, which makes use of automatically generated Collective Intelligence results and personalised recommendation techniques in the context of a travel scenario. The services provided facilitate the creation and planning of trips or events, through a mobile terminal, by enabling automatic Point of Interest (POI) detection, ranking, search and aggregation with semi-structured sources (e.g. Wikipedia), which are based on lexical and statistical analysis of mass data coming from Wikipedia, Yahoo! Geoplanet, query logs and flickr tags. While many mobile applications already provide personalization services, the use of mass content analysis results in mobile applications is rather limited and can be used to provide efficient and affordable functionalities to match user needs. For example, the well-known TripIt [6] mobile application simply reproduces personalised travel details already edited by the user in a desktop version of the product, while planning a trip. The herein described mobile application fetches personalised recommendations on the spot, by using analysis techniques for harnessing the power of mass user contributed content.

As the use of mobile is becoming more important, both in trip planning and while travelling, a lot of relevant mobile applications have been developed also in the literature with the aim to better organize a trip and manage information available on the spot [7][8][9], but again with no provision for exploiting mass intelligence. The herein presented mobile guidance application is based on a robust architecture that can support the User Interface requests by integrating the various analysis techniques, thus enabling the provision of the above described intelligent services. The User Interface also supports personalized access to these services, as well as the efficient exploitation of the mobile terminals resources (GPS etc.) by the back-end services.

This paper is organised as follows: the user requirements and a use case scenario are first described in Section II, while

978-0-7695-4084-9/10 $26.00 © 2010 IEEE
DOI 10.1109/ICMB-GMR.2010.71

in Section III an overview of the mobile guidance application is given. Section IV deals with the back-end intelligent services of the application, and Section V gives an insight of the system architecture. Section VI is devoted to the user evaluations of the mobile guidance application, presenting results from field experiments. The paper ends with the concluding Section VII.

II. Requirements and Scenario

From an IS design theory point of view, in the Walls et al. [10] sense, the system design consists of three interrelated elements: (1) a set of user requirements, (2) principles governing the development process, and (3) principles governing the design of the system. In this context, the initial hypotheses and principles in forming the user requirements are driven by a typical travelling scenario described in the following. Principles regarding the development process and design are given in Section V.

The developed services can analyze user contributed content from various sources, the application user profiles and their travel behaviour to help them and other members to discover essential information about what to see and do on travel or one-day cultural trip events. More specifically, by making use of automatically extracted Collective Intelligence results, it assists users in a travel exploration experience by identifying points of interest (POIs), ranking and prioritizing them (according to the most popular places and user´s profiles) and by presenting them along with additional background information aggregated from different sources.

The scenario contains two main parts: the Travel Preparation and Mobile Guidance. During the travel preparation part, users need a tool, which is able to provide them with information about the different candidate places to be explored and visited. Ideally, the users need a centralized point where to find information about multimedia content, text and opinions from other users. The Travel Preparation tool, in the form of a web application, provides the relevant information that users need to prepare their travel, e.g. information about the locations, multimedia content, points of interest, opinions, etc. This information is aggregated with content coming from different sources (e.g. Wikipedia), and is ranked according to trends to add value to user's experience. In addition, users activity is monitored in order to personalize their mobile experience.

In the second part, Mobile Guidance, users perform the planned trip. While doing so, mobile devices represent the perfect tool to allow the user to get relevant and position information. With the features offered by such devices and the developed application, users are able to access relevant information about their physical environment and search for new events or points of interest.

The system will provide recommendations based on the user profile analysis in connection with their current location, and can make event suggestions. Users can also take pictures and record videos of the places they are visiting, which are likely to correspond to the ones chosen in the travel preparation phase. These data, generated by users, such as images and videos, comments, ratings and notes, can then be added into the system where all this content is stored and shared with other users, enriching the repository of information for other users of the system.

