

The Human Face Semantic Web

Hamido Hourani, Mohammed Al Rawi, Abd El-Latif Abu Dalhoum, and Sabina Jeschke

Abstract— The vast development in the field of multimedia technology generates huge amount of data which result in difficulties for users to find their data on demand. Therefore, the process of efficiently storing and retrieving multimedia data is a very important task. In this paper, we use the Semantic Web notion to annotate and retrieve digital images; where we introduce two stages for annotating these images and two other processes for retrieving these images. The first stage is the dynamic annotation (absolute) in which the resultant annotation data may be stored inside the image. The other annotation is the manual (relative) in which the resultant annotation is stored in a separate file to be linked in some way to the image file. To retrieve those images, we build a Semantic Web search engine for retrieving the annotated images, and this search engine in its turn comprises of two steps: the first depends on the query entered by the researcher, and the second depends on the image which is selected from the previous step which is used as a seed for a second search round to retrieve the images that relate to the seeded image. To reduce the impact of differences of the interpretations among different annotators, we suggest two types of annotation attributes coarse and fine. We approach this process via implementing a case study based on Human Face images which is a very important task that has wide range of applications, e.g., crime investigation, relatives search, partners matching and finding, etc.

Using face images of more than 450 persons, the proposed human face semantic web search engine (HFSWSE) is evaluated through several experiments. The results of these experiments are encouraging; since we are able to mitigate the impact of differences of interpretations among the annotators. Compared to typical face recognition techniques, the HFSWSE obtains the retrieved set of the query image in a short time, and the accuracy that depends on the number of used attributes is more than good.

Keywords—Face Recognition, Image Retrieval, Search Engine, Semantic Web.

I. INTRODUCTION

In recent years, the amount of information available on the Web is exponentially growing, but the information being retrieved or consumed grows at best linearly. This happens due to the methods and techniques used today, which are not good enough to handle all available information in an

effective manner. On the other hand, this information is made for human processing and the only responsibility for the computer is to display it in a proper way. Therefore, the computer is unable to give the appropriate help for the researchers. The same case happens for images as the number of digital capturing tools is increased which makes the publishing and sharing digital media on the web more easily. As a result, finding the proper media on the web or on the personal computer becomes a tedious work [18]. In addition to the huge number of images on the web or on the personal computer, there is a problem of how to retrieve these images with high precision in an effective manner. Most of the techniques are focused on retrieving these images using the low level feature such as color, shape and spatial relationship [1], [2] which are not taken into consideration by normal people. Those people are mostly concerned with retrieving their images using their high level features such as the content of these images and not the properties of the image itself [17].

Several solutions were put forward to solve these problems; one of them is the text based where the owner of the image describes his images using the natural text language and stores this text beside the described image in the same database. The problem of this technique is the ambiguity, where the retriever (the person who submit a query to a search engine) cannot know what the annotator means with this text [3]. Just the same, we find ourselves stuck in the same problems that occur when retrieving documents, which is retrieving irrelevant data after submitting a certain query [19]. In addition, an image alone does not have any description, so we miss the portability aspect of this image. Even if we had annotated these texts inside the image itself, there is no widely used standard for dealing with these texts. Hence, the compatibility of dealing with those texts is lost. Despite the existence of standardization activities of International Organization for Standardization (ISO) and other related communities, they are not widely used in retrieving the annotated texts from the images. This is mainly because there are insufficient applications that would benefit from their use, and because of the complexity of some of these standards makes the multimedia annotation unnecessary difficult [4]. Some examples of these standards are the MPEG-7 [5], MPEG-21 [6], and Dublin Core [7]. Consequently, we dramatically lose the semantics when sharing these images among computers.

How the problem is snowballed? Most of these annotations are written for human reasoning and interpretation, so the computer does not have anything to offer but presenting images and transferring them. Therefore, we need to define the annotation in a well defined meaning in order to give the

Manuscript received March 13, 2008; Revised version received July 31, 2008.

Hamido Hourani is with the Institute of Information Technology Service, Stuttgart, Germany; e-mail: hamido.hourani@iits.uni-stuttgart.de.

