DESIGN OF THE PHOTO ALBUM ONTOLOGY

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Mridula Sarker

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

September 2010

Fargo, North Dakota

North Dakota State University
Graduate School

Title

DESIGN OF THE PHOTO

ALBUM ONTOLOGY

By

MRIDULA SARKER

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

ABSTRACT

Mridula Sarker, M.S., Department of Computer Science, College of Science and Mathematics, North Dakota State University, September 2010. Design of the Photo Album Ontology. Major Professor: Dr. Juan Jen Li.

At present, growing demand of the management of huge volume of web contents, especially photos, require the software tools, which can store, index and retrieve the photos from the repository more efficiently. In this paper, we proposed a design solution using the semantic web technology. The goal is to integrate this knowledge-based ontology into the photo management system to improve the experience of the user's accessibility in the photo warehouse.

Most of the existing photo album management software tools are built using the convention of traditional approaches, such as key-word based search or low-level image annotation approaches, which make it difficult to retrieve the user's desired photo successfully.

To access those stored photos and retrieve them in an efficient manner, we designed an ontology-based photo album, which will allow the users to find their necessary information, in this case, photos effectively. Query language like Simple Protocol and Resource Query Language (SPARQL) is used to initiate the query within the photo repository. Our experience shows that this domain knowledge-based concept can improve the photo retrieval experience fruitfully.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my advisor Dr. Juan Jen Li who was an excellent mentor during my Master's paper study. Her guidance, support and help were an integral part of the completion of this paper. I feel very fortunate to be her student. Finally I would like to thank the faculty of the Computer Science Department and the members of my supervisory committee for their tremendous support during my study.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1. Motivation

Currently, searching and browsing of images is a growing demand, which requires potential automatic image annotation. This motivated me to design a knowledge-based photo-album system on the notion of ontology of semantic web technology. Ontology, which is an evolutionary step, is capable of sharing and reusing information by creating specification on particular domain concepts and represents the relationships that may exist among those concepts [1].

This paper depicts the process of designing a photo-album ontology, which will assist to store, index and retrieve images more efficiently. At present, advancement of technology and the low cost of available digital photo management tools allow users to accumulate a large number of personal images in the web album. But those available commercial photo management tools are not intelligent enough to store, index; and retrieve images from the intended photo album in an efficient manner [2]. Those existing software tools are suffering from a number of shortcomings, such as high recall-low precision, low or no-recall, etc. To overcome these shortcomings in regards to management of photos within a large photo album repository, this paper has introduced a new knowledge-based photo album concept design using the Semantic Web technology.

The proposed design facilitates automatically finding all photos using the idea of inference within the ontology-based photo album system. Since ontology supports a formal as well as conceptual annotation for an effective inference query, intelligent photo retrieval is possible by managing photo albums using the concept of ontology design. For example, most popular photo management, tagging and editing software, such as Picasa [3], Flickr, [4], Windows Live Photo Gallery [5], etc. support the keyword-based search. These software support merely isolated keywords, which suffer from lack of linkages to provide a more meaningful context. As a result, there are limited options to choose to get the exact match what users are looking for. Most of the time, the key-word based search enables users to get the result with high recall-low precision, low or no-recall, etc. It is also difficult for the software developers to collect or use numerous keywords for the correct annotation of images due to the lack of information in the labeled image databases. Current traditional keyword-based photo management tools allow human-supplied text labeling to annotate images which are used to facilitate searching the required images from huge digital photo libraries. As images are tagged using key words, search results will display images relevant and partially relevant to those words. In addition to this, if the search words are part of a folder name where users stored their images, the result will display all pictures regardless of what users requested. This distinguishing feature prohibits connecting data and users in an efficient manner. Moreover, different people may use different keywords for the same image, which the available techniques do not support and therefore, can't answer users' queries consistently and reliably. Besides, these manual text-based image annotations are fairly expensive for a large scale image data repository [1].

Since text-based image retrieval techniques involve a limited number of reserved words for image annotation, in most cases this approach suffers from imprecision, unreliability; and inconsistency in regards to response to queries initiated by users. Moreover, lack of expressiveness in key-word based image annotation has made it extensively difficult for users to use the more relevant textual clues to retrieve their intended images from numerous and various digital image databases [6]. Another popular photo management technology, Content-based image retrieval (CBIR) uses only low-level features such as texture, shape, color etc. for indexing and image retrieval. The goal of this image retrieval technology allows users to find an image or images within a large and various collection of image repository. CBIR involves extraction of basic low level features automatically by images themselves. This low-level based distinctive feature creates a problem for automated image retrieval which is called semantic gap. That is, the technology fails to coincide with the information that one can extort from the visual data and the understanding of the same data from the users point of view in a particular state of situation [7].

In order to minimize the above mentioned shortcomings, one of the most popular semantic web languages, Web Ontology Language (OWL), has been used in this thesis to design the photo album ontology. This emerging approach focuses not only on domain objects but also on domain events. It helps to integrate one keyword with other keywords in association with defined relationships, which in turn facilitates the finding intended data more precisely. These new features help to lessen challenges of concurrency between data and users, and at the same time help to increase the sharing facility among users. Capability of interpreting visual data, in terms of machine readable as well as human understandable format, provides the facility to find the valid images from the photo data repository. It provides the facility to improve the interoperability among features related to the photo album. New features also can be added gradually as a part of continuous evolving process. Semantic technology supports labeling data uniquely. For example, Protégé, which supports semantic concepts to create classes and derive relationships among those classes, do not allow two classes with the same name, which reduces the possibility of overlapping of classes. [8][9]

This ontology-based photo album allows building a repository to manage different features of images which allows users to search through this repository according to their needs based on the query. Also, this new trend allows the system to increase computational power to manipulate entities of visual data more effectively. Semantic web interpreted with OWL statements has the ability to describe rich annotation structure. Its highly automated metadata creation capability facilitates the search concerning visual data, based on the similarity of contents of the problem domain, is the most important feature of the photo album. This discrete feature helps eliminating data redundancy compared to available photo management software tools. It also helps to increase the feature extraction ability and similarity measures on account of image retrieval.

OWL classes, which are considered as key building blocks of ontology, are grouped together as a specialization hierarchical tree format along with relevant properties applied on it. This important characteristic allows the users to find out the relevant answer necessitating by their query. Moreover, its domain specific features help to model a higher level of semantics, which enable the machine to realize the intended needs of different users in the various situations, in terms of image retrieval [9].

1.2. Problem Statement

In this paper, we propose to design a photo-album ontology using knowledge-based methodology by using Protégé and Web Ontology Language. This domain-specific knowledge management system consists of hierarchical problem domain concepts. OWL will allow representing the conceptual model of photo-album in order to understand the task of developed project knowledge management system. This unique feature will help to create an integrated photo album ontology framework effectively.

The whole process of the design includes the ontology construction, knowledge class organization and defines facts or rules phases.

- Ontology construction is used to create the domain ontology.
- Knowledge class organization phase is used to categorize the interrelationships between the knowledge classes.
- Facts or rules are used to fill in the facts or rules of knowledge classes elicited by domain experts.

This domain-specific, manually built ontology will provide access to the photos uploaded in the album and will also help to find the relevant pictures as queried by users. The Protégé OWL is used to define the terms related to visual data in order to represent and describe images with different attributes. More specifically, Ontology Web Ontology Language Description Logic (OWL-DL) is used to maximize the expressiveness and to fulfill all computation rationally without losing any computational properties. The ability to combine the features of Resource Description Framework (RDF) and OWL (RDF/OWL) in protégé platform, allows describing the instances and constraints of the concerning domain accurately and precisely. [9][10]

At the end, ontology query language Simple Protocol and Resource Query Language (SPARQL), is used to evaluate the photo-album ontology by using some inference actions. SPARQL, which is based on Structured Query Language (SQL), is capable of querying visual graph patterns along with their associations and disassociations. SPARQL search processes allow users to select the class of interest. After determining the class of interest, the semantics of photo-album ontology, along with image instances and properties, creates a relationship between a selected image and other images in the photo repository. Finally, all images relevant to concerning classes are retrieved by the system and then presented to the users.

1.3. Organization

The paper is organized as follows.

- Chapter 2 includes the background study on multimedia, semantic web and its key component, explanation of XML, RDF, OWL and Ontology Design. This chapter also includes other important work related to the photo album management system.
- Chapter 3 includes the overall design framework of the photo album ontology.
- Chapter 4 includes the result with example and detail analysis on those examples.
- Chapter 5 includes a conclusion which provides a brief discussion on the summary, limitations, and future works of the proposed solution.

CHAPTER 2: BACKGROUND STUDY AND RELATED WORK

This chapter provides an overview of the two important areas most relevant to our design. Section 2.1 briefly introduces the background knowledge required for our work. Section 2.2 describes the summary of some important work related to our design. A more detailed discussion of these areas can be found in the literature referenced through this chapter.

2.1. Background Study

2.1.1. Multimedia

Multimedia is the combination of multiple forms of media and content. It comprises the integration of text, audio, still images, animation, video, and interactivity content forms. For example, any software, which includes sound, text, still images and animations, is called multimedia software. Nowadays, multimedia in all forms (audio, video, still images, etc.) is exploding on the web due to the improvement of computer speeds and storage space [9][11].

Theoretically, multimedia and semantic web are an ideal match; where multimedia applications require the description of metadata of their media items, to ease storing, indexing, searching, and retrieving the multimedia information. On the other hand, semantic web, which describes the semantics of web content, involves a number of languages and technologies for annotating web resources to allow the machine to process metadata. However, the role of the semantic web is still limited in lots of multimedia applications where most of them apply non-RDF based techniques [11][12].