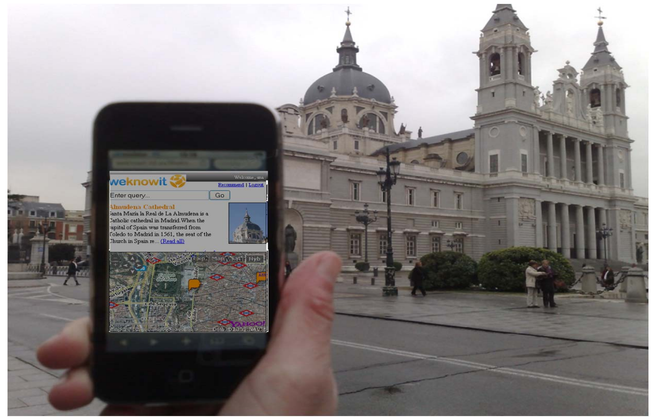

Figure 1. POI information: "Almudena Catedral, Madrid"

III. Mobile Guidance Overview

In this section, the Mobile Guidance part is described in more detail. The Mobile Guidance is an application developed for Android and iPhone including the following functionalities:

- **Search for places** (e.g. Barcelona). When the user makes a place query, the Mobile Guidance application returns a list of places that match the query. In some cases there might be several places called "Barcelona", and a user searching for Barcelona should choose between the options presented as a response to their query (Figure 3. a). When the GPS information is available, the application is able to automatically perform the place disambiguation task and directly present the user with results according to his location.
- **Retrieve information about a place.** When the user selects one place from the list, the system presents them the available information about this place and additional information for several POIs in this location.
- **Search recommended POI.** The user makes a POI query and the system returns a list of recommended and ranked POIs that match the query.
- **Retrieve information about a POI**. The user selects one POI from the list of recommended POIs, and the system shows information about this POI (Figure 1.).
- **Display image gallery of a place or POI**: Since images are a powerful way to convey information, the system makes available to the user a specific functionality for accessing a gallery of images of a place or point of interest, at the same time that the information about a place or POI is being retrieved (Figure 2.).
- **Search for friends:** When a group of friends is travelling together, they can find it interesting and helpful knowing where other members are. When a user applies

for it, the Mobile Guidance tool will display the location of his friends on a map.

- **Logs store:** User position is stored in a database every time that the user makes an operation but it´s also automatically and periodically stored by the system. The storage of the GPS location of users, allows the mobile application to trace the user itineraries and to provide useful data for further analysis for the improvement of the personal recommendations.

Figure 2. Image gallery

IV. Identifying Points of Interests using Collective Intelligence

In this section, the Collective Intelligence back-end services for identifying points-of-interest (POIs) related to a place are described. We combine several sources of mass information in order to automatically extract and link the required information:

- **Wikipedia:** We extract lists of POIs from Wikipedia using explicitly geo-coded pages (http://wikipedia.org). Media intelligence is hereby enabled.
- **Yahoo! GeoPlanet™:** We relate the POIs to Places (cities, states, and counries) using Yahoo! GeoPlanet -- a resource for managing all geo-permanent named places on Earth (http://developer.yahoo.com/geo/geoplanet/), thus enabling *organizational intelligence*
- **Search Engine Query Logs and Flickr Photo Annotations:** For a given place we create a ranked list of the most interesting POIs is generated using a search engine query-log and Flickr photo annotations (*mass intelligence*).

We will now describe the identification process in some detail, but the interested reader is referred to [11] for more details.

A. Extract Points of Interest

We parse the whole text dump of Wikipedia and extract the titles of all geo-coded pages together with the geo-coordinate. E.g., for the Wikipedia page for the Tower of the Winds (http://en.wikipedia.org/wiki/Tower_of_the_Winds) we extract information of the form:

- **Title:** Tower of the Winds
- **Longitude:** 23.727003
- **Latitude:** 37.974178

We then use bounding box information from Yahoo! GeoPlanet to identify the surrounding town, county, state and country:

- **Town:** Athens (WOEID: 946738)
- **County:** Athina (WOEID: 24543541)
- **State:** Attiki (WOEID: 12577879)
- **Country:** Greece (WOEID: 23424833)

Where WOEID is a so-called Where-on-Earth Identifier used to identify locations in GeoPlanet. WOEIDs are used to distinguishing between ambiguous locations, such as Athens (Greece) and Athens (Georgia, U.S.) In the following text we will however for sake of simplicity ignore the WOEID information and use the place name to refer to locations.

(a) (b)

Figure 3. Search Places (a) and POIs (b): Application of the disambiguation while the user searches Places

Using the combination of Wikipedia titles and GeoPlanet information, we create tuples of the form:

- (Athens, Tower of the Winds)
- …
- (Greece, Tower of the Winds)

Having repeated this process for all geo-coded Wikipedia pages, a long list of Points of Interests for a large number of places on earth is derived. E.g., a list of 90 Points of Interest is generated for the city of Athens and 1996 Points of Interest are induced for Greece. Given this large number of POIs it is crucial to provide a ranking so that for a given place we can get a short list of the top POIs.