Mohammed Al Rawi is with the King Abdullah II School for Information Technology, Computer Science Department, University of Jordan, Amman, Jordan; e-mail: rawi@ju.edu.jo.

Abd El-Latif Abu Dalhoum is with the King Abdullah II School for Information Technology, Computer Science Department, University of Jordan, Amman, Jordan; e-mail: a.latif@ju.edu.jo.

Sabina Jeschke is with the Institute of Information Technology Service, Stuttgart, Germany; e-mail: sabina.jeschke@rus.uni-stuttgart.de.

computers the chance to help us in searching for these documents or images. For that reason, the Semantic Web comes to the fore, which is elaborated by Tim Berner-Lee in [8] as an extension of the current web in which information is giving a well defined meaning, better enabling computers and people to work in cooperation. To this, the role of the Semantic Web is to provide a well defined meaning of the domain of the application and formally describe the concepts found in that domain, and a well defined meaning for the relationships between these concepts and their properties. Consequently, metadata can be shared, reused and exchanged among computer applications.

In this paper we introduce a process for annotating and retrieving images based on Semantic Web. The annotation process divides into two steps, the first step done dynamically by reading the information about images from a database and converting it according to the semantic web standards and recommendations. After that, it annotates the converted information inside the image. This gives an image the portability aspect so that it could be shared without missing its semantic. And we specify the data that should be stored inside an image to be independent from the annotator point of view. The second step annotates the images manually through a system we build. Then, we store the annotation which is based also on the Semantic Web recommendations outside the image header to improve the performance in the retrieval steps. That is as far as the annotation process is concerned. As for the retrieval process, it is divided into two steps; the first step retrieves images based on the query that a user has already input. The second step retrieves images based on the selected image by the user.

To get rid of some of the problems of annotation such as the differences of interpretations of the images among the annotators, we build the values of the attributes as tree (hierarchy) where the leaf nodes are used in the annotation and the parent nodes are used in the retrieval. So, when we want to annotate an image we choose one of the leaf nodes where each set of these leaf nodes belong semantically to one of the parent nodes, that is for the annotation process. In the retrieval process, we use the parent nodes in the first part, based on a user's query, and use the leaf nodes in the second part, based on the selected image from that user. We call this kind of values for the attributes the coarse values, while we call the traditional values the fine values. The remainder of this paper is structured as follows: Section II discusses the related works. Section III lists the human facial characteristics. System architecture is introduced in section IV. Section V presents the annotation stage. Section VI presents the retrieval stage. Section VII analyses the experimental results. Finally section VIII concludes the paper.

II. RELATED WORKS

Hyvonen et al in [9] show how ontologies can help a user in formulating the information need, the query and the answer. They elaborate the problems that appear from using keywords

based search, which are,

- Formulating the information need, which comes when the user doesn't know what question to ask, how can they help the user to focus within the keywords used in the database contents.
- Formulating the query, the user can't necessarily figure out what keywords he should use to formulate a certain query.
- Formulating the answer, retrieving images according to the searched keywords may miss the most interesting aspect of the repository, since the images are related to each other in many interesting ways.

They elaborate the common ways to annotate images, which are,

- Keywords, which are used to describe and index the images to enhance of retrieval results.
- Classification, annotate images by setting them in categories that describe them, so when we want to retrieve an image, all the images that are related to that image category will be recommended to retrieve as well.
- Free Text Description, the way used the keyword based search in the background.

Semantic Web ontology techniques and metadata languages are used to give these classifications a meaning with well defined semantic and a flexible data model for representing metadata descriptions. In their work [9], they store these ontologies outside of the image. Their retrieval approach divides into two stages. The first stage, the View-Based Search retrieves the annotated images according to user query. The second stage, Semantic Browsing retrieves related images to the selected image.