2.1.2. The Semantic Web

Nowadays, most of the web content is easily accessible by different individuals. People typically use the web to store, index, search and retrieve the online or personal data. But the currently available software tools are not potential enough to support these activities effectively. Still, there are some common problems associated with their use, such as high recall-low precision, low or no recall etc. Moreover, most information available in the web is weakly structured [8][9].

According to the creator of the World Wide Web, Tim Berners-Lee, the web can be made more intelligent and even more perceptive on how to fulfill the user's needs using the idea of semantic web. The semantic web technology allows us to represent the semantics of the information and services of the web content in an effective manner. It also helps the machine to understand and satisfy the requests of people to use the web content compared to other technology. Unlike the World Wide Web, which contains huge information in the form of documents, semantic web data is capable of processing application or domain independently [8][9][13].

Semantic web requires the addition of metadata or data to information resources, to carry out their implementation. This additional capability allows machines to process the data, described in terms of semantic technology more effectively. The more semantic information associated with data, the more computers can make logical judgment about those data. In short, it is possible to make computers understand, what a data resource is and how it relates to other data. The main objective of the semantic web technology is to make the web more meaningful and increase the reusability of the web contents [8][14].

2.1.3. Components of Semantic Web

The semantic web follows certain standards and tools in terms of its components. However, XML, XML Schema, RDF, RDF Schema and OWL are considered the main components of the semantic web.

2.1.3.1. XML

Unlike HTML, which is a markup language for a certain type of hypertext document, XML is intended as a markup-language for uninformed document structure. XML describes the fundamental syntax of the web documents in a structural way. It not only contains the information about the pieces of the web document, but also demonstrates the relationship among them following the nesting structure. Because of these distinguishing features, XML document is quite easily accessible to machines [9][15] .

XML is a metalanguage for markup, which allows the users to define their own tags. XML elements consist of an opening tag, its content and a closing tag. These elements represent what the XML documents talk about. A document can be represented in a well-formed structure using XML by following certain syntactic rules.

The main purposes of using XML are:

- to maintain the serialization of the syntax for other markup languages in the XML specifications.

- to use as a semantic markup of web-pages.
- to use as a uniform data-exchange format [9][16] .

2.1.3.2. XML Schema

XML schema, which defines the structure of XML documents, is considerably a richer language. The main characteristic of it is that its syntax is based on XML itself, which has assisted to improve the readability, reuse, and refinement of the schemas. To get a separate syntax doesn't require writing separate parsers, editors, pretty printers and so on. XML schema represents the document type at a high level of abstraction. The XML schema is quite straightforward. It enables to analyze the validity of XML parser, which assists us to find some piece of bad XML. Briefly, XML schema language allows providing and restricting the structure of elements, contained within XML documents. Though XML provides a uniform framework for describing data and metadata, it does not provide any means to represent the semantics of data [9][15][17].

There are several languages, which have been developed particularly to represent XML schemas. Such as, The Document Type Definition (DTD) language, XML Schema (W3C) and RELAX NG. Among these, DTD has comparatively limited facility. Other two, XML Schema (W3C) and RELAX NG are more popular at this time [9][17].

2.1.3.3. RDF

RDF stands for Resource Description Framework, which is a data model. It is a universal framework to describe the content of the websites. RDF is a domain-independent concept, through which users can define their own terminology on different domain concepts. RDF has the capability to describe the author of the resource, date of creation, date of modification, key words for search engines, categories of subject, organization of pages in the web site and so on, which is referred to as metadata or data about data. For example, in case of a photograph, RDF consider photographer, title, place, name, date and time, camera model, film used, etc. in the RDF specification. These unique characteristics of RDF allow everyone to share the contents of the web site more easily and effectively. RDF also allows the software developers to build the products which can act as an intelligent agent by providing effective search engines and directories, that in turn help the users to get more control of what they are viewing in web site [9][18].

Although RDF has sufficient expressive power, it uses only binary properties. The basic idea of RDF involves resources, properties and statements. [9]

Here resources denote an object about which we talk. For example, people, places, cars, etc. Each resource has a unique Universal Resource Identifier (URI). Properties describe the relationship between resources. It is also identified by the URI. Statement asserts the properties of resources, which consist of a resource, a property and a value [19].

Basically, the RDF data model is quite similar to the classic conceptual modeling approaches, such as, Entity-Relationship or Class diagrams. Because, its fundamental concepts are based on the idea of making statements about web resources in terms of subject-predicate-object expressions which is called triples in RDF language. Here subject denotes the resource, the object denotes attributes of the resource; and predicate represents the relationship between the subject and the object. For example, "The tree has the color green." According to the term triple in RDF, "The tree" denotes a subject, "has the color" denotes the predicate and "green" denotes the object. The encoding of the triple idea varies from format to format [9][19][20].

In fact, RDF is an abstract model with several serialization formats, in which a group of RDF statements basically represents a labeled, directed multi-graph. The abstract syntax of the RDF assists to connect its concrete syntax to its formal semantics [9][20][21].

2.1.3.4. RDF Schema

RDF schema is a language which describes the properties and classes of RDF-based resources. It also describes the semantics of generalized-hierarchies created by using those properties and classes as well. There are two important RDF Schema constructions: subClassOf and subPropertyOf. The type property is used to denote the RDF objects, which may be instances of one or more classes. The subClassOf property assists to organize the hierarchical association among the classes in the RDF Schema specification. Also, the subPropertyOf performs the same activity for properties. Moreover, it is possible to assign constraints on properties, using domain and range constructs. These constructions will extend the capability of RDF Schema, in terms of vocabulary and interpretation of RDF expressions [9][22][23].

2.1.3.5. OWL

OWL stands for Web Ontology Language. OWL is semantic web standards, which provides a framework for the excellence management of web information and also to increase the sharing and reusability of the web information. It sufficiently supports the well-defined syntax, the effective formal semantics, the efficient reasoning and the convenience of expression of the web information. Basically, OWL is an extension of the RDF Schema. But OWL has greater machine interpretability, more vocabulary and precise syntax compare to RDF. OWL describes the web information, which almost holds the exact meaning of that information. It also describes the relationships among the web information. Additional vocabulary makes the OWL capable of advanced web search and greater machine interpretability of the web content. OWL is basically designed to strengthen the foundations of the semantic web, which is able to carry out richer integration and interoperability of data across the domain of different concerns [10][24].

OWL has three sublanguages with different levels of expressiveness:

- OWL Lite: OWL Lite, which has limited facility, was basically intended to support the users who are concern with simple hierarchical organization and simple constraints. Though its expressiveness is restricted, OWL Lite is easy to grasp and easy to implement for the users. But since it is difficult to develop the OWL Lite tools, it is not widely used.
- OWL DL: OWL DL is so named because it is associated with the description logic, a field of research, where different logics have been studied to form the schematic foundation of OWL. In fact, OWL DL was designed to present maximum expressiveness, retaining the complete computational efficiency. The main advantage of the OWL DL is that, it allows the efficient reasoning support. Although OWL DL includes all constructs of OWL language, they can be used only under certain restrictions. It is not fully compatible with RDF. But every legal OWL DL document is a legal RDF document.

- OWL Full: OWL Full supports completely different semantics from OWL Lite or OWL DL. It uses the entire OWL languages primitives. The main advantages of the OWL Full is that it preserve the compatibility with RDF schema, in terms of syntax and semantics. So any legal RDF document is a legal OWL Full document. Although OWL Full allows an ontology to evolve the meaning of the pre-defined vocabulary, it has lack of complete and efficient reasoning support [9][10][24][25].

2.1.4. Ontology Design

Ontology design, which is a phase of the ontological engineering life cycle, specifies both the concepts of the problem domain and the interrelationship between those concepts. In other words, ontology design helps to create explicit formal specifications in terms of the domain and relations among them. The main reason to design an ontology is to provide a common structured vocabulary for researchers, to improve the sharing, reusability and to make explicit domain assumption of knowledge about facts, that are perceptible by senses, with regard to world of interest. Because, both users and following designers do not always share the same assumption with the pioneer designers, it is quite difficult to realize the embedded scope of ontology. Moreover, this limitation makes it difficult to differentiate the fundamental traits within ontologies [9][26]. Common components which are used to design the ontology are:

- Classes: represent the collection of concepts or set of objects.
- Individuals: define the instances or objects or concepts.
- Attributes: features, characteristic or parameters, that classes can have.

- Slots or roles or properties: describe various features and attributes of the concept.
- Relations: the way classes and individuals are related to each other.
- Restrictions or facets or role restrictions: formally stated restrictions on slots.

There are a number of languages available to design ontologies. The most recent advancement of ontology language is Web Ontology Language (OWL), developed by the World Wide Web Consortium (W3C) [9][26][27].

In this paper, we will apply OWL to design the of photo album ontology using the ontology development tool, Protégé. Protégé and OWL will describe elaborately in the next chapter.

2.2. Related Work

In this section, we will discuss the previous research effort related to our work. A brief discussion will also be given about popular websites in terms of multimedia objects sharing and semantics-based management systems.

Over the past years, a lot of relevant researches have been made out on managing personal digital photos. During this time, there have been a number of popular, marketable, free software packages, which support only basic photo management operations, such as storing, indexing, searching and retrieving personal photos. For example, Picasa (Google) [3], flickr [4] , facebook [28], orkut [29], Photosmart Essential (HP) [30] , Windows Live Photo Gallery (Windows) [5] and so on. Although these software packages maintain a high standard in terms of managing the digital photos, their capability is still limited in the searching and retrieving images.

Since advanced multimedia technologies have made it possible to manage photos with low cost, the size of the photo collection is increasing rapidly day by day. So it has become a cross-cutting issue of how to manage images in the website [2].

