Semantic Web and Augmented Reality for searching people, events and points of interest within of a University Campus

David Chimbo*, Pablo Contreras*, Andrés Tello†, and Mauricio Espinoza†
*School of Systems Engineering, University of Cuenca, Ecuador
{david.chimbov, andres.contreras08}@ucuenca.ec
†Computer Science Department, University of Cuenca, Ecuador
{andres.tello, mauricio.espinoza}@ucuenca.edu.ec

***Abstract*—The advance of technology makes that mobile devices have gained widespread popularity. Modern mobile devices include build in sensors, cameras, compasses, and enhanced storage and processing capabilities, which allow developers to use those features to create applications with new or enhanced functionality. In this context we present a mobile application for searching places, people an events within a university campus. In our work we leverage semantic web and augmented reality to provide an application with a high degree of query expressiveness and an enhanced user experience. In addition, we validate our approach with a use case example that shows the complete searching process.**

I. Introduction

Mobile devices (e.g., tablets, smartphones) have become essential technological tools because of their use in several daily activities. This situation is caused by the fast technological development which allows to build more robust and reliable devices with features that include different types of sensors, cameras, and enhanced storage and processing capabilities. Such features allow that new technologies may be exploited in the context of mobile devices. Thereby, allowing mobile applications to include new or enhanced functionality [1]. Semantic Web and Augmented Reality are two of those technologies exploited in this context with promising results. In this work we leverage those technologies to build an interactive mobile app to improve the searching and localization of points of interest. Particularly, we are focused on solving the problem of searching and localize people, places, or events within a university campus.

Semantic technologies allow modeling and storing information with a high level of expressiveness, enhancing the searching process [2]. This is an advantage to the traditional database systems which has limitations to deliver significant and relevant results to the end users [3]. On the other hand, Augmented Reality is a technology that combines the physical environment with virtual environments created by a computer in real time. Thus, enhancing the users' experience of making activities in a specific physical context [4], for instance, localizing places or people in a specific area.

Today, there are many applications of these technologies separately in different areas of study (e.g., turism, education, commerce, etc.) with excellent results [5]. In the last years, there are several applications to facilitate the searching and localization of hotels, restaurants, etc., through the visualization of geo-localized multimedia content [6]. Some prominent Augmented Reality Navigation Systems are Layar[1] and Wikitude[2]. Although these systems allow users to specify which elements they want to search and visualize, the information retrieval techniques are based on syntactic searching. In this work we present a similar application but focused on a specific domain, adding an ontological model to represent the data within the domain and using semantic web techniques to retrieve information. Thereby, the user delegates the extraction and classification of relevant information to the computer since data is represented in a formal language understandable by humans and computers.

We believe that semantic search systems complement augmented reality applications which goal is to find and visualize the location of people, places, events, and other elements. The integration of semantic based techniques improves the searching process because there is no need that a query provided by the users syntactically matches the data stored in the database.

The remainder of this document is as follows. Section II presents the related work in this area. In section III, we describe the architecture of the proposed system. Section IV shows a detailed description of the three main phases of the searching process (i.e., indexing, searching, and ranking). In section V, we describe the visualization process of the results using augmented reality. In section VI, we show an application example of our system. Finally, in section VII we present the conclusions of our work.

II. Related Work

In this section we present previous work in the field of Augmented Reality, Semantic Web, and those which combine both areas.

[1]https://www.layar.com/
[2]https://www.wikitude.com/

A. Augmented Reality

In this field there are several works focused on outdoor navigation. Matuszka and Attila, in [7], present a system which provides a guided navigation within the Kerepesi cemetery in Hungary. It shows the graves of famous characters and the information related to the visited grave. Ortiz, in [8], depicts a mobile application to show information about specific places within the Polytechnic University of Valencia, guided by geo-localization. This application allows registered users to create content which can be visualized by other users of the system. These kind of systems which combine geo-localization with augmented reality allow to obtain detailed information of specific places, enhancing the user experience.

The features included in the aforementioned systems make augmented reality suitable for tourism applications because they provide detailed information about places of interest, and they may suggest places based on user preferences. The latter feature is important because many applications are limited to provide suggestions to a general group of users, ignoring the fact that there are different group of users with different likes and personalities. Leiva, in [9], describes an augmented reality system for tourism which gives suggestions based on users profiles.

Zhindón and Martín, in [10] state that one disadvantage of traditional navigation systems is that they show information in an abstract way, e.g., digital maps. Hence, the user need to interpret the maps to translate the indications showed in the graphic into real world locations. The augmented reality is an effective way to show detailed information of a specific point of interest by automatically detecting the user's location, the device orientation angle and the pointing direction.

Quevedo says that augmented reality may be used in more technical domains [11]. He proposes the use of this technology together with geo-localization to allow EMAPAL-EP company to localize components of the water supply system. Thus, avoiding leak reparation delays due to an improved and faster seeking of damaged components.

These examples show that augmented reality is a suitable technology for facilitating navigation and localization of specific elements within a particular area. Thereby, it justifies the use of augmented reality in this work.

B. Semantic Search

The core within the semantic search process is the knowledge database, which is represented through ontologies. An ontology is a formal definition which includes concepts, properties, restrictions, and relationships between entities within a specific domain. Hence, an ontology within a specific domain define non ambiguous semantic concepts.

Tello states that an ontology define concepts and relationships of some domain, in a shared and consensual way, and that such conceptualization must be represented formally, legible, and usable by computers [12].