In [10], Hliaoutakis et al investigate approaches to compute the semantic similarity among terms such as the natural language and the medical terms. Semantic similarity relates to computing the similarity between two terms in conceptual level (Ontology) but not necessarily lexical similar terms. They depend of this notion to build Semantic Similarity based Retrieval Model, capable to discover similarities between documents containing conceptually similar terms. They claim that this model has promising performance improvements over classical information retrieval methods that used the plain text matching. In our approach we employ that conceptual level to mitigate the differences of interpretations among the annotators.

In our approach, we use two types of annotations. The first type is stored inside the image, while the second type is stored outside the image. In addition, we use two types of ontology instances, to reduce the impact of differences of interpretations among annotators. In the retrieval approach we employ the idea of the view based and the semantic browsing from [9] with some modifications, where we improve the performance of them.

III. HUMAN FACIAL CHARACTERISTICS

In this section we list the facial characteristics which we use

in this case study. These characteristics consider the main characteristics for the human face which is taken from [11]. Table I lists them with their candidate values.

Table I: Human Facial attribute

Facial Characteristic	Candidate values
Face Shape	Square, Round
Skin Color	Black, very dark brown, dark brown, brown, light brown, Tan, Pale face
Hair Type	Curly, wavy, straight
Hair Color	Black, dark brown, brown, light brown, blonde, light blonde, strawberry blonde
Chin Prominence Shape	Very Prominent, Less Prominent
Chin Shape	Round, Square
Cleft Chin	Present, Absent
Color of Eyebrows	Very Dark, Dark, Light
Eyebrow Thickness	Bushy, Fine
Eyebrow Placement	Not connected, connected
Eye Color	Dark Brown, Brown, Light Brown, Blue with some brown, dark blue, blue, light blue.
Eye Distance Apart	Far, Average, Close
Eye Size	Large, Medium, Small
Eye Shape	Almond, Round
Eye's Slantness	Horizontal, Slanted
Eyelash Length	Short, Long
Mouth Size	Long, Average, Short
Lip Thickness	Thick, Thin
Lip Protrusion	Very, Slightly, Absent
Nose Size	Big, Medium, Small
Nose Shape	Rounded, Pointed
Nostril Shape	Rounded, Pointed
Ear lobe attachment	Free, Attached
Darwin's Earpoint	Exist, Absent
Ear pits	Exist, Absent
Hairy Ears	Exist, Absent
Cheek Freckles	Exist Absent
Dimples	Exist, Absent
Forehead Freckles	Exist, Absent
Widow's Peak	Exist, Absent

Note that these facial characteristics are inherited from the parents to their children according to Inheritance Rules and the whole of these facial characteristics take only the specified values which it is listed in Table. I.

IV. SYSTEM ARCHITECTURE

Our approach is divided into two steps; Annotation Step and Retrieval Step. In the annotation step, Dynamic Stage and Manual Stage are used. The retrieval step is divided into two stages; user query based retrieval and image based retrieval. In this architecture the two steps work independently and interact

with one another through the annotation files (external and internal).

V. ANNOTATION STEP

The aim of this step is to annotate the images with metadata according to Semantic Web format, to prepare these images for the retrieval steps. Part of this step is done automatically, whereas the second part needs a human cooperation to annotate these images manually. The part of the resulted annotations which is in RDF syntax, are stored inside an image to give it the portability aspect, when it is shared or distributed over the web. The characteristics of this part of the annotations are to be unchangeable, and describe the absolute things, like the name of the artist and the title of the painting. And avoid describing relative things which are changed from person to person such as the strength, thickness, and length. This part is usually done dynamically by reading the data from the database and storing it inside an image after converting it according to the Semantic Web recommendations. The other part of annotation is stored outside an image to improve the performance and the flexibility of the retrieval step. This part of annotation is done manually using a system built for this purpose.

A. Dynamic stage

Dynamic stage is the first stage of the annotation step. This stage is carried out by reading the data that are related to a specific image from a database, and convert it to the Semantic Web standard using a component we build it for that purpose. After that, it stores the resulted annotation inside the image header. The aim of storing these annotations inside an image is to give it the portability aspect when an owner wants to share his image with his friends or when he wants to publish it over the internet. Anyone can receive this image and read the annotation using any application supports the Semantic Web, or use this image directly inside their Semantic Web application by mapping their ontology with the ontology which is stored inside that image.