There are a number of text-based image search engines, such as Yahoo™, Google™, etc. available nowadays. But these techniques are suffering from some shortcomings. Because their approach basically emphasizes keywords, it requires adequate text information, which can be able to describe the images more precisely and correctly. Because of the lack of using proper keywords relevant to the concerning issue, irrelevant information might be provided by the search engines [6][31].

Other than the text-based image retrieval, there is another approach named as ontological-based image management systems. This previous ontological-based image management systems focused only on querying and browsing a side of the concerning problem [32][33]. The term annotation is discussed in terms of doing browsing a hierarchy tree or performing keyword-based searches for concepts [34][35]. There is a primary ontological-based approach for image annotation in context-based image retrieval (CBIR). This approach uses the low-level features to facilitate automatic image content extraction, such as texture, shape and color [31]. The idea of the content-based image retrieval (CBIR) was proposed to assist users for efficient image retrieval [36][37]. In recent years, even though the idea of content-based retrieval [38][39][40] has applied to manage photos, the techniques (ShowBox [41][42], MyPhotos [43], QBIC [44], VisualSEEk [45], Photobook [46] etc.) only support visual similarity based search in terms of color, texture, and shape .

Concisely, performance of the CBIR systems is quite inadequate to depict the semantic content to a great extent. Because users are basically comfortable to recognize images based on high-level concepts, such as, text and generally query images by semantics [47][48]. The solution of the problem is image annotation, with important keywords, which is capable of learning the correspondence between visual features and semantics of images [49].

A number of attempts have been taken to exercise the automatic image annotation having a base on low-level image features. In that respect, images are classified into different categories. Afterwards, labels have been provided by important keywords relevant to those categories. Wang JZ, Li J, Wiederhold G in their research classify images into semantic categories (For example, textured or nontextured and graph or photograph), which makes it possible to improve the image retrieval, adapting the semantic based search methods [50]. Research has also been done combining the visual and textual features [51]. Applying the idea of the tf-idf scheme [52], images are classified into two categories, such as indoor and outdoor according to their visual objects. Images are also classified according to the city versus landscape [53] and portrait versus nonportrait [54]. Even though it was believed that, CBIR could be able to lead to a promising future; its performance was not that much satisfactory [31].

Effective relevant feedback methods have also been applied by a few researchers, which require much more user interaction. For image annotation, the user feedback method has been put to use by integrating additional keywords. In this case, as soon as users notify some images are relevant to a query, the method updates the annotation of the images by linking the query with the images. It has been proved that feedback techniques are quite efficient and accurate comparing to manual annotation [55].

2.3. Summary

From the above discussion it is somewhat apparent that a number of significant efforts have been carried out by researchers regarding photo management system. At present, there are several web album or image hosting services available to organize images. Amongst those, Picasa and Flickr are considerably popular, but both of them use traditional approaches. In order to describe images, both services allow tags and captions with explicit keywords. The keyword based approach sometimes faces difficulty in regards to rich precision and formal semantic information. Moreover, users are almost not able to label hundreds and thousands of images [9]. Since images contain rich formal semantic information, designing an ontological based photo album, might assists to manage images in a systematic fashion. In this paper, a photo album will be sketched out using the perception of ontology based framework. Different classes (concepts), properties and instances will be created in a hierarchical manner, using the ontology development tool Protégé.

CHAPTER 3: DESIGN FRAMEWORK

3.1. Overview

This chapter provides a brief description of Protégé, which has been used to create OWL ontologies for photo album. Chapter 3 also describes different components of OWL ontology, a brief introduction on Simple Protocol and RDF Query Language (SPARQL), concepts of FOAF and Dublin Core. As a part of the detail design framework, purpose of all classes and subclasses, along with their relevant properties, restrictions, instances have been discussed, concerning the developed photo album ontology.

3.2. Tools and Enable Techniques

3.2.1. Protégé

Protégé, which has been developed at Stanford University in partnership with the University of Manchester, is a free, open source ontology editor and knowledge-management framework. It helps to create, envision and implement a rich structure of knowledge-based models, to provide a domain friendly support to enter data. Moreover, Protégé can be customized extensively to create flexible knowledge acquisition tools and application, with the help of a plug-in architecture and a Java-based Application Programming Interface (API) [56].

The effort, that carried out to evolve Protégé, was mainly to allow individuals or groups to share significant and clear understanding of complex organizational perceptiveness. Therefore, help to lessen the redundant work, time consumption, that lead to increase capital turnover and the adaptability, in accordance with ever-changing organizational environments. Initially, the development of domain independent Protégé has been mainly influenced by biomedical applications. But, nowadays, Protégé has been successfully using in many applications, such as, knowledge-based information management and integration systems, e-commerce, logical facts and semantic web services. Ontologies that are created using Protégé, can be exported into various formats- such as Resource Description Framework (RDF-S), Web Ontology Language (OWL) and Extensible Markup Language (XML Schema) [9][57].

At this time, Protégé is used to resolve knowledge-based solution, such as, biomedicine, organizational insights and intelligence gathering. It is endorsed by a privileged group of people like government, business academics, developers and commercial users [9].

3.2.2. Components of OWL

There are different formats of ontology language available in Protégé platform based ontologies. In our paper, we are going to use one of the popular standard ontology languages OWL, to design our photo album ontology. Because, it supports the case, where the concepts have more than one parent and have richer set of operators, such as union, intersection and negation, which are significant features for photo album ontology. Moreover, complicated concepts are also described precisely for different logical models using Protégé OWL platform. Availableness of reasoner allows us analyzing the consistent of the definition of concepts, their relationship and any other relevant documents too. In short, explore the uniformity of overall hierarchical trees for concerning problem domains [10].

Although Protégé supports the similar components to OWL ontologies, the terminology, used in Protégé is little bit different than the terminology used in OWL ontologies. Therefore, whereas Protégé frames use instances, slots and classes, OWL ontologies use individuals, properties and classes respectively [58].

3.2.2.1. Individuals

Individuals represent the objects of concerning domain concepts. It also refers to an instance of classes.

```
<Colleagues rdf:ID="Amanda">
    <rdfs:label rdf:datatype="&xsd;string">Eva</rdfs:label>
  </Colleagues>
```

In this example, individual Amanda is an instance of the class Colleagues. The property rdfs:label links Amanda to a typed literal with the XML Schema datatype string. The XML schema document has then necessary information about the syntax of this concerning datatype [59].

3.2.2.2. Properties

Properties which support binary relationship, link two individuals together. For example, the property takenWith might link the individual of class Photo to the individual of class Colleagues. OWL properties can have different types of characteristics- such as, functional, inverse, symmetric, asymmetric, transitive, reflexive, etc., which allow it to design the ontology more effectively [58].

3.2.2.3. Classes

OWL classes are the concrete representation of the classes, considered as the container of the sets of individuals. Classes can be structured as a superclass-subclass hierarchical manner. For example, consider the classes Camera and Digital, where Digital is the subclass of Camera. This says that all cameras are digital cameras and all members of the class Digital are members of the class Camera. Being a Digital implies that it is a Camera and Digital is subsumed by Camera [60].

3.2.3. Friend of a Friend (FOAF)

FOAF, a machine-readable ontology, assists to create associations among the people or objects who are literally not involve with each other. It describes the real world objects through the relevant properties [68].

FOAF, which doesn't have any centralized data sources; collects properties and classes using W3C's RDF technology. FOAF is a kind of text file document, that use the syntax of the Resource Description Framework (RDF) using the format of XML,N3 and RDFa. It gathers a number of different terms to describe individuals, groups, different documents etc. for any kind of application domains. If anybody uses the format of FOAF document to publish their own information, machine can utilize the information efficiently as vocabularies, with the help of some significant concepts along with their relationship. Precisely, if someone wants to extend their own ontology with the existing one, they can use the FOAF format. For example, classes like foaf:person, foaf:location, foaf:photo can be defined in FOAF format. Then, relevant properties such as foaf:depicts, can be used to relate foaf:photo to foaf:person or to foaf:location. This FOAF concept can be used in our ontology by parsing RDF resources [69].

Even though we have used our own camera concepts for this ontology, it is also possible to use other existing camera ontology using the FOAF terminology. Precisely, the FOAF concept, which assists to correlate among the objects from different domain concepts, can be used to extend our ontology by correlating with the existing camera ontology [70].

3.2.4. Dublin Core

The word "Dublin" named after Dublin, Ohio, U.S., was invented in 1995, a combined effort of the Online Computer Library Center, a library consortium. And the word "Core" refers the list of metadata elements. Anytime this list can be updated with new terms. The Dublin Core semantic standard is maintained by an international, cross-disciplinary group of professionals from librarianship, computer science, text encoding, museums, and other related fields of scholarship and practice. This standard is used to describe, manage and exchange set of metadata information in the digital libraries to use the World Wide Web (WWW) environment. It also can be used to exchange information among incompatible secured domains and among interoperable data resources as well. It usually uses the convention of XML and Resource Description Framework. 15 base text fields convention are used to describe all sorts of digital materials, such as, books, video, audio, images etc. The name of the fields are, Title, Creator, Subject, Description, Publisher, Contributor, Date, Type, Format, Identifier, Source, Language, Relation, Coverage and Rights [71] [72] [73].

So, this handy new concept can also be used to make our ontology more flexible and viable in the web world.

3.2.5. SPARQL

SPARQL, which stands for Simple Protocol and Rdf Query Language, is a query language to access RDF data resources of the domain of interest [61]. It was standardized by the RDF Data Access Working Group (DAWG), which is a part of the World Wide Web Consortium, in order to apply in the semantic web technology [62]. SPARQL basically maintains the following framework to query the RDF data sources [63]. Following figure shows the anatomy of SPARQL query language.