During the last years several research groups have used semantic search in different domains. This technology contributes to obtain more precise search results and to provide mechanisms for query disambiguation [13].

There are different types of semantic search systems and they can be classified in multiple ways. Fernández et al. propose a categorization based on the way the users express their queries [14]. The types of queries according Fernández et al. are search by keywords, natural language queries, controlled natural language queries, structured queries based on ontology query languages.

From these types of systems, the searches by keyword and the natural language queries are the most popular because they are user friendlier regarding the interaction with the user [15]. In this work we opted by keyword based queries, represented by a formal language. The use of keywords reduce the complexity to the user, specially taking into account the limited space in mobile applications. In addition, this type of queries require less time to write compared with other alternatives.

The aim of our proposed search engine is to generate and choose from a set of formal queries, i.e., queries are defined in a formal ontology query language derived from a set of keywords.

Lei et al., in [16], present one of the first systems of this type. In their work, the keywords typed by the user are translated to queries in a formal language, which are executed to obtain the final results. First the system tries to understand the semantic meaning of the keywords. To do so, it performs a matching process, or mapping, between the keywords and the resources represented within the ontology (e.g., concepts, instances, properties). Then, using predefined templates, the system translates the set of semantic resources into a set of queries in a formal language. Next, it executes the queries against the data repository obtaining partial results which are sorted based on parameters that indicate the degree of similarity with the intended user search. One limitation of this work is that the user must know the data schema in order to match the keywords with the results of the ontology. Moreover, the system requires the user to indicate to which class or concept the keyword he is searching belongs to. In our work we use templates similar to those used in [16]. They allow creating complex queries based on simple rules between two semantic entities defined in the templates. This way, we can control the relations to be considered when creating the queries.

Mukhopadhyay et al., in [17], present a semantic search system which attempt to produce exact results. The system bases its search on identifying semantic relations. First, it extract the relevant terms from the user query which correspond to classes or instances in the ontology. Then, it finds relations among those terms by means of the axioms *range* and *domain* in the properties of the ontology. If the system does not find a direct relation through *range* and *domain*, it extends the search through the ontology tree. Once it finds the properties which match with the terms it can retrieve an instance by means of a mapping process. In such process, the system also consider class synonyms which allows including more terms than those originally included in the ontology. One limitation of this work is that aiming to deliver just one result, the system might avoid other valid interpretations of the query.

There are other systems [18], [14] where the input are queries expressed in natural language. Such queries have a more complex structured than those based on keywords. Hence, to translate them to queries based on formal languages (e.g., SPARQL[3]) we need to use specific libraries or frameworks (e.g., PowerAqua[4], QUEPY[5])

C. Augmented Reality combined with Semantic Technologies

There are several works, specially in the context of mobile applications, which combines the capabilities of both, semantic technologies and augmented reality. Tamás and Attila, in [7], described in section II-A, present an important contribution of this type of systems. Similar approaches include [19], [20] which aim is to link physical places, objects and people to digital content. These applications leverage the Open Link Datasets to extract relevant information of a particular context which is displayed, later on, using augmented reality. There are other projects related to other areas besides tourism. For instance, Hervás et al., in [21], present a mobile application which support users daily activities through augmented reality. The app provides customized information of specific elements from an environment. The architecture presented in this work combines augmented reality and ontologies. After evaluation tests, this system obtained high scores in user satisfaction. The evaluation tests included usability and degree of customization of the information delivered.

Most of the approaches described before follow a client-server architecture. Thus, the server supports the storage of the knowledge database and the semantic processing, reasoning and inference; while the mobile application is in charge, only, of the user interaction. On the other hand, there are projects (e.g., [22], [23]) that try to adapt some tools in order to process the semantic information directly on the mobile devices. However, as stated in those works, it requires a large additional implementation effort. In addition, the performance is poor compared with those client-server systems. In our approach, we opted for a client-server architecture to avoid overloading the mobile device with processing tasks and there by do not reduce the performance of the application.

III. System Description

In this section we show the architecture of the proposed system. In addition, we describe the details of each component and its function within the searching process since a user performs a query until he receives the results of his search.

[3]https://www.w3.org/TR/sparql11-query/
[4]http://technologies.kmi.open.ac.uk/poweraqua/
[5]http://quepy.machinalis.com/about.html

A. System Architecture

Our systems follows the client-server architecture as depicted in figure 1. The mobile application is the client, which main function is the interaction of the user with the system. It allows users to define queries and the subsequent visualization of its results. In the client side we used elements from augmented reality which facilitates the user's interaction with the system. On the other hand, at the server side, the system performs the searching process from the keywords proposed by the user.

As shown in the figure 1, the system architecture includes five main components.

1) The Semantic Model (Ontology).
2) The Indexing component.
3) The Query Creation and Query Execution component.
4) The Results Classification and Ranking component.
5) The Visualization with Augmented Reality component.

The first four components are implemented at the server-side, while the last one is at the client-side through the mobile application.

In the next section we describe the role of each component within the entire searching process.

B. Description of the System Operation

The **Semantic Model** is the core of the system because it represents and specifies, formally, the information about the points of interest, events, and people within the context of the University of Cuenca. Given it is a semantic model, it does not only store data and concepts but relations and restrictions on them. Such information allows performing the semantic searches at a later stage. The semantic model is the University of Cuenca Ontology. We create this ontology following the NeOn methodology proposed by Suárez-Figueroa in [24]. A detailed description of the ontology creation process is presented in [25].