The candidate data that are preferred to be stored inside images are the data that describe something absolute and independent from the annotator's point of view (absolute data). In other words, something that has just this value such as the name of the persons who is appeared in this image, date of birth, the number of persons in that image, or the name of the artist of this painting. The purpose of that is to give an image the portability aspect in the annotation, and to make these annotation permanents for this image. Considering the human face pictures, the candidate data are the name of the person, date of birth, security national number, gender, and race.

The programming component is built from scratch using Java language to implement the process of storing the annotation inside images. This component is mainly based on *javax.imageio* package (Image IO Java Package, 2007), which supports at least two metadata formats (*Metadata* is referred in *javax.imageio* package to the data stored in an image file that

does not represent the actual pixel values) for each known image format such as JPG, GIF, and PNG. The first format that we use in our approach, is a common neutral format called *javax_imageio_1.0*. The other one is highly specific native format that exposes all internal structure of the metadata for a specific image format [12]. Since these metadata may contain complex and hierarchical structure, Java language represents them as an XML Document Object Model (DOM) [12].

According to the Document Type Definition (DTD) for the neutral metadata format *javax_imageio_1.0*, it contains a root node called "*javax_imageio_1.0*" which has children nodes "*Chroma*", "*Compression*", "*Data*", "*Dimension*", "*Document*", "*Text*", and "*Transparency*". The complete DTD for this metadata can be seen in [13]. The fig. 1 shows a sample for the neutral metadata format, as XML syntax.

```
<javax_imageio_1.0>
  <Chroma>
    <ColorSpaceType name="YCbCr" />
    <NumChannels value="3" />
  </Chroma>
  <Compression>
    <CompressionTypeName value="JPEG" />
    <Lossless value="false" />
    <NumProgressiveScans value="1" />
  </Compression>
  <Dimension>
    <ImageOrientation value="normal" />
  </Dimension>
  <Text>
    <TextEntry keyword="comment" value="" />
  </Text>
</javax_imageio_1.0>
```

Fig.1 Sample for neutral metadata format.

In our component we use the neutral metadata format to make our approach applicable on all images format. The fig. 2 shows an overview of this component "*ImageAnnotation*".

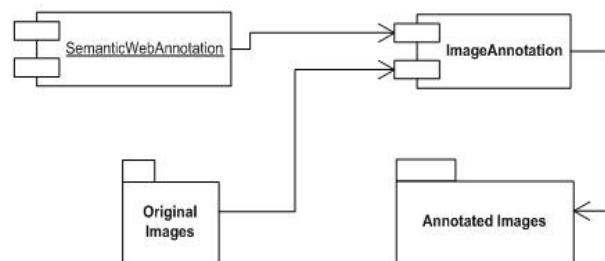


Fig. 2 An overview of the ImageAnnotation Component.

As we see from the fig. 2, the *imageAnnotation* component takes Semantic Web annotation of a specific image and takes a reference for that image. After that it puts the annotation which is in RDF syntax inside a copy of that image. According to the Java neutral metadata format, we choose to put our semantic annotation inside the *value* attribute of the *Text's* child node *TextEntry* tag. Since RDF syntax is based on XML syntax which contains symbols that are not allowed to be used as a value for the XML tag attributes, we cannot put it

directly as a value inside the *value* attribute of the *TextEntry* tag. Consequently, we decode the RDF syntax by converting those symbols to other neutral symbols. Table. II displays those symbols. After that we put the annotation as a value for the *value* attribute of the *TextEntry* tag.

Table. II RDF Symbols with its equivalents Neutral Symbols.

RDF Symbol	Neutral Symbol
<	<
>	>
"	"
'	´

The *ImageAnnotation* gets the Semantic Web annotation from the *SemanticAnnotation* Component as it is presented in fig. 2. *SemanticAnnotation* is used to convert the data from database to annotation according to the Semantic Web recommendations. The annotation which is generated from this component contains the instances of the OWL classes, in addition to the definition of those classes. Despite that the common way is to separate those instances from its classes; however, we follow this merging approach to give each image the independency and the portability aspects, through storing the annotation inside an image; where the annotation itself contains the ontology definition and the instances which derived from that definitions. The purpose of this definition is to be used on the ontology mapping.