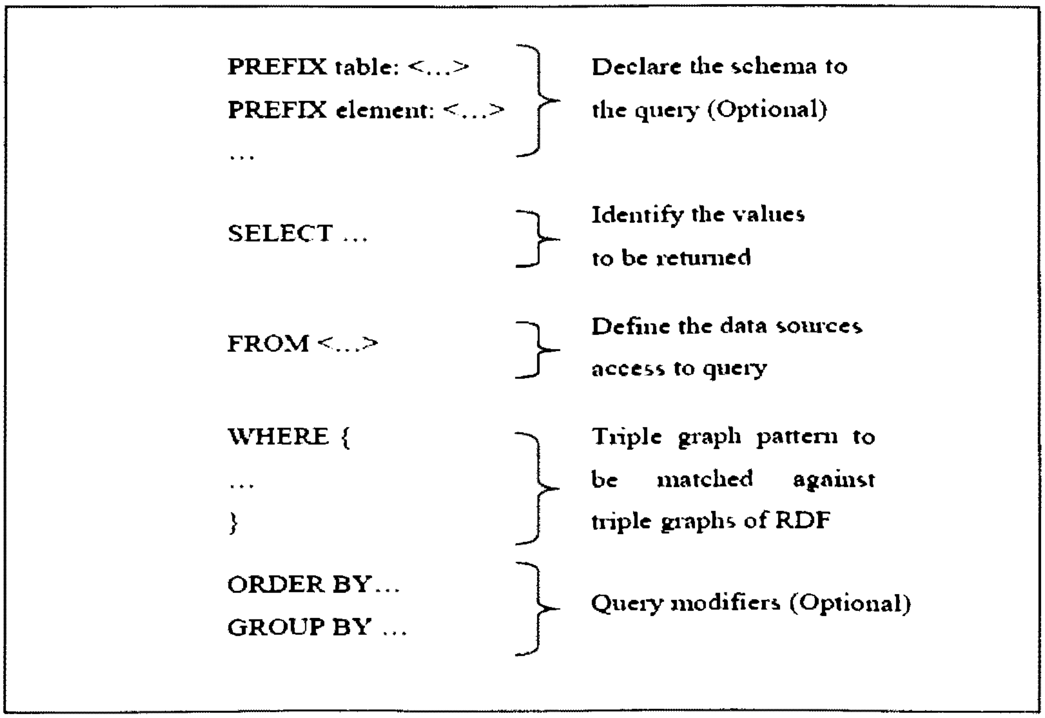


Figure 1: Anatomy of SPARQL query

Following is the example of SPARQL query:

```
PREFIX table: <http://localhost:8080/2010/photoalbum0710V3.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?Person ?Camera
FROM <http://localhost:8080/2010/photoalbum0710V3.owl>
WHERE
{
?Photo table:takenWith ?Person.
?Photo table:takenBy ?Camera.
}
```

In the above instance, SPARQL query pattern consists of a couple of triple graph patterns. Here each of the triple pattern must have matched with the relevant desired properties: takenWith and takenBy. In other words, this query will match any data resources that have all two of the desired properties. The resources, which do not contain these properties will not be included in the query results as it will not satisfy the related triple graph patterns. That is, one can not select a variable which is not included into graph pattern [64].

3.3. Important Parts of Protégé and SPARQL

Following are important points in developing an application using Protégé and SPARQL:

- Protégé is able to use the reasoner to do automatic classification among the class hierarchy. If there is an ontology, which involves several thousands of classes in them, reasoner can be helpful to check the consistency between subclass-superclass of the ontology. Without a reasoner, it is difficult to maintain large and logically related classes efficiently [58].

- Protégé support the agility, which allows modifying the existing ontology based on the needs of customer and users feedbacks. It increases the usability of application, created using the protégé.
- It is very handy to visualize the design of developed knowledge based domain using protégé. [65]
- SPARQL supports more complicated and worldly queries. [66]
- It is easy to map between subject and object using the properties in SPARQL. [64]

3.4. Ontology Design

3.4.1. Ontology Design: Classes

To build the photo album ontology, several numbers of classes have been considered, which are significantly relevant to a photo. The following discussion will be carried out on how those classes have been organized and correlated using different properties.

In protégé, the owl:Thing is the initial class, which contains all individuals of all classes. In this case, PhotoAlbum, Photo, Person, Location, Events, Title, Time and Camera are considered as the subclasses of owl:Thing.

3.4.1.1. Class: PhotoAlbum

PhotoAlbum, which is one of the important classes in the photo album ontology, contains all kinds of photos. Besides it has its own properties. In this ontology, the class PhotoAlbum has properties hasPhoto and hasTitle. So using these properties, anybody can find out their desired picture entering the name of photos.

3.4.1.2. Class: Photo

The class Photo has a number of properties to make associations among classes, in this case, Person, Location, Title, Camera, Events and Time. Using these properties someone can find the photos which have been taken with a particular person at a particular moment and at a particular event. For example, if someone tries to find the list of photos that have been taken in the USA with friends, using the properties takenAt and takenWith, it is possible to find those photos by associating with the classes Location and Person.

3.4.1.3. Class: Location

Based on the knowledge of taking photos at different places, Location class is created in this ontology. To specify the place based on the city, country, state and continent, the properties hasCity, hasCountry, hasState and hasContinent are created. Using these properties, the Location class will be able to find the photos taken at any location, such as, in any city, any country, any state and any continent.

3.4.1.4. Class: Camera

Photos can be taken using different cameras. There are a number of different cameras are available in these days. Based on that, we created the class Camera and then further extended it to DSLR, Digital and FilmCamera. Following is the figure of the class Camera.

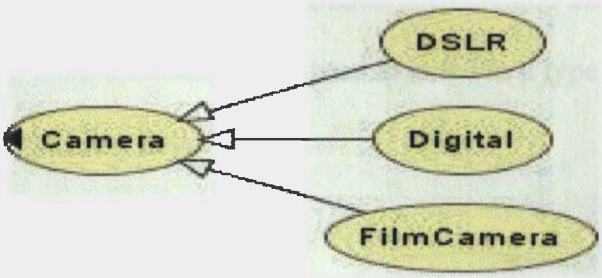


Figure 2: Camera class

In this case, if users are interest to find any photos, which have been taken with a particular camera type, it is possible to find those photos based on this ontology design. To validate this reasoning, rdf:type, which is the instance of the rdf:property, has been used to asserted the different camera types.

3.4.1.5. Class: Time

Time class also has been further classified into Date, Months and Years to keep track of the period of time, the picture being taken. Following is the figure of the class Time.

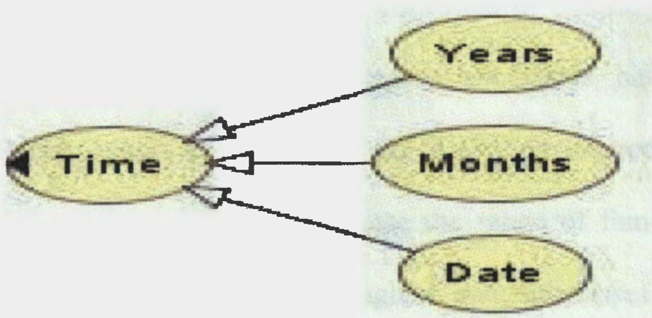


Figure 3: Time class

3.4.1.6. Class: Events

Considering the different event concepts- Vacation, Official, Games, Ceremony and Festival, subclasses with some name convention, have been created under the class Event. To perform the reasoning, these subclasses have been asserted as a type of the class Events. Following is the figure of the class Events.

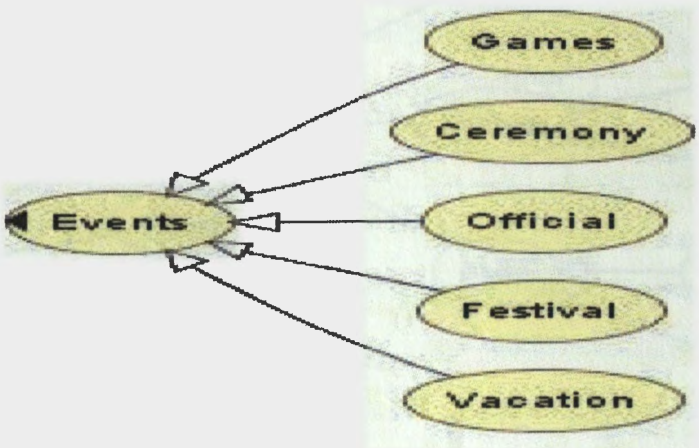


Figure 4: Events class

3.4.1.7. Class: Person

Photos might be taken with different individuals at different moments. Individual can be categorized as family member, friend, colleague or others. Based on this basic concept, FamilyMembers, Friends and Colleagues subclasses have been created as subclasses of the Person class, so that, users can find their desired pictures, searching against the type of the person. In addition, since the range of family members can be considered as mother, father, sister, brother, daughter, son; subclasses Parents, Sibling and Child have been created as a subclass of FamilyMembers. In addition, Parents class is further extended to Mother and Father subclasses; Siblings class is further extended to Sister and Brother subclasses and Child class is further extended to Daughter and Son subclasses. In this design, all subclasses have been assigned as the type Person. Following is the figure of the class Person.