In addition, besides the semantic model, the **Indexing** component creates an index of the semantic entities described in the ontology. Such index is used by the **Query Creation and Query Execution** component which takes as input the keywords entered by the user and returns a set of semantic entities as possibles matches of the query. This process is described as follows.

- The first step is the **recognition of the semantic entities**. This process takes as input the keywords provided by the user and returns a set of semantic entities. This set is created by matching each keyword with one or several concepts, instances or properties within the ontology. Such matching uses the previously created index by the Indexing component.
- The formal query generation takes each element from the semantic entity sets and group them. Thus, each group has a combination that represents a possible interpretation of the keywords entered by the user.
- From this point, each searching group is analyzed independently as follows.

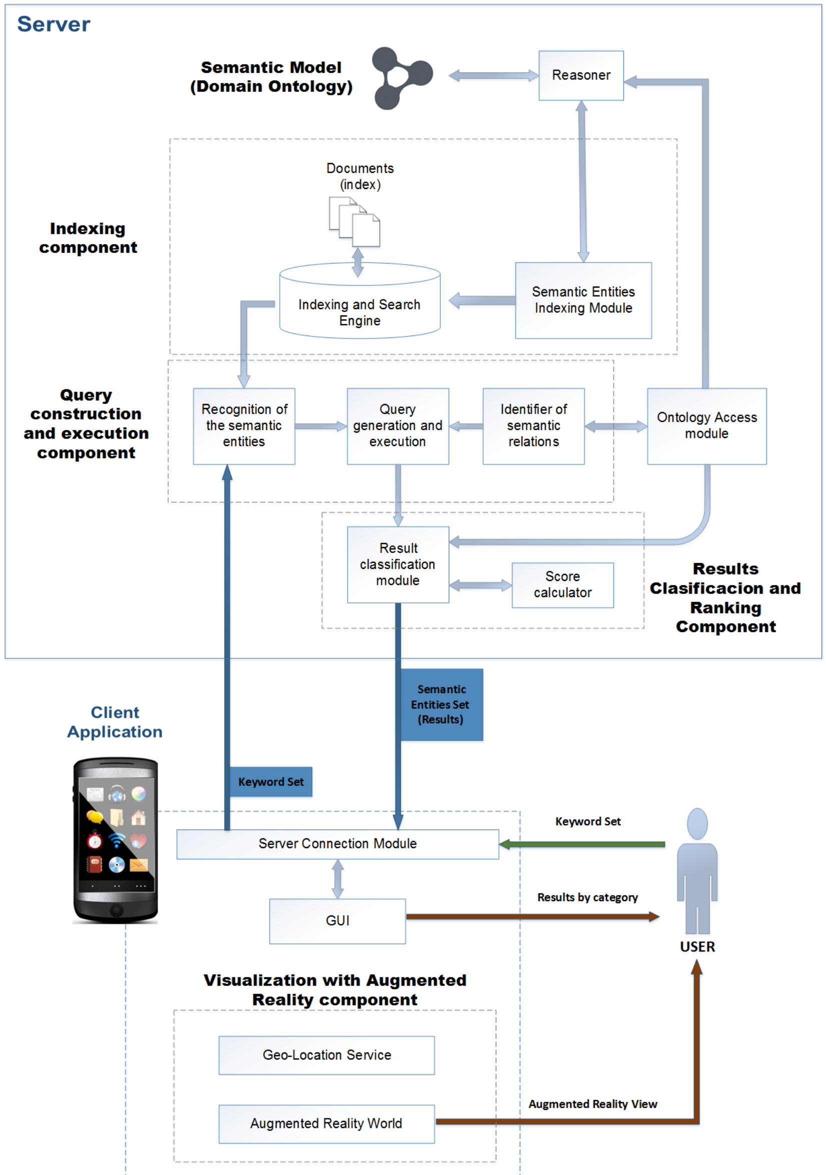

Figure 1. System Architecture

- For each group, the first step is to **identify the main concepts of the search**. i.e., to which class the element searched by the user belong to.
- The next step is to **identify semantic relations** to find relevant relations among every element within the searching group.
- Then, the system creates the query graphs taking the main concept as root node, and from it, it tries to link the other semantic entities of the group through the semantic relations identified in the previous step.
- Once all the elements within a group are linked, the next step is the **creation of the formal queries**, in SPARQL language, using a set of templates. Each of these queries is a possible representation of the user intended search.

• Finally, the **Query Execution** component allows executing the queries generated in the previous step. The results are a set of instances which represent the possible response to the query provided by the user.

Then, the **Results Classification and Ranking** component takes the set of instances obtained in the previous step and classifies them hierarchically, grouping the instances that belong to the same concept, according to the concepts within the ontology. In addition, at this step the results are sorted according a parameter which indicates the degree of relevance of the results with respect to the query provided by the user. Finally, a data structure which contains the results, categorized and sorted, is sent to the mobile device.

The idea behind this data structure of the results is that the user can navigate through them following the hierarchy of concepts within the ontology, starting from the most general concepts to the most specific ones. Once the results are on the mobile application, the user can visualize the location of each of them through an Augmented Reality View generated by the **Visualization with Augmented Reality** component. This component takes an image in real time using the camera of the mobile device and add tags to it with a description or name of each result and its location. Hence, the user can visualize, on the mobile screen, each result of his search, as long as, it is within the vision range of the camera.