B. *Ranking Points of Interest*

For ranking Points of Interest for a given place we use term co-occurrence statistics from different sources. For example, for the tuple (Athens, Tower of the Winds) we look at the co-occurrence of the two entities in:

- **query sessions:** i.e., how often people search for Athens and Tower of the Winds within a certain time limit;
- **query terms:** i.e., how often people search for Athens and Tower of the Winds with in the same query; and
- **flickr annotations:** i.e., how often people annotate the their photos with both Athens and Tower of the Winds.

For each of the information source a number of co-occurrence metrics is calculated, such as conditional probability, joint probability, mutual information gain, and is consequently used the to create a ranked list of Points of Interest for a given place. We refer to [11] for more details.

C. *Serving Points of Interest*

Points of Interests are exposed in a number of services in the developed application as was also described in the previous section:

- **Search places:** Given a place name, such as Athens, the service returns a list of places called Athens in decreasing order of the number of Points of Interest.
- **Get Points of Interest for a Place:** Given a place identifier such as Athens, Greece (WOEID=946738) the service returns the top-n most relevant Points of Interest for the place.
- **Get Information for a Place or Point of Interest:** Given a Place or a Point of Interest the service returns detailed information about the entity, such as a short text description from Wikipedia, latitude, longitude, etc.

We use these services to drive the Mobile Guidance described in the remainder of this paper.

V. ARCHITECTURE FOR MOBILE GUIDANCE

The developed application is built as part of the FP7 project WeKnowIt: Emerging, Collective Intelligence for personal, organisational and social use [12], and makes use of the WeKnowIt architecture for its development.

The design theory principles followed during the development process of the application deal with the fact that the application is constantly enriched, thus new services are likely to emerge. Therefore, the API of existing services should be able to undergo some changes. In this case, iterative development approach proves to be more plausible. For each iteration, the goals, in terms of required API, are based on the actual needs of the supported services. As a consequence the iterations are planned to adapt to the changes in the operational environment.

The application is based on a layered architecture with four distinguishable layers, presented in Figure 4.

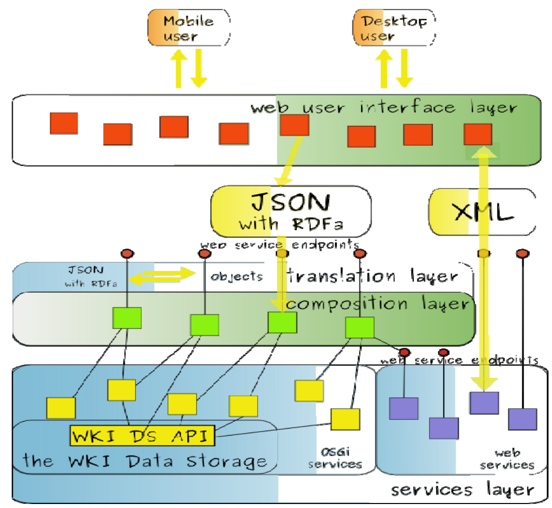

Figure 4. WeKnowIt architecture

The **Service Layer (SL)** gets together various types of intelligent services, and is responsible for all the computation and storage activities. The **Composition Layer** (CL) is responsible to receive the requests from the UI layer, pass the request to appropriate services of the Service Layer and prepare the response data. The **Translation Layer** (TL) translates the data returned by the Composition Layer to format required by UI components. Finally**,** the **User Interface Layer** (UI) is responsible for presentation of data to the end user.

Regarding the system design, the approach taken follows the theoretical guidelines for designing as an artifact and for design evaluation, as discussed in [13], aiming at producing viable artifacts in the form of instantiations after undergoing a series of evaluation tests regarding the utility, quality, and efficacy of each design artifact. Every artifact is created on a Continuous Integration server (after performing integration, functional and performance tests) and is deployed to the repository. Build reports are available for inspection via web interface of the server, and only artifacts created by builds performed on the server and stored in the repository are allowed to be deployed on staging and production environments.

In the Mobile Guidance application, the collective travel services presented in the previous section (**SearchPlaces, GetPOIs, InformationPlace, InformationPOIs**) and provided by the **Service Layer**, are accessed directly by the **User Interface Layer** due to the fact that they are exposed as web service endpoints. Mobile Guidance uses this method to eliminate some complexity produced by the aggregation of services and translation of the services output done internally by some WeKnowIt layers and which are not currently needed.