The *SemanticAnnotation* component is based on Jena API which is a Java framework for building applications for the Semantic Web. Jena is open source and grown out of work with the HP Labs Semantic Web Program (<http://www.hpl.hp.com/semweb/>) [14].

B. Manual stage

Manual stage is the second stage of the annotation step, which is done manually by an annotator through the application we build for this purpose. In this stage the annotator describes the features that are dependent on the annotator point of view (relative data) such as the face shape of the person who is appeared in an image, the degree color of his eyes and soon on. The output of this stage is converted to RDF format and it is stored outside the image. The aims of storing the output of this stage outside the image are; 1) to give the retrieval part more efficiency by reading the RDF directly without needing more steps further for extracting it from the images. 2) To keep the size of the annotated image as small as possible; since the size of the output data from this stage is relatively huge. Finally, because these annotations are based on point of view of the annotator, so we cannot put them inside images where we just keep the absolute data inside them. For that, we put the annotation outside an image and keep a reference between the resulted annotation and it is

related image.

Because these annotations which depend on the annotator point of view and suffered from the impact of the differences of interpretations among the annotators, make the retrieval part depending on the annotator point of view. In order to mitigate this situation we build the ranges of the attributes based on the OWL relationships "owl:subclass". To clarify this method considers the fig. 3 which presents an example.

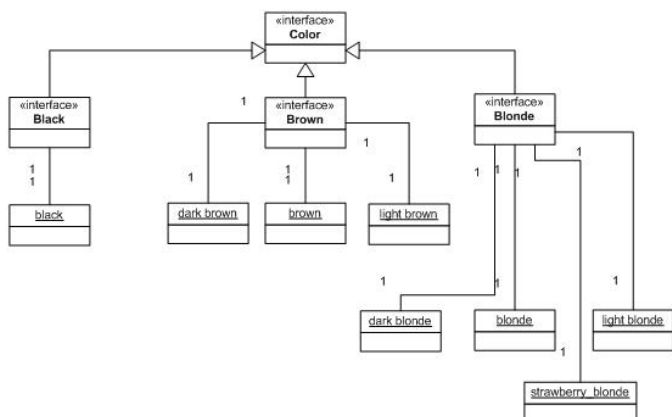


Fig. 3 An example of the color value Ontology.

Suppose we have an attribute Hair Color, that an annotator uses it to describe images, the range of this attribute is the ordinary human hair color. Using our method we enforce the annotator to select one of the leaf nodes which are displayed in fig. 3 as a value of the *Hair Color* attribute for the person who is appeared in the image. The annotator will select the approximate degree of color for that hair such as dark blonde, blonde, light blonde, and strawberry blonde. Despite the differences of interpretations among the annotators, they will agree that this hair has a blonde's degree color. In the retrieval stage we use the abstract parent in the first stage and the leaf nodes in the second stage.

Those two stages separate the ontology domain; one of them is stored inside the image itself, and the other is stored outside the image. Since we need the portability aspect for images, we make the ontology which is stored inside the image the owner of the main ontology and the other ontologies are related to it by mapping their ontology to the ontology which is stored inside the image. For this purpose we stored the ontology classes with the instances inside an image in the previous stage. To give the output of this stage the flexibility, we separate between the ontology classes and the instances in files. In other words, we have one file for the ontology classes and multi files for the instances which contain the output annotation in RDF format. As a consequence, we can modify the ontology classes file just once and the result reflects to all instance files without needing to access each file, as in case if the owl classes are stored with the instances.

The programming component is built to manipulate this stage by using Java language and is based on Jena API. This

component takes the descriptions which are entered by an annotator and converts it to RDF format, in the same time, gets the reference from the metadata which is stored inside the image, to use it as a reference inside the RDF format which is resulted from this step. Fig. 4 shows an overview for this component with the other component.