IV. Semantic Search Process

The semantic process involves different steps as described in section III. In this section we preset a detailed description of each phase during the semantic search process.

A. Semantic Entities Indexing

The first step during the searching process is to identify the semantic entities to which the user refers to with each keyword provided. To achieve this task we need to create an index of semantic entities from the information within the ontology. Such index stores relevant data of each semantic entity with the aim of easing its identification and retrieval at a later stage. Hence, prior to create the index we need a populated ontology. The index creation is executed the first time we load data on the system, and every time we need to update or modify the exiting information.

To determine which information to include in the index we annotate within the ontology the relevant properties of every concept or instance as suggested in in [26]. Thus, we avoid the index is overloaded with trivial information. Specifically, the index registers an identifier and two additional fields for each semantic entity within the ontology. The identifier is the semantic entity URI. The indexing and search engine return this identifier when it determines that a keyword corresponds to a semantic entity registered in the index. The first additional descriptive field is the semantic entity *type*, i.e., concept, instance, or property. The second field is the *name* or *description* of the semantic entity defined as a annotation in the ontology, e.g., "label@es". Later,the indexing and search engine use this information to create the mapping between keywords and semantic entities. The index for the "Instance" semantic entities also store the value of the property. We use the annotation "anotIndexacion" the properties of the relevant data for each concept. In a similar way, the system uses annotations to determine which information is to be displayed. This is explained in the subsection IV-D

Since each keyword provided by the user may link many semantic entities, we create different "searching groups" with one interpretation of each keyword at a time. Therefore, at the end we create as many groups as required to include all possible combinations of keyword interpretations.

B. Formal query construction

The component intended to create the formal queries takes as input the "searching groups" identified in the previous step and deliver one or more queries in SPARQL language. Each of those queries represent a different interpretation of the intended user query.

First we need to identify which is the main concept of the search, i.e., to which class the instances the user is searching belong to. Since our application is focused on finding places, people, or events within the University of Cuenca, the set of central concepts are limited by the concepts "Place", "Person", "Event", and all their subclasses. Hence, to determine the main concept we search within the group of semantic entities whether there is a match with one of the possible searching concepts. If we find more than one central concept, then we perform an independent query construction process for each of them.

Once we identify the main searching concept, the next step is to identify whether the rest of the semantic entities of the set can be related in some way with such central concept. We defined a set of rules to determine whether there is relation or not between two semantic entities. Given two semantic entities, such rules specify the conditions that need to be met to determine a valid relation between them (See Table I). Thereby, taking combinations of two entities it is possible to relate all the semantic entities from the set as long as such relation is valid. To achieve such entity pairing we rely on graphs, where nodes represent entities and edges are valid relations. The main node in the graph represents the central concept, and from it the rest of semantic entities; but taking into account that there must be a valid relation to link two entities in the graph. At the end, if we can obtain a graph with all the semantic entities connected through valid relations, then the graph is considered as a valid graph. Such graph is the base to create the query in a formal language, i.e., SPARQL. To create the formal query we use templates which determine the SPARQL sentences to be created and linked based on the relation established before. Table II shows the SPARQL sentences defined for each type of relation.

C. Results Ranking

The formal query construction process may produce too many queries to be executed. The main limitation of this approach is that the number of results might be overly large. Hence, we propose a mechanism that allows sorting the searching results according to some relevance with respect to the keywords provided by the user, i.e., *results ranking.*

The results ranking construction relies on an important characteristic of ontologies, the "relations heterogeneity" [27]. This aspect refers to the fact that in a ontology every entity is related with other semantic entities through different types of relations, where each relation has a different importance or relevance for that particular entity.

The first step to achieve such ranking is to take every concept within the ontology and all its properties, which might be instance or object properties. Then, using annotations we assign a numeric value to them. This number represents the relevance degree of such property regarding that concept. In this work, we use arbitrary numbers to weight the relations but

Table I

RELATION VALIDATION RULES BETWEEN TWO SEMANTIC ENTITIES

Entity type	Relation Name	ValidationRule
Concept (C1) Concept (C2)	CC	If there are instances of Concept C1 which are related with one or many instances of Concept C2
Concept (C1) Instance (I1)	CI (or IC)	If Instance I1 is of type concept C1.
Concept (C1) Instance (I1)	CPI (o IPC)	If Instance I1 has some Property which relates it with some instance of Concept C1.
Concept (C1) Property (P1)	CP	If there is some sentence where Property P1 has,as a subject, an instance of concept C1.
Instance (I1) Instance (I2)	II	If Instance I1 has one or more relations with Instance I2.
Instance (I1) Property (P1)	IP	If there is some sentence where Property P1 has as a subject, to Instance I1.
Property (P1) Concept (C1)	PC	if there is some sentence where Property P1 has,as object, an instance of concept C1.
Property (P1) Instance (I1)	PI	If there is some sentence where property P1 has,as object, the instance I1.
Property Property	It is not considered because there must be a concept or instance which links the two properties.	