When a user wants to use the mobile application to find out more about a city, its POIs, or to request recommendations, he first logs on the Mobile Guidance application. The mobile access functionality is controlled by a login mechanism, which identifies the user, using the login/password that the user provides, or using an identification service via OpenID [14]. Afterwards, the user queries are performed through the **User Interface Layer**. The **User Interface Layer** receives the Ajax requests from users, processes them and calls the appropriate WeKnowIt service via REST to resolve the user request. The called WeKnowIt service returns an XML file with the results, which is then processed by the User Interface Layer and translated to JSON format, so that it can be suitably presented back to users.

When the user wants to search information for a place, e.g. Barcelona, he enters a text string with the name of the place in the search box of any Mobile Guidance application page (Figure 3. a) The **User Interface Layer**, first gathers from the text string the name of place to search for, and then forwards the request to the WeKnowIt **SearchPlaces** service via REST. This service searches effectively for places and returns an XML file with the information about the Places which matches the query (0-a). The **User Interface** collects this information and processes it to check if the geo-position of the user has the same longitude and latitude as any result individual place. In case that the **User Interface** doesn´t find any matches, the places data on the XML are translated to JSON and are forwarded to the **mobile device interface**. In the **mobile device interface** a list of the places names is presented to the user (Figure 3. a) and then the geographic positions of these places are displayed on the map (got from an external call to Yahoo! Maps API services [15]).

Place


```
- <result q="Barcelona" results="2" results_tot'
  - <place>
      <id>20</id>
      <woeid>753692</woeid>
      <name>Barcelona, Spain</name>
      <types>town</types>
      <longitude>2.170050</longitude>
      <latitude>41.385719</latitude>
    </place>
  - <place>
      <id>79795</id>
      <woeid>1162514</woeid>
      <name>Barcelona, Philippines</name>
      <types>town</types>
      <longitude>124.150002</longitude>
      <latitude>12.866700</latitude>
    </place>
  </result>
```


(a)

POI


```
– <place>
    <id>20</id>
    <woeid>753692</woeid>
    <wikipediaid>4443</wikipediaid>
    <name>Barcelona, Spain</name>
    <types>town</types>
    <wikipediatitle>Barcelona</wikipediatitle>
    <longitude>2.170050</longitude>
    <latitude>41.385719</latitude>
  </place>
– <pois>
  – <poi>
      <id>122644</id>
      <woeid/>
      <wikipediaid>4115942</wikipediaid>
      <name>Arc de Triomf</name>
      <types/>
      <wikipediatitle>Arc de Triomf</wikipedi
      <longitude>2.18056</longitude>
      <latitude>41.3908</latitude>
      <score>1</score>
```


(b)

Figure 5. Outcome of SearchPlaces (a) and GetPOIs (b)

In case the **User Interface** finds out that the position of one place of the XML File of Places is the same as the user´s geo-position, it will forward a new request to the WeKnowIt **GetPOIs** service to get the information about the POIS of the place where the user is currently located. The **GetPOIs** service, which retrieves the information applied for, returns an XML file (Figure 5. b) with an ordered list of POIS recommended and with its scores in the range from 0-1, classified and scored by the WeKnowIt mass intelligence layer, according to the mass trends. Finally the XML file with the recommended POIs is translated to JSON again and the User Interface Layer forwards it to the mobile device interface, where the results are shown to the user as a list of POIs with a number of stars signifying the strength of the recommendation. The locations of POIs are also shown on a map, so the users can see their location relative to other POIs (Figure 3. b). The Mobile Guidance application also shows the physical address corresponding to the user GPS coordinates using the GClientGeocoder API for the reverse translation [16].

The Mobile Guidance application is developed for two different types of mobile devices (Android and iPhone). In order to make the integration of the WeKnowIt architecture more compatible with these types of mobile terminals, we have developed a special architecture for the User Interface (Figure 6.) which simplifies the procedure of getting the GPS location of the users. This architecture consists of two different interfaces for each mobile platform (one for iPhone and one for Android), so as to hide the details of implementation of each one to the **WeKnowIt User Interface Layer**. The **iPhone interface** consists of a JavaScript Interface that makes use of the W3C Geolocation **API implementation** [17], which is supported by version 4 of the Safari browser. For Android, we developed a Java native interface based on the **WebChromeClient** class of Android API (v1.5) [18], which makes use of **LocationManage**r object to get user's position from the Android app (Figure 7.). Both interfaces (Android and IPhone) set up communication with the **WeKnowIt User Interface Layer** using a **JavaScript channel** to forward the requests made by the user with his mobile. This way, the WeKnowIt User Interface Layer is capable of processing the requests got from the JavaScript channel in a device-independent way.