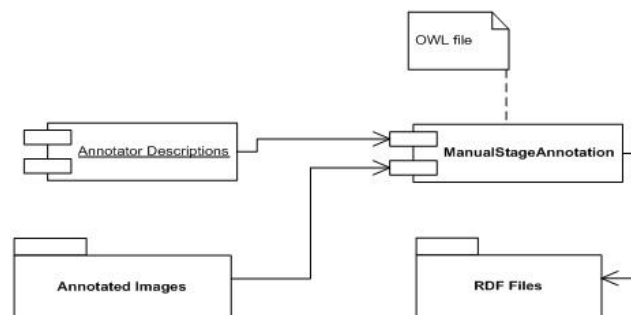


Fig. 4 An overview of the ManualStageAnnotation component.

As we see from this figure, the *ManualStageAnnotation* component takes its input from an annotator's description to convert it to RDF format. In addition it takes the reference from the annotated images, which is needed to make the association between the image with its description, and using the OWL file which contains the ontology classes which is needed to create the instances. The output is the RDF file which contains the instances of the descriptions according to the OWL classes.

VI. THE RETRIEVAL STEP

The aim of this step is to retrieve the images which are semantically annotated in the previous step. This step is divided into two stages; the first one retrieves the images according to the query which a user has entered, and the second one retrieves the images that are related to the image which the user is selected. One of the techniques we introduced in the manual stage (section V.B) to mitigate the difference of interpretations was by building ranges in hierarchy where an annotator annotates an images using the leaf nodes and a retriever retrieves those images using the parents for these leaf nodes.

A. Query based

Query Based stage is the first stage of the Semantic Retrieval, which is done by the user who searches for an image. To find an image, the user needs to describe it using an application which is built for that purpose. The user describes the image that he is looking for by using the parent nodes of the values. To clarify it, consider the fig. 3 and back to our Hair Color example. As we mentioned before, we enforce the annotator to describe the Hair Color of the person who is appeared in the image by one of leaf nodes of color degree such as dark blonde, blonde, light blonde, strawberry blonde. That is done in the annotation step. For the retrieval step especially in *Query Based* stage we enforce the user to

describe an image using the parent node which the annotator used one of its leaf nodes. In our example the user describes the hair color of the person who is appeared in an image by using one of the parent colors such as the black, brown and blonde.

According to this technique, the resulted images from this query will be the set of images that have the same parent color. In case a user selects the blonde color, the result will contain the images which have persons with the dark blonde, blonde, light blonde and strawberry blonde. In other words, all leaves that are from the selected parent node will be retrieved. Fig. 5 shows a sample structure of ontology which is displayed in fig. 3 in OWL syntax.

```
<owl:Class rdf:ID="Color"/>
<owl:Class rdf:ID="Blonde">
  <rdfs:subClassOf
rdf:resource="#Color"/>
</owl:Class>
<Blonde
rdf:about="&range;dark_blonde"/>
<Blonde rdf:about="&range;blonde"/>
<Blonde
rdf:about="&range;light_blonde"/>
<Blonde
rdf:about="&range;strawberry_blonde"/>
```

Fig. 5 Sample for the structure in OWL Syntax

As it is seen from fig. 5, we create a class called Color as a root for our color range hierarchy, and we create another class called Blonde as a subclass from the root class and which it is used by a user in the first retrieval stage. After that, we create four instances from the Blonde class which are; dark blonde, blonde, light blonde, and strawberry blonde. These instances are used by an annotator in the second semantic annotation stage.

To increase the performance of the system, we first check the matching of a user's query with the external annotation of the images, which is stored outside the image, as the result of this matching we get a subset from the complete set of the images *set1*. From *set1*, we extract the semantic annotations from them and compare these annotations to the query part that is related with the internal annotations. When the matching is occurred we get a subset from that set, *set2*. The finally subset is showed to the user. And the selected image will be used as a seed for the next stage *Image Based* stage. Fig. 6 shows the pseudo-code which describes this process.