Table II

SPARQL SENTENCES FOR EACH IDENTIFIED RELATION

Relation Name	ValidationRule
CC	{?Instance_C1 ?property_C1_C2 ?Instance_C2.} UNION {?Instance_C2 ?property_C2_C1 ?Instance_C1.} ?Instance_C1 rdf:type <URI_C1>. ?Instance_C1 rdf:type <URI_C2>.
CI (or IC)	?Concept_C1 rdf:type <URI_C1>. FILTER (?Concept_C1 = <URI_I1>) .
CPI (o IPC)	{?Instance_C1 ?property_C1_I1 <URI_I1>.} UNION {<URI_I1>?property_C1_I1 ?Instance_C1.} ?Instance_C1 rdf:type <URI_C1>. FILTER (?property_C1_I1 != rdf:type) .
CP	Not implemented yet
II	Is not necessary to create a SPARQL statement for this relationship
IP	Is not necessary to create a SPARQL statement for this relationship
PC	Not implemented yet
PI	FILTER (?Instance_Range_P1 = <URI_I1>).

taking into account which properties has a higher relevance degree with respect to the concept. As a general rule, we assigned a higher weight to an instance property than an object one, because the first ones are direct descriptors of the semantic entity to which belong. Thereby, they have more importance.

The ranking of a particular result includes the partial value of each of the semantic entities within the searching group that generated such result. Since each of those entities should be related with the result through a property, the final ranking is the sum of the weights of each property.

D. Results Categorization

In addition to the results ranking, we propose a categorization to facilitate the navigation through the search results. The goal is that the user can navigate quickly and efficiently through the results set until he reaches the searched element. This mechanism relies on some properties and other information within the ontology as described next.

Our approach follows a hierarchical categorization. Hence, the user first has to choose a specific category to find the searched result. The system uses the concepts hierarchy tree from the ontology to define categories and subcategories. Thus, the user can navigate from general concepts until the most specific ones by navigating through such concept hierarchy. Since the results may be only one of the three general classes, i.e., People, Place, or Event, these classes define the three basic categories. From these categories we present to the user only the subcategories which are part of the results. This way we reduce the number of steps the user need to find the right category. Once the use select a specific category the system shows only the result that belong to that category. This approach avoids the screen to be overwhelmed with items, making easier for the user to find the searched elements.

Currently, the information presented within a result is nothing but the URI, which allows the user to locate the instance within the ontology. Consequently, before sending the results to the mobile phone, we retrieve from the ontology all the information to be shown for every instance. The instance and object properties represent such information within the ontology. Since some of these properties might not be relevant enough to be shown to the user, first we define which properties are relevant for each concept by means of annotations as shown in Table III. Hence, when visualizing the instances of a specific concept, we only show the values of the properties labeled with such annotations.

Table III

DATA PROPERTIES ANNOTATIONS

Annotation	Description
anotDescripcion	The information of each semantic entity to be shown to the user in the general results list.
anotInfPrincipal	Both include the detailed information of a specific semantic entity to be shown to the user .
anotInfSecundaria	

V. VISUALIZATION WITH AUGMENTED REALITY

The last stage on the system is the presentation of results to the user through a mobile application. The goal is to show the physical location of the results, which may be people, places or events, through augmented reality. In this section we describe the flow of the mobile app from the moment the user provides the keywords for the search until the results are shown.

First, the application shows to the user the searching window where he must provide the keywords to search. Next, the app calls the semantic web searching service providing the keywords as parameters. Then, the web service processes the request and generates four map data structures, which are the service response. The maps delivered by the service are described as follows.

- **Concept Classification Map.** For each pair in this map the key is the URI of the main concept, and the value is a list of its corresponding sub-concepts.
- **Concepts Index Map.** For each pair in this map the key is the URI of the concept, and the value is an object with the data which belongs to this concept.
- **Instance Map by Concept.** For each pair in this map the key is the URI of the concept, and the value is a list of instances that belong to that concept.
- **Instances Index Map.** For each pair in this map the key is the URI of the instance, and the value is an object with the information of the instance to be visualized. When available, this information includes the geospatial coordinates.

Once the mobile phone receives the information this is stored locally, thereby, avoiding to make additional server calls. With the information available locally at the mobile application, the user can select the searched result within the complete results set. As mentioned before we propose a mechanism to facilitate the user to navigate through the results by grouping them into categories. The next is a detailed description of this process.

1) Using the Concepts Classification Map, the application shows the main categories (People, Place, or Event) window to the user (See figure 2a).
   In addition, there is the option of showing with augmented reality the locations of the results (when available). To do this, we need to include geospatial information (i.e., latitude, longitude, altitude) within the instance's data. The visualization with augmented reality (see figure 3) includes a label which contains the description of each instance at a specified location.
   Moreover, we included a radar to visualize the instance location regarding the user's current position. We also included an option to limit the range of visualized instances (see figure 4). Hence, the application only shows the labels of the instances which are within the range (distance in meters), taking as a center the user's location.

Figure 2. Overview of the application operation

2) Once the user selects the main category, the next step is to show a list of sub-categories (see figure 2b). If there are more categories under a sub-category, the system shows the sub-categories at the lowest level. This approach avoids the user to select again a sub-category in the next step.

Figure 3. Points of interest using augmented reality

Figure 4. Visualization range specification

3) When the user selects a specific subcategory, using the Instances by Concept Map, the application displays a list with the instances that belong to that category (see figure 2c).
4) Finally, when the user selects an specific instance from the previous step, the system, using the Instances Index Map, shows the detailed information of that instance (see figure 5). The same as in the step 1, here we also include the option of showing the results location in a static 2D map, or using augmented reality. Additionally, in the augmented reality view, we include an arrow, pointing the instance location to facilitate user orientation.