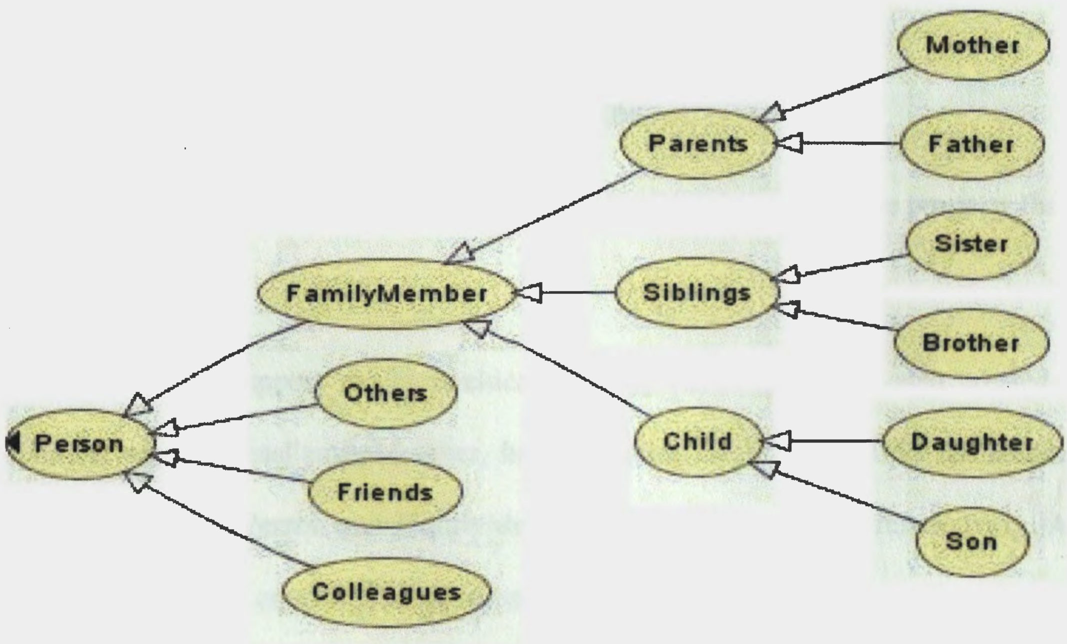


Figure 5: Person class

3.4.2. Ontology Design: Properties

In this photo ontology design framework, several numbers of properties have been used to do the association among different classes, which are described below.

takenWith and Takes: These are the properties, which relates the class Photo with the class Person and vice versa. That is, both of them are the inverse with each other, which allows us to say a Photo takenWith a Person or a Person Takes a Photo. For example, we can say, list the photos, which a person has taken in his or her vacation. Following figure depicts the inverse property.

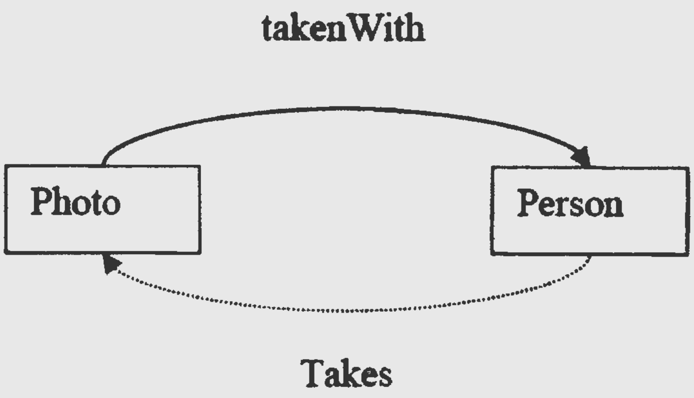


Figure 6: An example of an inverse property: takenWith has an inverse property that is Takes

In order to support the hierarchical property relationship, we also created other properties, such as, hasFamilyMember, hasColleagues, hasFriends to do the association between the classes Person and FamilyMember, Colleagues, Friends respectively. At the same time, we have created inverse property of those like, isAPerson. Again we have created the property hasParents, hasSiblings, hasChild to relate the class FamilyMembers with the classes Parents, Siblings, Child respectively and isAFamilyMember as their inverse property. In addition to this, hasMother, hasFather properties; hasDaughter, hasSon; hasBrother, hasSister; properties have been created to associate the classes Parents with Mother and Father classes; with Daughter and Son classes; with Brother and Sister classes respectively. Simultaneously, isAParent, isAChild and isASiblings properties have been created as their inverse property.

This design framework has made it more flexible to find the desired photos in different ways. So if someone wants to find the photos and if they search for it either way for example, based on the name of the photo or based on mentioning a particular person like sister, brother etc.

takenAt: It relates the class Photo with the class Location. As the Location class has supported the properties hasCity, hasCountry, hasState and hasContinent, anybody can find the pictures taken at different type of location. Besides, isALocation inverse property has been created to increase the flexibility of the ontology design framework.

eventsOn and isAEvents: It relates the class Photo with the class Events. For example, a photo must have taken on an event. In this ontology ceremony, festival, games, official and vacation are considered as events. As an inverse of eventsOn property isAEvents is used.

nameOfPhoto: It relates the class Photo with the class Title. For example, a photo must have a title or name.

takenBy and isACamera: It relates the class Photo with the class Camera. isACamera, inverse property of takenby is used to correlate the classes DSLR, Digital, FilmCamera with the class Photo.

rdf:type: rdf:type which is instance of the rdf:property is used to refer that a resource of rdfs is an instance of a class. It follows the triple graph pattern such as,

?Person rdf:type :FamilyMember.

This photo album ontology contains the subclasses FamilyMember, Friends, and Colleagues – different kind of people, under the superclass Person. So if we take a picture, which can have different people (lets say with the friends) and the family members (in the same photo); we need to use this restriction. Because, we do not need to pull out the names of other people (in this case the friend's name) other than the family members. If we do not use this restriction, then, it will show all names, those who are in that photo, including the friend's name and the family member's name as well.

rdf:label: It is used to refer to an instance to understand as human-readable format. It also follows triple pattern such as,

?Person rdfs:label "Eva".

This line refers that, whose information we are seeking here (in this case, name of the person is Eva).

3.4.3. Ontology Design: Instances

Several numbers of instances have been created as members of each class, to check the feasibility of our designed ontology. Instances from different classes can be referred to a particular class, even though they are disjoint in regards to sibling issue. For example, instances of the class FamilyMember, Friends and Colleagues referred to a class Person. Following figure shows the relationship among instances within the Person class.

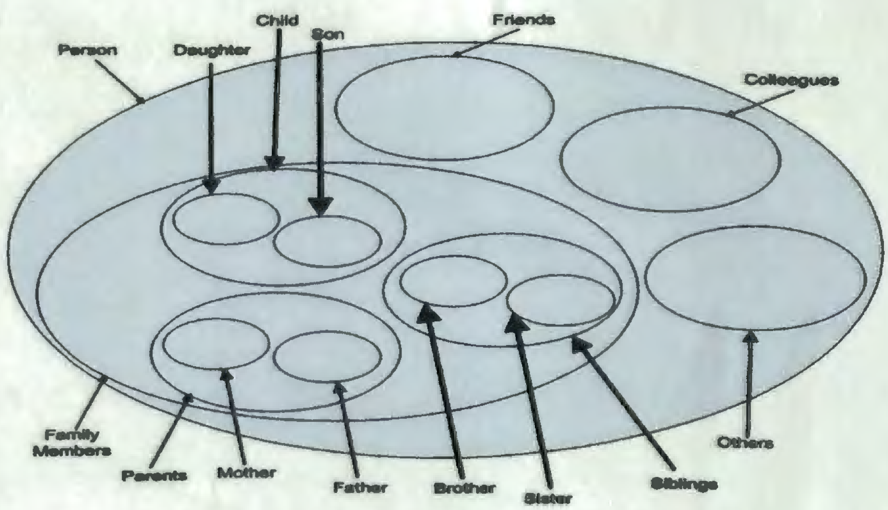


Figure 7: Relationship among the instances of subclasses within the Person class

But instances of FamilyMember can't be instances of Friends. So if a user is specific about the information of a person, with whom he takes the photo, using this ontology design concepts, it is quite easy to find that desired photo.

In or to find the required pictures based on this design framework, a query language, SPARQL, has been used, which is discussed in details in the next chapter.

CHAPTER 4: CASE STUDIES

4.1. Scenario

Eva is a person who has a photo album. On it, she stored lots of her pictures, which she took with her family members, friends and colleagues. Those photos have been taken in different locations, at different events, with different people, by different cameras.

4.1.1. Case Study 1 and Analysis

Find those pictures, which Eva has taken with her family members, in different countries in Asia. Pictures were taken with digital cameras.

```
PREFIX table: <http://localhost:8080/2010/photoalbum0710V3.owl#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?Person ?Location ?ImageName ?Camera

FROM <http://localhost:8080/2010/photoalbum0710V3.owl>

WHERE

{

?Photo table:takenWith ?Person.

?Photo table:takenAt ?Location.

?Photo table:nameOfPhoto ?ImageName.

?Photo table:takenBy ?Camera.

?Person rdf:type :FamilyMember.

?Location rdf:type :Country.
```

?Person rdfs:label "Eva".

?Location rdfs:label "Asia".

?Camera rdf:type :Digital.

}

Following section provide the information to find the desirable images:

?Photo table:takenWith ?Person. /*Name of the person*/

?Photo table:takenAt ?Location. /*Where it has been taken*/

?Photo table:nameOfPhoto ?ImageName. /*Name of the Image*/

?Photo table:takenBy ?Camera. /*Name of the Camera*/

In the following line provide the information about the person with whom picture has been taken.

?Person rdf:type :FamilyMember.

Photo album ontology contains the subclasses FamilyMembers, Friends, and Colleagues – different kind of people under the superclass Person. So if we take a picture, which can have different people (let us say with friends) and family members (in the same photo); we need to use this restriction. Because, we do not need to pull out the names of other people (in this case the friend's name) other than the family members. If we do not use this restriction, then, it will show all names those who are in that photo, including the friend's name and the family member's name as well.

The following lines say, in which countries of Asia, the pictures have been taken. If we take a picture on different continents other than Asia and, if we do not use this restriction, the query will also pull all different photos from other places, like North America or Africa.

?Location rdf:type :Country.

?Location rdfs:label "Asia".