VI. EVALUATION AND RESULTS IN CONSUMER SOCIAL GROUP APPLICATION

The Mobile Guidance application that integrates the Collective Intelligence services described in Section IV has been evaluated within its first implementation cycle.

Two different evaluators' teams performed participatory evaluations in order to assess the features of the prototype and find areas of improvement.

The first evaluation team consisted of 4 members with joint experience in research, system design and architecture, and

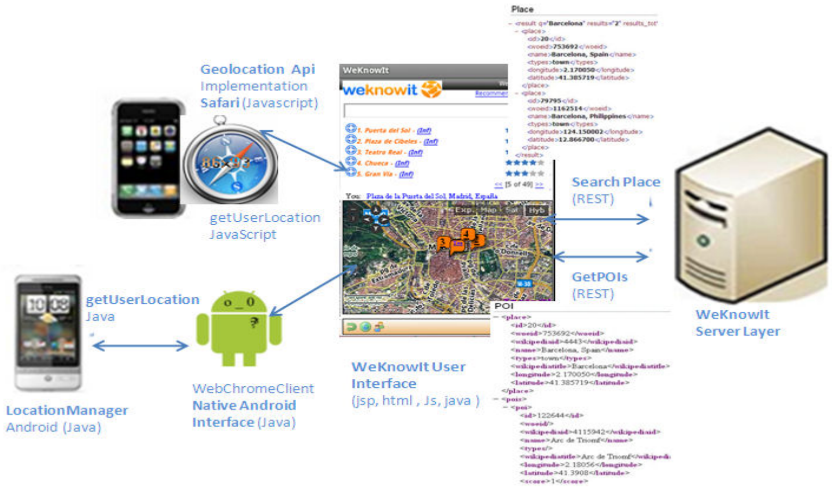

Figure 6. Mobile Guidance tool: User Interface Architecture

project management. They held a field evaluation experiment on 16 December 2009 in Madrid (Spain). Their feedback was the outcome of a consensus meeting.

The second evaluation team was formed by 22 people contributing to the WeKnowIt Project in a field evaluation experiment held on 21 January 2010 in Barcelona (Spain). Their feedback was collected by means of a questionnaire.

The evaluation addressed the following user needs: touristic information and recommendations, field navigation guidance and group communication, which were supported by Collective Intelligence automatically extracted results.

The items of functionality considered for evaluation were: get recommendations of Points of interest; search for places; search for Points of Interest; search for friends; get detailed information about places; get detailed information about

Figure 7. Personal Recommendations using Android

Points of interests; get image gallery about Points of interest.

The following dimensions were explored in the evaluation: availability, usefulness, usability, simplicity, appearance, responsiveness, suitability, completion and contribution of Collective Intelligence to the user experience.

The evaluation outcome is displayed in Figure 8. Most dimensions received positive feedback, which means complete or partial agreement on that dimension.

Most evaluators were positive on the ability to exploit Collective Intelligence in the travel scenario using the Mobile Guidance application. The evaluators also found suitable the functionality made available in the first-iteration prototype, at the same time they clearly indicated that the application has great improvement potential, and suggested ideas for further development in several areas:

- **Route:** Find the optimal path to a Point of interest; include information about public transport, such as subway map or relevant buses. Visualize routes, what you have done, what is ahead. Allow pictures taken to be displayed in the route. Export your route in graph format.
- **Points of interest:** improved search of POIs; list of nearby POIs, and attractions; information and news about POIs such as opening hours, distances to POIs; include kinds of POIs such as “where to take a break”, “where to lodge”, etc.
- **Recommendations:** improved recommendations based on the route; recommendation of events and recommendations for groups.
- **Group communication:** broadcast messages to your travelers group; broadcast emergency alerts.

The WeKnowIt consortium is considering the outcome of the evaluation in the second implementation cycle.