VI. System Application Example

In order to evaluate the proposed system, in this section we describe the following use case[6]: let us imagine a student who is looking for teachers who know about Web technologies. One way to express this query through keywords could be "Web Professor" (Profesor Web). These keywords are the input to the system to perform the search of semantic entities as described below:

A. Semantic Entities Indexing

The first step is to determine whether the two keywords entered correspond to one semantic entity or two different entities. This process is done taking as input the complete query "Web Professor" (Profesor Web), since in the index there is no record containing the two words, then the system

[6]Along this example, the parenthesis symbol will be used to represent the Spanish language terms, because the ontology terms were created for this language

Figure 5. Results detailed information

separates the query into two keywords, in this case “Professor” (Profesor) and “Web” (Web). For each of these keywords the identification process of semantic entities is performed. For the keyword “Professor” (Profesor) the system detects only the concept “Professor” (Profesor), while that for the keyword “Web”, 6 instances are identified as shown in Figure 6a. From these two sets, the search groups are created combining each element of the first set with each element of the second set. In Figure 6b, shows the groups obtained from these combinations.

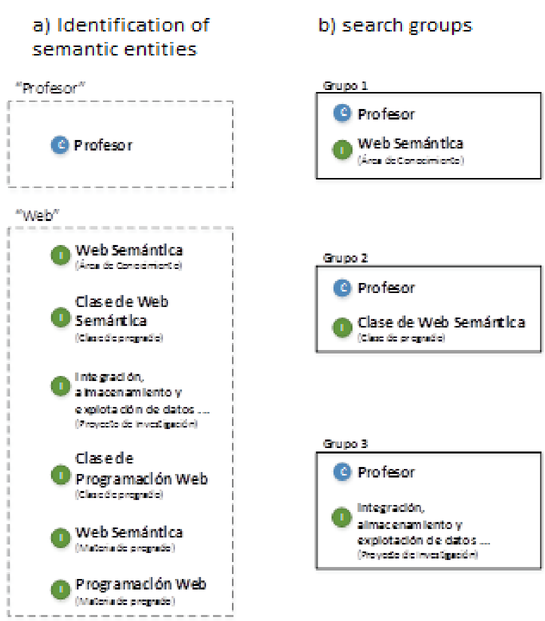

Figure 6. Graphic representation of the Use Case

B. Formal query construction

In this step the system tries to identify if there are semantic entities in the search group that correspond to the concepts Person, Place or Activity, or of any of its subclasses. If so, these entities are considered as search central concepts and the system attempts to construct a formal query for each of the concepts identified. The central concept determines of what type or class will be the results to be obtained at the time of perform the formal query.

As example, for the first search group conformed by the concept “Professor” (Profesor) and by the instance “Semantic Web” (Web Semántica) of the type “Knowledge Area” (Area de Conocimiento), the central concept is the class “Professor”, but it was not possible to find valid relationships between these two entities according rules defined in Table I from section IV-B. In the second group, the central concept is “Professor”, but also a CPI type relationship was identified (as defined in table I from section IV-B), between the “Professor” concept and the “Semantic Web Class” (Clase de Web Semántica), then a valid graph is considered and the query is created using the template corresponding to the CPI relation, according to Table II from section IV-B. The resulting query is as follows:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT DISTINCT ?Profesor WHERE{
  {
    ?Profesor ?propiedadProfesorclase_preg_web_sem_1
    <http://www.ucuenca.edu.ec/ontology/appbusqueda#clase_preg_web_sem_1> .}
    UNION
    {<http://www.ucuenca.edu.ec/ontology/appbusqueda#clase_preg_web_sem_1>
    ?propiedadProfesorclase_preg_web_sem_1 ?Profesor .}