Following line says, what type of camera has been used to take those pictures. The query is particular to find the photos, which has been taken by Digital Camera, that's why we need the restriction.

?Camera rdf:type :Digital.

Following line is saying whose information we are seeking here (in this case, name of the person is Eva)

?Person rdfs:label "Eva".

Following figures show the result of the case study 1.

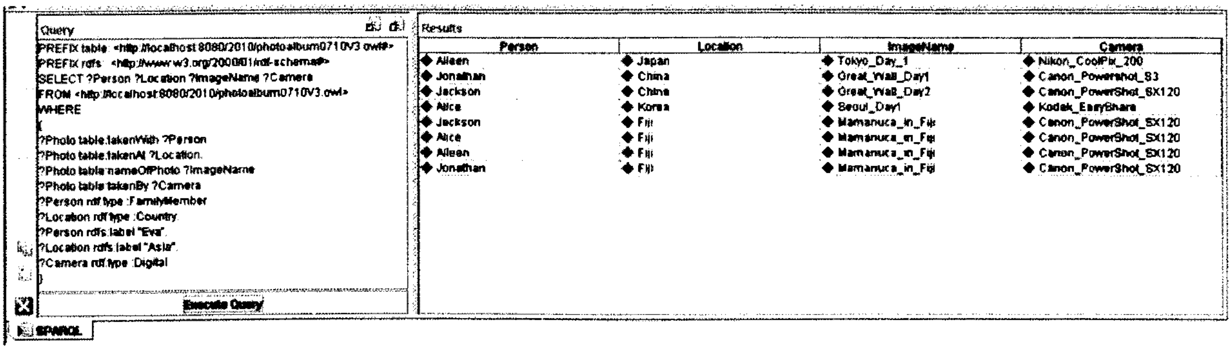
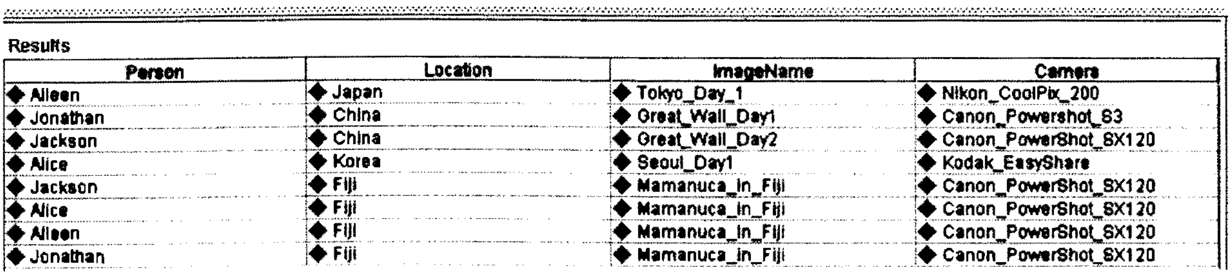


Figure 8(a): Result of case study 1



Results

Person	Location	ImageName	Camera
◆ Aileen	◆ Japan	◆ Tokyo_Day_1	◆ Nikon_CoolPix_200
◆ Jonathan	◆ China	◆ Great_Wall_Day1	◆ Canon_Powershot_S3
◆ Jackson	◆ China	◆ Great_Wall_Day2	◆ Canon_PowerShot_SX120
◆ Alice	◆ Korea	◆ Seoul_Day1	◆ Kodak_EasyShare
◆ Jackson	◆ Fiji	◆ Mamanuca_in_Fiji	◆ Canon_PowerShot_SX120
◆ Alice	◆ Fiji	◆ Mamanuca_in_Fiji	◆ Canon_PowerShot_SX120
◆ Aileen	◆ Fiji	◆ Mamanuca_in_Fiji	◆ Canon_PowerShot_SX120
◆ Jonathan	◆ Fiji	◆ Mamanuca_in_Fiji	◆ Canon_PowerShot_SX120

Figure 8(b): Result of case study 1

4.1.2. Case Study 2 and Analysis

Find those pictures while Eva was on vacation with her family members in tropical countries, using different cameras.

```
PREFIX table: <http://localhost:8080/2010/photoalbum0710V3.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?Person ?Location ?ImageName ?Camera ?EventsName
FROM <http://localhost:8080/2010/photoalbum0710V3.owl>
WHERE
{
?Photo table:takenWith ?Person.
?Photo table:takenAt ?Location.
?Photo table:nameOfPhoto ?ImageName.
?Photo table:takenBy ?Camera.
?Photo table:eventsOn ?EventsName.
?Person rdf:type :FamilyMember.
?Location rdf:type :Country.
?Person rdfs:label "Eva".
?Location rdfs:label "TropicalCountry".
?EventsName rdf:type :Vacation.
}
```

Following section provide the information to find the desirable images:

```
?Photo table:takenWith ?Person. /*Name of the person*/
?Photo table:takenAt ?Location. /*Where it has been taken*/
```

?Photo table:nameOfPhoto ?ImageName. /*Name of the Image*/

?Photo table:takenBy ?Camera. /*Name of the camera*/

?Photo table:eventsOn ?EventsName. /*Name of the event*/

In the following line provide the information about the person with whom picture has been taken.

?Person rdf:type :FamilyMember.

Photo album ontology contains the subclasses FamilyMembers, Friends, and Colleagues – different kind of people under the superclass Person. So if we take a picture, which can have different people (lets say with the friends) and family members (in the same photo); we need to use this restriction. Because we do not need to pull out the names of other people (in this case the friend's name) other than the family members. If we do not use this restriction, then, it will show all names those who are in that photo, including the friend's name and the family member's name as well.

Following lines say, in which Tropical country pictures have been taken. If we take a picture in different continents other than Tropical Country, and if we do not use this restriction, it will pull all different photos from other places, like Asia, North America, or Africa.

?Location rdf:type :Country.

?Location rdfs:label "TropicalCountry".

In this query, we did not use any restriction on camera. Because, the query is only concern about different types of cameras other than any particular type.

Following line is saying, whose information we are seeking here (in this case, name of the person is Eva)

?Person rdfs:label "Eva".

The following line says on which event pictures have been taken. In the query it was asked to find the photos, which have been taken on different vacations. That is why we need the restriction.

?EventsName rdf:type :Vacation.

Following figures show the result of the case study 2.

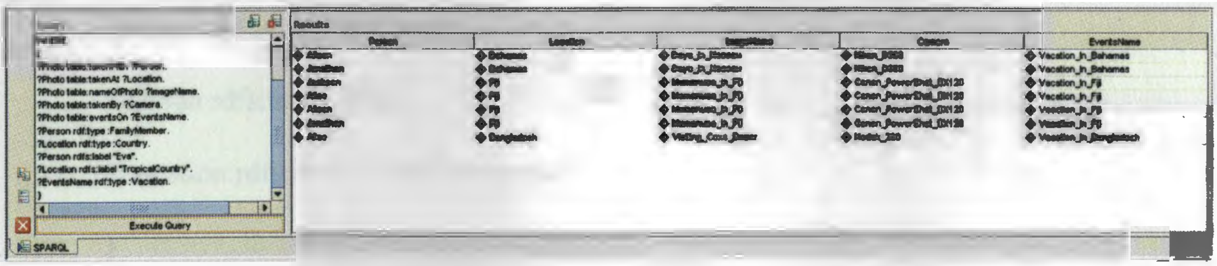


Figure 9(a): Result of case study 2

Results

Person	Location	ImageName	Camera	EventsName
Aileen	Bahamas	Days_in_Nassau	Nikon_D300	Vacation_in_Bahamas
Jonathan	Bahamas	Days_in_Nassau	Nikon_D300	Vacation_in_Bahamas
Alice	Bangladesh	Visiting_Coxs_Bazar	Kodak_200	Vacation_in_Bangladesh
Jackson	Fiji	Mamanuca_in_Fiji	Canon_PowerShot_SX120	Vacation_in_Fiji
Alice	Fiji	Mamanuca_in_Fiji	Canon_PowerShot_SX120	Vacation_in_Fiji
Aileen	Fiji	Mamanuca_in_Fiji	Canon_PowerShot_SX120	Vacation_in_Fiji
Jonathan	Fiji	Mamanuca_in_Fiji	Canon_PowerShot_SX120	Vacation_in_Fiji

Figure 9(b): Result of case study 2

4.1.3. Case Study 3 and Analysis

Find those pictures, which Eva took with her colleagues, at different location in North America.

PREFIX table: <http://localhost:8080/2010/photoalbum0710V3.owl#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?Person ?Location ?ImageName

```
FROM <http://localhost:8080/2010/photoalbum0710V3.owl>
WHERE
{
?Photo table:takenWith ?Person.
?Photo table:takenAt ?Location.
?Photo table:nameOfPhoto ?ImageName.
?Person rdf:type :Colleagues.
?Person rdfs:label "Eva".
?Location rdfs:label "North America".
}
```

Following section provide the information to find the desirable images:

```
?Photo table:takenWith ?Person. /*Name of the person*/
?Photo table:takenAt ?Location. /*Where it has been taken*/
?Photo table:nameOfPhoto ?ImageName. /*Name of the Image*/
?Photo table:eventsOn ?EventsName. /*Name of the event*/
```

In the following line provide the information about the person with whom picture has been taken.

```
?Person rdf:type : Colleagues.
```

Photo album ontology contains the subclasses FamilyMembers, Friends, and Colleagues – different kind of people under the superclass Person. So if we take a picture, which can have different people (lets say with the friends) and colleagues (in the same photo); we need to use this restriction. Because, we do not need to pull out the names of other people (in this case my friend's name) other than the colleagues. If we do not use this restriction, then, it will show all names those who are in that photo, including my friend's name and the colleague's name as well.