B. Image based

The set of the retrieval images from the previous stage may contain a large number of images. That because we are used the parent nodes in query based stage. The opportunity to have the wanted image in this set is very large. To find the wanted image, a user needs to browse all these images which may a hundred of images until he finds his wanted image, or he can select the most similar image that have the same feature of the wanted image and make it the seed for the *Image Based* stage.

```
Query_based ( user_sem_annot, images_set )

  set1 ← parent_node_match (
        user_sem_annot,
        external_annot( images-set )
    );

  inside_img_sem_annot ←
  extract_relatedPart ( user_sem_annot )

  set2 ← match ( inside_img_sem_annot,
    images ( set1 ) )

  return set2
end
```

Fig. 6 Query Based pseudo-code

Image Based stage is the second stage of the Retrieval Step, and it is done by an application user. This stage takes its input from the previous stage, and uses the leaf node in its matching process. Once the user selects an image from the result set of the Query Based stage, the Image Based extracts the external annotation for that image and uses it to find images that relates semantically to this image like having the same description. This stage uses the leaves node in matching process. After that, displays the selected image with its internal and external annotations in read only manner, in addition to display the set of images that are resulted in the leaves node matching process. From this set of images that user can select anyone of them, to begin another round of leaves node matching. Fig. 7 shows a pseudo-code that describes this process.

```
image-based ( selected_img, img_subset )

  ext_annot ← extract_sem_ext_annot (
    selected_img );

  img_subset_ext_annot ←
  extract_sem_ext_annot ( img_subset );

  set1 ← child_nodes_matching ( ext_annot,
    img_subset_ext_annot );

  end
```

Fig. 7 image-based pseudo-code

As it is seen from fig. 7, the matching is done with all images that are retrieved from the Query Based stage. In other words, the result from this algorithm is a subset from the set that is resulted from the Query Based stage. We clarify the goal from this by getting back to our Hair color example. Suppose a user retrieves all images that have a person with blonde hair color in Query Based stage. According to this we have a subset *set1* contains all the images which have a person with blonde degree hair color. The user from *set1* selects an image *image1* to be a seed for the next stage *Image Based* stage. In this stage the algorithm uses the leaf node range in matching, for example uses the leaf node strawberry blonde in matching instead of using the parent node blonde. As consequence, all the images in the *set1* that have matching

with the same degree of hair color will retrieved, in addition to the matching with the other attributes.

VII. EXPERIMENTAL RESULTS

In this section we present several experiments which we run on our application. Each of these experiments are designed and performed to measure a specific metric such as the impaction of the annotator's point of view on the image retrieval, the number of used attributes impact on a retrieval set, and the impaction of running a query on the external annotations before running it on the internal annotations of an image on the performance of the application. An analysis and discussion of the results are followed each of those experiments. To perform these experiments we used around the 450 distinct personal images which we took the most of them from [15] [16]. In addition, we used three different persons to annotate these images. In this application we have 25 attributes that have values from fine type, and 5 attributes that have values from coarse types.

A. Experiment 1

The goal of this experiment is to see if users can find their wanted images using this application despite the difference of interpretations among annotators. We request from two persons to retrieve five images, which are displayed in Table. III, using our retrieval application. One of those persons participated in the annotation for these images. The statistical results of this experiment are displayed in table. IV and V for expert user who participated in the annotation stage, table. VI and VII for trained user. And a sample from this result is shown in table VIII.

Table. III The five query images which are used in the experiment.