  ?Profesor rdf:type <http://www.ucuenca.edu.ec/ontology/appbusqueda#Profesor> .
  FILTER(?propiedadProfesorclase_preg_web_sem_1 != rdf:type).
}
```

Figure 7. Query construction for the first two groups.

The same process is used for the other groups, generating the corresponding queries. Figure 8, shows how the semantic entities are related for the first two search groups.

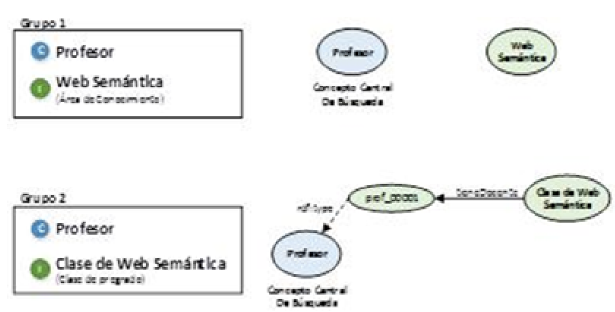

Figure 8. Query construction for the first two groups.

Once all the groups have been analyzed and all possible queries have been generated, the formal queries are executed. The results obtained for each of the generated queries were the instances: prof_00001, prof_00005, prof_00007, prof_00008, and prof_00009.

C. Results Ranking

To order the results, the weight assigned to the relationships involved in the construction of graphs is taken into consideration. For example, in the search group 2, the instance prof_00001, is linked to the instance “Semantic Web Class” (Clase de Web Semántica), through the “hasTeacher” (tieneProfesor) relation. On the other hand, in the search group 3, five results were obtained, which correspond to the “Professor” (Profesor) concept but in this case are linked to the instance “Project: Integration, Storage....” (Proyecto: Integración, Almacenamiento...) through of the “hasParticipant” (tieneParticipante) relation.

In the system was previously defined that the relationships between the “Professor” concept and the concept ”Class” (Clase) is more relevant than the relationships between the concepts “Professor” and ”“Project” (Proyecto) . Therefore, the instances linked with the relation “hasTeacher” have a higher score than the instances of the relation “hasParticipant”. In this case the concept prof_00001 will appear first in the result list because it has a higher score, while the remaining 5 results will appear after, without a particular order because them are part of the relation ”has participant”, so they have the same score.

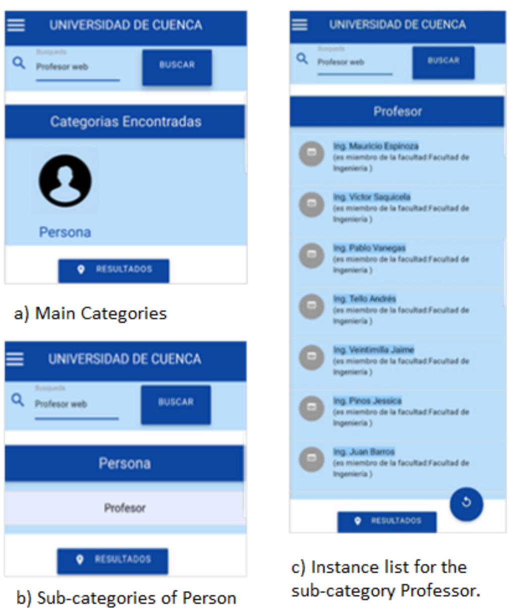

Figure 9. Mobile Screens with results.

1) Results Categorization: Because all results belong to the concept “Professor”, and this concept in turn is a sub-concept of “Person”, then the results area presented in this order: Person, Professor and all instances of this last concept. This information is extracted from the ontology from the annotations made in their entities.

2) Results Presentation: In the mobile application the main categories for which there are instances are presented, in this case only the Person category is displayed (see Figure 9a). Then, by selecting one of the main categories, all corresponding subcategories are listed (see Figure 9b). Similarly, by selecting one of the subcategories, the list of results (ontology instances) are displayed (see Figure 9c). The user can see the details of a specific result, selecting an element of the list, then a more complete description of the instance is showed (see Figure 10).

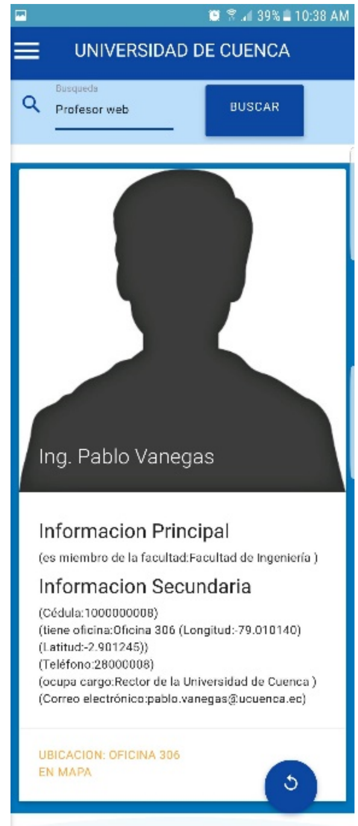

Figure 10. Detail screen of the result instance.

3) Visualization with Augmented Reality: Finally, the user has the option of to visualize the search results, through an augmented reality view. As mentioned in Section V, the augmented reality view offers certain visual elements that help the user to locate the required result. In Figure 11 we can see the augmented reality view generated for this particular example.

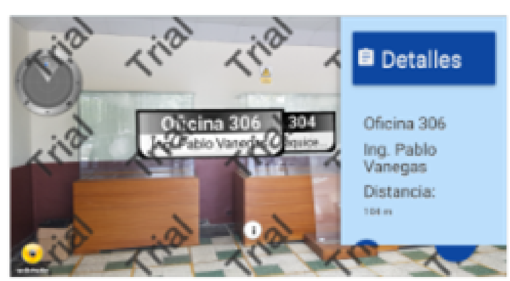

Figure 11. Augmented reality view.

VII. Conclusion

In this paper we presented a mobile application that combines semantic web technologies and augmented reality to allow users searching for places, people or events within a university campus. We presented the challenges that we encounter during the implementation of the three main phases of the searching process, i.e., indexing, searching, and results ranking. Our approach relies on a semantic web ontology to deliver more precise search results. In addition, the use of semantic web technologies gives the application the capability to cope with query disambiguation. Furthermore the use of augmented reality provides an enhanced user experience when visualizing the results on the mobile device. Finally, we presented a use case that allowed us to validate our approach.

One limitation of this work is that not all the formal queries expressed in SPARQL, as depicted in Table I from section IV-B, are implemented yet. In the future we plan to implement all the missing SPARQL sentences to complete the template for formal query construction.

References

[1] C. Te-Lien and C. Lih-Juan, "Augmented reality smartphone environment orientation application: A case study of the fu-jen university mobile campus touring system," *Procedia-Social and Behavioral Sciences*, vol. 46, pp. 410–416, 2012.

[2] M.-C. Carmen, B. I. J, and V. M. Amparo, "Ontologies versus relational databases: are they so different? a comparison," *Artificial Intelligence Review*, pp. 1–20, 2012.