Following line is saying, which informative data we are seeking here (in this case, name of the countries in North America). If we take an image in different continents other than North America, and if we do not use this restriction, it will pull all different photos from other places, like Asia, Tropical country, or Africa.

?Location rdfs:label "North America".

Following line is saying, whose information we are seeking here (in this case, name of the person is Eva)

?Person rdfs:label "Eva".

Following figures show the result of the case study 3.

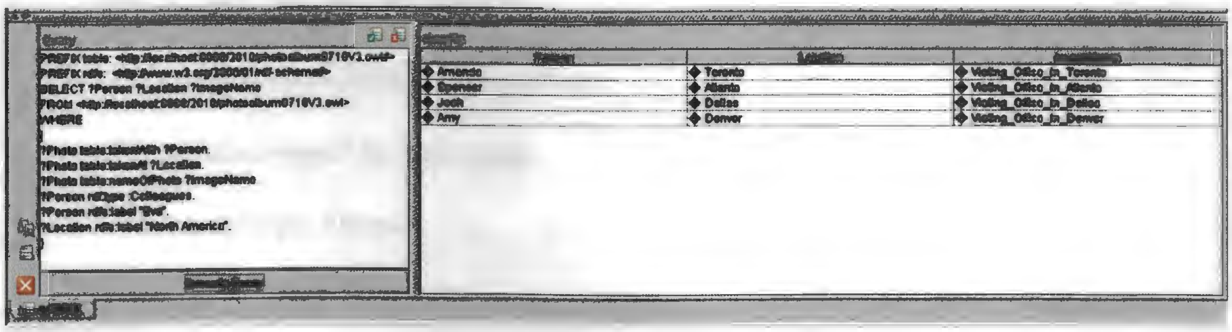
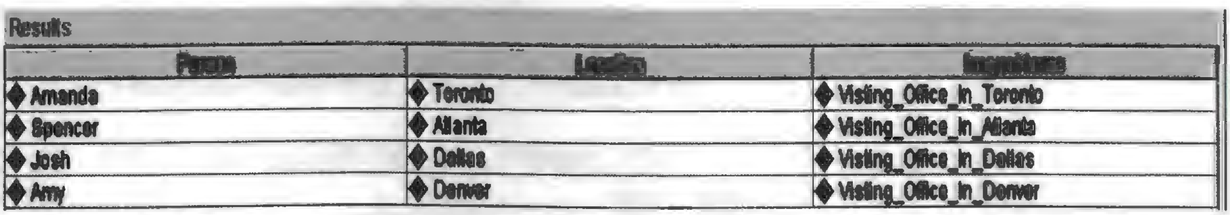


Figure 10(a): Result of case study 3


Results

[illegible]	[illegible]	[illegible]
Amanda	Toronto	Visting_Office_In_Toronto
Spencer	Atlanta	Visting_Office_In_Atlanta
Josh	Dallas	Visting_Office_In_Dallas
Amy	Denver	Visting_Office_In_Denver

Figure 10(b): Result of case study 3

4.1.4. Case Study 4 and Analysis

Find those pictures, which Eva and her friends taken at different ceremonies, with different camera.

```
PREFIX table: http://localhost:8080/2010/photoalbum0710V3.owl#
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?Person ?Camera ?ImageName ?Ceremony
FROM <http://localhost:8080/2010/photoalbum0710V3.owl>
WHERE
{
?Photo table:takenWith ?Person.
?Photo table:takenBy ?Camera.
?Photo table:nameOfPhoto ?ImageName.
?Photo table:eventsOn ?Ceremony.
?Person rdf:type :Friends.
?Person rdfs:label "Eva".
}
```

In the following section, I am trying to find the information regarding the image:

```
?Photo table:takenWith ?Person. /*Name of the person*/
?Photo table:takenBy ?Camera. /*Name of the camera*/
?Photo table:nameOfPhoto ?ImageName. /*Name of the Image*/
?Photo table:eventsOn ?Ceremony. /*Name of the ceremony*/
```

In the following line provide the information about the person with whom picture has been taken.

?Person rdf:type :Friends.

Photo album ontology contains the subclasses FamilyMembers, Friends, and Colleagues – different kind of people under the superclass Person. So if we take a picture, which can have different people (lets say with the colleagues) and the friends (in the same photo); we need to use this restriction. Because we do not need to pull out the names of other people (in this case the colleagues' name) other than the friend's name. If we do not use this restriction, then, it will show all names those who are in that photo, including the colleagues and the friend's name as well.

In this query, we did not use any restriction on camera. Because, the query is only concern about different types of cameras other than any particular type.

Following line is saying, whose information we are seeking here (in this case, name of the person is Eva)

?Person rdfs:label "Eva".

Following figures show the result of the case study 4.

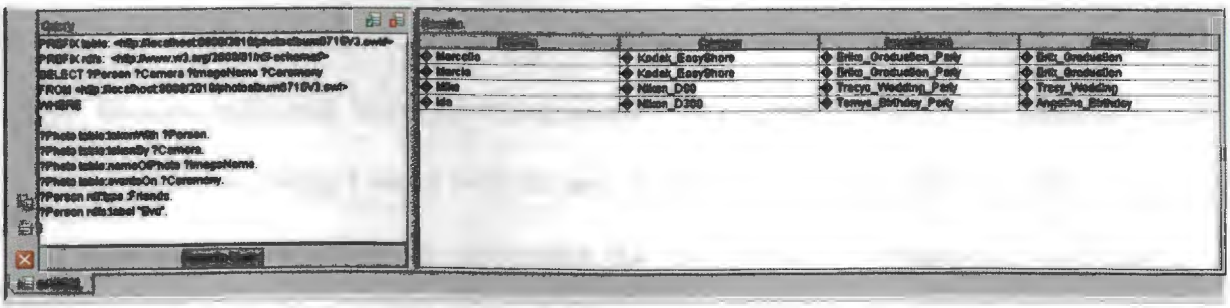
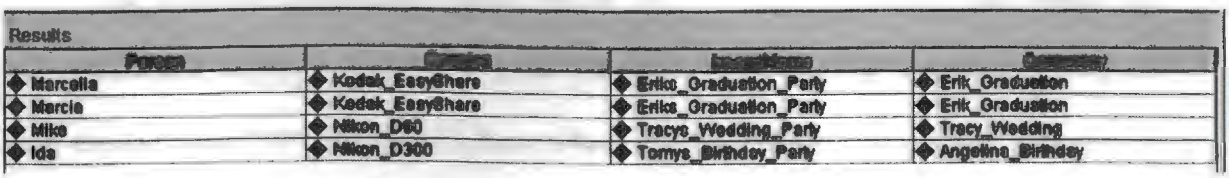


Figure 11(a): Result of case study 4



Results

Person	Camera	ImageName	Ceremony
Marcella	Kodak_EasyShare	Eriks_Graduation_Party	Erik_Graduation
Marcia	Kodak_EasyShare	Eriks_Graduation_Party	Erik_Graduation
Mike	Nikon_D60	Tracys_Wedding_Party	Tracy_Wedding
Ida	Nikon_D300	Tomys_Birthday_Party	Angelina_Birthday

Figure 11(b): Result of case study 4

CHAPTER 5: CONCLUSION

5.1. Summary

These days, digital cameras are considered the most well accepted image acquisition tools. People are using this technology as their valuable and necessary company, to sustain their personal information; in this case pictures that have been taken at different places, different moments and different times, etc. At the same time, availability of economical and different types of powerful cameras has been leading the society to take a huge number of pictures at a very shorter period time. So it has become necessary to manage those pictures efficiently, so that users can find their desirable pictures without any hastle. Nowadays, there are a number of economic software packages available to manage the photo albums, which are effective in regards searching the desire photos. [2] But, all these software have been built following the traditional approach, such as, key word based search, which sometimes make it difficult to get an accurate result based on the user's query. Because traditional keywords-based search, which organizes the most predictable words as tag format, doesn't match with the users provide knowable keywords. In that case, since machines are not able to understand the human readable language, users are experienced unnecessary result information from their searches [1].

But semantic technology, which can be used to make a machine understand the human readable language, with the help of complicated and worldly relationship [67], we used this technology to design basic photo album ontology. As an initial effort, protégé has been used to design the ontology, which involves classes, properties and individuals. To support the query, SPARQL has been used, in order to fulfill the users' query, to find the correct information regarding photo. This proposed solution can be extended by using more complicated relationship properties in order to enhance the power of managing photo albums.

5.2. Limitations and Future Work

There are a number of limitations that have been identified in this paper. Especially, any complicated reasoning has been omitted, which can be implemented as an extension of this current effort. That is, we didn't implement any critical reasoning to support our query. For example, if we consider the logic A<B, B<C and C<D and write the query like, "Find all the instances which are greater than A", then the query should be able to return B, C and D, not just B. Likewise, there are other characteristics of the properties such as functional, symmetric, asymmetric, etc., which can be applied for complicated reasoning which are not used in our ontology. Also we did not create any graphical user interface to show our results. We have used protégé editor to show our results for the queries. So linking between the data resources, created by protégé and SPARQL query language, results can be shown using a graphical user interface. In addition to this, in order to expand our ontology and acceptable in the WWW environment, we can also use the concept of FOAF and Dublin Core in future.

REFERENCES

[1] Brendan Elliott, Z. Meral Özsoyoğlu. "Annotation Suggestion and Search for Personal Multimedia Objects on the Web" Case Western Reserve University, EECS Dept.(2008): Web. September 1 2009.

[2] "Photo Management & Sharing" PCMAG, Web. August 13 2009.

[3] "Picasa" Google, Web. August 27 2009.

[4] "Flickr" Yahoo, Web. August 25 2009.

[5] "Windows Live Photo Gallery" Windows, Web. August 28 2009.