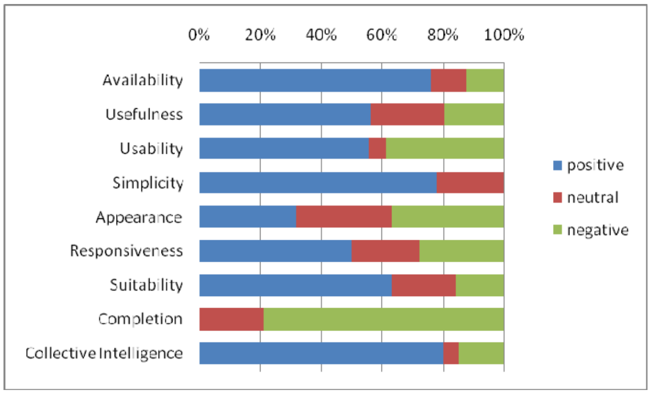

Figure 8. Evaluation outcome

VII. CONCLUSIONS

We have presented a Collective Intelligence enabled mobile guidance application which exploits and combines various intelligent services, supporting functionalities such as POIs recommendation, POI extraction from Wikipedia and GeoPlanet, search and retrieval capabilities for POIs and places, as well as friend localization on a map. Field test evaluations of this first version has designated the need for advanced analysis services in mobile consumer social applications, as well as the potential of the tool for integrating even more intelligent features.

ACKNOWLEDGMENT

The work presented in this paper was supported by the European Commission under contract FP7-215453 WeKnowIt.

REFERENCES

[1] L. S. Kennedy, M. Naaman, S. Ahern, R. Nair, T. Rattenbury, "How flickr helps us make sense of the world: context and content in community-contributed media collections," ACM Multimedia, 2007, pp. 631-640, http://doi.acm.org/10.1145/1291233.1291384

[2] L. Winerman, "Crisis communication," Nature, vol. 457, January 2009, pp. 376-378, doi:10.1038/457376a

[3] T. Yoshida, S. Luo, H. Xia and Z. Wang, "Toward collective intelligence of online communities: A primitive conceptual model," Journal of Science and System Engineering, vol. 18(2), 2009, pp. 203–221.

[4] E. Giannakidou, V. Koutsonikola, A. Vakali and I. Kompatsiaris, "Co-clustering Tags and Social Data Sources," In Proc. 9th International Conference On Web-Age Information Management (WAIM' 2008), IEEE Computer Society, Zhangjiajie, China, July 2008, pp 317-324.

[5] L. Specia and E. Motta, "Integrating folksonomies with the semantic web," Proc. of the 4th European Semantic Web Conf., (ESWC '07), June 2007, pp. 624–639

[6] TripIt – Travel Itinerary – Travel planner. Last retrieved Apr. 2010. http://www.tripit.com/.

[7] A. Scherp, and S. Boll, "Generic support for personalized mobile multimedia tourist applications," In Proceedings of the 12th Annual ACM international Conference on Multimedia. MULTIMEDIA '04. ACM, New York, NY, pp. 178-179.

[8] P. Hertzog, and M. Torrens, "Context-aware mobile assistants for optimal interaction: a prototype for supporting the business traveler," In Proceedings of the 9th international Conference on intelligent User interfaces (Funchal, Madeira, Portugal, January 13 - 16, 2004). IUI '04. ACM, New York, NY, pp. 256-258

[9] E. Rubegni, S. Gerardi, and M. Caporali, "Mobile applications for helping users to keep track of their travel experience," In Proceedings of the 14th European Conference on Cognitive Ergonomics: invent! Explore! (London, United Kingdom, August 28 - 31, 2007). ECCE '07, vol. 250. ACM, New York, NY, pp. 311-312

[10] J. G. Walls, G. R. Widmeyer, and O.A. El Sawy, "Building an Information System Design Theory for Vigilant EIS," Information Systems Research, Vol 3(1), March 1992, pp. 36-59.

[11] R. van Zwol, B. Sigurbjörnsson, "Faceted Exploration of Image Search Results," In WWW'10: Proceedings of the 19th international conference on World Wide Web. Raleigh, NC, USA, 2010, in press.

[12] WeKnowIt Project– Emerging, Collective Intelligence for personal, organisational and social use http://www.weknowit.eu

[13] A.R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in Information Systems research," MIS Quarterly, vol. 28(1), pp. 75-105.

[14] Open Id: Last retrieved Jan 2010. http://openid.net/

[15] Yahoo! Maps: http://maps.yahoo.com/ . Last retrieved Dec. 2009

[16] CGeocodeClient Rerveser. Last retrieved Dec. 2009. http://code.google.com/apis/maps/

[17] Geolocation Api (W3c). Last retrieved Dec. 2009. http://dev.w3.org/geo/api/spec-source.html.

[18] Android location manager. Last retrieved Dec. 2009. http://developer.android.com/reference/android/.