[3] S. Sangjin, S. Dong-Hoon, K. Jihoon, S. Minjae, E. Sungkwang, L. Kyong-Ho, P. Jinsung, L. Seungwon, and J. Jooyeon, "Augmenting mobile search engine with semantic web and context awareness," in *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 03*. IEEE Computer Society, 2013, pp. 278–279.

[4] M. Tamás, "Augmented reality supported by semantic web technologies," in *Extended Semantic Web Conference*. Springer, 2013, pp. 682–686.

[5] K. C. D, K. A. E, Zamani, and E. D, "Enhancing the tourism experience through mobile augmented reality: Challenges and prospects," *International Journal of Engineering Business Management*, vol. 4, p. 10, 2012.

[6] G. Jens, L. Tobias, and G. Raphaël, "Augmented reality browser survey. tehnical report," *Institute for Computer Graphics and Vision, Graz University of Technology, Austria*, 2011.

[7] M. Tamás and K. Attila, "Alive cemeteries with augmented reality and semantic web technologies," *International Journal of Computer Information Science and Engineering*, vol. 8, pp. 32–36, 2014.

[8] F. M. Guillem Ortiz, "Upv-mobarguide: aplicación android de realidad aumentada para guía interactiva de la upv orientada a móviles," Ph.D. dissertation, 2012.

[9] J. L. Leiva Olivencia, "Realidad aumentada bajo tecnología móvil basada en el contexto aplicada a destinos turísticos," Ph.D. dissertation, Universidad de Málaga, 2014.

[10] M. Zhindón and G. Martín, "Implementación de un sistema de navegación con realidad aumentada basado en puntos conocidos para geo localización de puntos de interés," Master's thesis, Universidad del Azuay, 2014.

[11] Q. S. Andrés Sebastián, "Realidad aumentada en dispositivos móviles "android" aplicada a la geolocalización de equipamientos de agua potable de emapal-ep," Master's thesis, 2015.

[12] T. Adolfo Lozano, "Ontologías en la web semántica," *Jornadas de Ingeniería Web*, vol. 1, 2001.

[13] S. Kara, Ö. Alan, O. Sabuncu, S. Akpınar, N. K. Cicekli, and F. N. Alpaslan, "An ontology-based retrieval system using semantic indexing," *Information Systems*, vol. 37, no. 4, pp. 294–305, 2012.

[14] M. Fernández, I. Cantador, V. López, D. Vallet, P. Castells, and E. Motta, "Semantically enhanced information retrieval: An ontology-based approach," *Web semantics: Science, services and agents on the world wide web*, vol. 9, no. 4, pp. 434–452, 2011.

[15] S. S. L. Lau and P. M. Dew, "A survey of semantic keyword search approaches," *International Journal of Applied Computer Technology and Information Systems*, vol. 2, no. 2, 2016.

[16] Y. Lei, V. Uren, and E. Motta, "Semsearch: A search engine for the semantic web," in *International Conference on Knowledge Engineering and Knowledge Management*. Springer, 2006, pp. 238–245.

[17] D. Mukhopadhyay, A. Banik, S. Mukherjee, J. Bhattacharya, and Y.-C. Kim, "A domain specific ontology based semantic web search engine," *arXiv preprint arXiv:1102.0695*, 2011.

[18] R. Bansal and S. Chawla, "An approach for semantic information retrieval from ontology in computer science domain," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 4, no. 2, 2014.

[19] L. Nixon, J. Grubert, G. Reitmayr, and J. Scicluna, "Semantics enhancing augmented reality and making our reality smarter," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2012, pp. 863–870.

[20] C. van Aart, B. Wielinga, and W. R. van Hage, *Mobile Cultural Heritage Guide: Location-Aware Semantic Search*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 257–271. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-16438-5_18

[21] R. Hervás, J. Bravo, J. Fontecha, and V. Villarreal, "Achieving adaptive augmented reality through ontological context-awareness applied to aal scenarios." *J. UCS*, vol. 19, no. 9, pp. 1334–1349, 2013.

[22] L. Descamps-Vila, J. Misterio, J. Conesa Caralt, and A. Pérez Navarro, "Cómo introducir semántica en las aplicaciones sig móviles: expectativas, teoría y realidad," 2011.

[23] R. Yus, C. Bobed, G. Esteban, F. Bobillo, and E. Mena, "Android goes semantic: Dl reasoners on smartphones." in *Ore*. Citeseer, 2013, pp. 46–52.

[24] M. C. Suárez-Figueroa, "Neon methodology for building ontology networks: specification, scheduling and reuse," Ph.D. dissertation, Informatica, 2010.

[25] P. Contreras, D. Chimbo, and M. Espinoza, "Aplicaci ón de tecnolog ías sem ánticas y realidad aumentada para realizar b úsquedas de personas, puntos de inter és y actividades dentro del campus central de la universidad de cuenca," University of Cuenca, Undergraduate Thesis, 2017.

[26] C. Bobed, G. Esteban, and E. Mena, "Enabling keyword search on linked data repositories: An ontology-based approach," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 17, no. 1, pp. 67–77, 2013.

[27] V. Jindal, S. Bawa, and S. Batra, "A review of ranking approaches for semantic search on web," *Information Processing & Management*, vol. 50, no. 2, pp. 416–425, 2014.