[6] Imran Shafiq Ahmad. "Text-based Image Indexing and Retrieval using Formal Concept Analysis" School of Computer Science, University of Windsor, (May 6 2008): Web. August 20 2009.

[7] Ritendra Datta, Dhiraj Joshi, Jla Li and James Z. Wang. "Image Retrieval: Ideas, Influences, and Trends of the New Age" ACM Comput. Surv. 40, 2, Article 5 (April 2008), 1-60. Web. August 9 2009.

[8] "Semantic Web" Wikipedia, (April 9 2008): Web. July 7 2009.

[9] Grigoris Antoniou, Frank van Harmelen. "A Semantic Web Primer" Massachusetts Institute of Technology, (2003): Web. June 2 2009.

[10] "Web Ontology Language" Wikipedia, (June 17 2003): Web. July 7 2009.

[11] "Multimedia" Wikipedia, (September 25 2008): Web. July 2 2009.

[12] "Multimedia" The Tech Terms Computer Technology, (2005): Web. July 7 2009.

[13] "Semantic Web" SearchSOA, (November 15 2006): Web. July 8 2009.

[14] "Definition of: Semantic Web" PCMAG, Web. July 8 2009.

[15] "Definition of: XML" PCMAG, Web. July 7 2009.

[16] "XML" Wikipedia, (August 6 2009): Web. July 7 2009.

[17] "XML schema" Wikipedia, (July 21 2004): Web. July 7 2009.

[18] "Resource Description Framework (RDF)" W3C, (February 10 2004): Web, August 15 2009.

[19] Dave Beckett. "Dave Beckett's Resource Description Framework (RDF) Resource Guide" planetrdf, (September 23 2005): Web. July 28 2009.

[20] "Definition of: RDF" PCMAG. Web. July 7 2009.

[21] RDF" Wikipedia, (May 27 2002): Web. July 9 2009.

[22] "Definition of: RDF Schema" PCMAG. Web. July 9 2009.

[23] "RDF Schema" Wikipedia, (June 2 2005): Web. July 9 2009.

[24] "Definition of: OWL" PCMAG. Web. July 17 2009.

[25] "W3C Semantic Web Activity" W3C. Web. July 17 2009.

[26] "Definition of: ontology" PCMAG. Web. July 20 2009.

[27] "Ontology" Wikipedia, (June 6 2002): Web. July 20 2009.

[28] "Facebook" facebook. Web. July 2 2009.

[29] "orkut" gmail. Web. July 6 2009.

[30] "Digital Photography" hp. Web. July 8 2009.

[31] Huan Wang, Song Liu, Liang-Tien Chia. "Image retrieval with a multi-modality ontology." Multimedia Systems (2008): Web. August 3 2009.

[32] E. Hyvönen, S. Saarela and K. Viljanen: "Ontogator: Combining Viewand Ontology-Based Search with Semantic Browsing." Proceedings of XML Finland, (2003): Web. August 3 2009.

[33] K. Yee, K. Swearingen, K. Li., and M. Hearst. "Faceted metadata for image search and browsing." In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, (2003): Web. August 8 2009.

[34] L. Hollink, G. Schreiber, J. Wielemaker, and B. Wielinga. "Semantic annotation of image collections." In Workshop on Knowledge Markup and Semantic Annotation, KCAP'03, (2003): Web. August 10 2009.

[35] A. T. Schreiber, B. Dubbeldam, J. Wielemaker, and B. Wielinga. "Ontology-Based Photo Annotation." IEEE Intelligent Systems 16, 3, (2001), pp. 66-74. Web. September 3 2009.

[36] Rui Y, Huang TS, Chang S-F. "Image retrieval: current techniques, promising directions and open issues." J Visual Comm Image Rep (1999): 10(1): 39-62. Web. August 22 2009.

[37] Smeulders AWM, Worring M, Santini S, Gupta A, Jain R. "Content based image retrieval at the end of the early years. IEEE Trans Patt Anal Machi Intel (2000): 22(12): 1349-1380. Web. September 20 2009.

[38] Lim, J.-H., Tian, Q., Mulhem, P. "Home Photo Content Modeling for Personalized Event-Based Retrieval." Multimedia 10(4), (2003): 28–37. Web. September 2 2009.

[39] Fung, C.Y., Loe, K.F. "Learning primitive and scene semantics of images for classification and retrieval. In: Proc of the 7th ACM Int'l Conf on Multimedia (part 2), pp." ACM Press, Orlando (1999): 1–9. Web. September 20 2009.

[40] Yang, S., Kim, S.-K., Ro, Y.M. "Semantic Home Photo Categorization." IEEE Transactions on Circuits and Systems for Video Technology 17(3), (2007): 324–335. Web. October 2 2009.

[41] "DART Home Page" An archive of AT&T Laboratories Cambridge, (2002): Web. October 2 2009.

[42] Mills, T.J., Pye, D., Sinclair, D., Wood, K.R.: Shoebox. "A Digital Photo Management System." Technical Report, AT&T, (2000): Web. October 10 2009.

[43] Sun, Y., Zhang, H., Zhang, L., Li, M. "MyPhotos – A System for Home Photo Management and Processing. In" Proceedings of the 10th ACM International Conference on Multimedia, Juan les Pins, France, (2002): 81–82. Web. December 2 2009.

[44] Flickner M, Sawhney H, Niblack W, et al. "Query by image and video content: the QBIC system." IEEE Computer, (1995): 23-32. Web. December 4 2009.

[45] Smith JR, Chang S-F. "VisualSEEk: a fully automated content-based image query system." Proceedings of the ACM International Conference on Multimedia, Boston, Massachusetts, (November 1996): 18-22. Web. December 8 2009.

[46] Pentland A, Picard RW, Sclaroff S. "Photobook: content-based manipulation of image databases." Int J Comp Vision, (1996): 233-254. Web. December 31 2009.

[47] Eakins JP. "Towards intelligent image retrieval. Patt Recog", (2002): 3-14. Web. January 1 2010.

[48] Barnard K, Duygulu P, Forsyth D, de Freitas N, Blei D, Jordan MI. "Matching words and pictures." J Mach Learn Res. (2003): 1107-1135. Web. January 20 2010.

[49] Antani S, Kasturi R, Jain R. "A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video." Patt Recog, (2002): 945-965. Web. October 10 2009.

[50] Wang JZ, Li J, Wiederhold G. "Simplicity: Semantics sensitive Integrated Matching for Picture LIbraries." IEEE Trans Pattern Anal Mach Intell, (2001): 947–963. Web. September 20 2009.

[51] Paek S, Sable CL, Hatzivassiloglou V, Jaimes A, Schiffman BH, Chang S-F, McKeown KR. "Integration of visual and text-based approaches for the content labeling and classification of photographs." In: Proc. ACM SIGIR workshop on multimedia indexing and retrieval, (1999): Web. October 9 2009.

[52] Salton G, Buckley C. "Term weighting approaches in automatic text retrieval." Inf Process Manage, (1988):513–523. Web. January 3 2009.

[53] Vailaya A, Jain A, Zhang H-J. "On image classification: city images vs. landscapes." Pattern Recog, (1998):1921–1935. Web. February 3 2010.

[54] Gevers T, Aldershoff F, Smeulders AWM. "Classification of images on Internet by visual and textual information." In: Proc. SPIE Internet Imaging (1999): 16–27. Web. February 10 2010.

[55] Lu Y, Hu C-H, Zhu X-Q, Zhang H-J, Yang Q. "A unified framework for semantics and feature based relevant feedback in image retrieval systems." In: Proc. ACM international conference on multimedia, (2000): 31–37. Web. February 10 2010.

[56] "welcome to protégé." stanford, Web. 11 May 2009.

[57] "what is protégé?" stanford, Web. 11 May 2009.

[58] Matthew Horridge. "A Practical Guide To Building OWL Ontologies Using Prot´eg´e 4 and CO-ODE Tools Edition 1.2" The University Of Manchester (March 13 2009): 1-108. Web. 11 May 2009.

[59] "OWL Web Ontology Language Reference." W3C, (10 February 2004): Web. 11 May 2009.

[60] "Introduction- An API for OWL." GNU Lesser General Public License (LGPL), Web. January 10, 2010.

[61] Fabien, Gandon, Inria. "SPARQL, in a nutshell" Slideshare, (2007): Web. February 3, 2010.

[62] "SPARQL" Wikipedia, (August 2005): Web. October 4 2009.

[63] Lee Feigenbaum, Eric Prud'hommeaux. "SPARQL By Example A Tutorial" Cambridge semantics, (2008): Web. May 6 2010.

[64] Leigh Dodds. "Introducing SPARQL: querying The Semantic Web." xml, (November 16 2005): Web. October 1 2009.

[65] Holger Knublauch, Ray W. Fergerson, Natalya F. Noy and Mark A. Musen. "The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications" Stanford Medical Informatics, Stanford School of Medicine, (2004): Web, June 12 2010.

[66] Matthew Perry, Amit P. Sheth, Prateek Jain. "SPARQLST: Extending SPARQL to Support Spatiotemporal Queries" Wright State University Dayton, OH, USA, (November 2 2009): Web. February 3 2010.

[67] "Natural language processing" Wikipedia. (June 9 2006): Web. September 3 2009.

[68] "FOAF (software)", Wikipedia, (September 10 2005): Web. January 1 2010.

[69] "FOAF Vocabulary Specification 0.98", xmlns (August 9 2010): Web. January 2 2010.

[70] Roger L. Costelb. "Camera Ontology" The MITRE Corporation. (2003): Web. March 20 2010.

[71] "Dublin Core". Wikepedia. (April 22 2010): Web. May 2, 2010.

[72] "Resource Description Framework (RDF)", Rensselaer.Web. April 30 2010.

[73] "The Dublin Core® Metadata Initiative", dublincore. Web. April 30 2010.