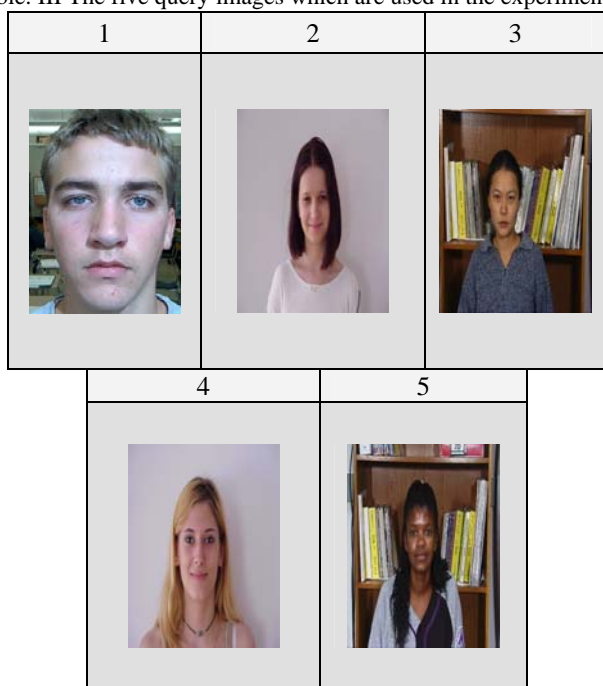


Table. IV Experiment results for expert user on finve different images. Where we request from this user to retrieve the query image by using the set of attributes which have coarse detail values only.

Query Image	Number of retrieved images out of 450	The existence of the wanted image
1	11	Yes
2	6	Yes
3	38	Yes
4	10	Yes
5	19	Yes

Table. V Experiment results for expert user on five different images. Where we request from this user to retrieve the query image by using set of attributes which have fine detail values, in addition to the set of attributes which have coarse detail values.

Query Image	Number of fine details attributes that are similar to the query attributes. This is the intersection between all the fine details attributes	Number of retrieved images out of 450	The existence of the wanted image
1	24	1	Yes
2	24	1	Yes
3	19	1	Yes
4	24	1	Yes
5	22	1	Yes

Table. VI Experiment results for trained user on five different images. Where we request from this user to retrieve the query image by using the set of attributes which have coarse detail values only.

Query Image	Number of retrieved images out of 450	The existence of the wanted image
1	11	Yes
2	6	Yes
3	38	Yes
4	10	Yes
5	19	Yes

Table. VII Experiment results for trained user on five different images. Where we request from this user to retrieve the query image by using the set of attributes which have final detail values, in addition to set of attributes which have coarse detail values.

Query Image	Number of fine details attributes that are similar to the query attributes. This is the intersection between all the fine details attributes	Number of retrieved images out of 450	The existence of the wanted image
1	23	1	Yes
2	21	1	Yes
3	19	1	Yes
4	21	1	Yes
5	24	1	Yes

Table. VIII Sample results from experiments.






Query Image			
Retrieval Images			
			

Table. IV and VI consist of three columns; the first column refers to the query image which we need to find the information about it by using our application. The second column refers to the number of the retrieval images according to a user query. And the third column indicates if a wanted image is existed in the set of images which is fetched. The purpose from these two tables is to show if a user can retrieve his wanted image despite the difference of interpretations between him and the annotator who annotated the wanted image, and how our proposed idea about using the attributes that have coarse values is able to mitigate these differences of interpretations. According to the result from these tables IV and VI, the users were able to retrieve the query image using

the whole set of attributes that have coarse values, where the number of the attributes that have coarse values is 5 attributes. From this result we are able to mitigate the differences of interpretations among the annotators.

Tables V and VII consist of four columns; the first column refers to the query image which we need to find the information about it by using our application. The second column refers to the number of the attributes that have fine values which a user was able to use to retrieve the wanted image; this number is calculated by making the user describes the query image by using the whole attributes that have fine values, where the number of these attributes is 25. After that we take the intersection between the user description and the annotator description based on the attribute matching and we suppose that the annotator has the right description. The third column refers to the number of the retrieved images according to the user query. And the third column indicates if the wanted image is existed in the set of images which is fetched. The purpose from these two tables is to show the impact of the differences of interpretations between retriever and the annotator who annotated the wanted image on the retrieval of the images.

Considering the first two tables IV and V, that show the experiment which was done by the expert user who had participated in the previous annotation step for these images. This user according to those tables IV and V was able to find all his wanted images using the whole attributes which have values from the coarse type, but with number of retrieval images large relatively. Whilst, that user can find his wanted images by using in average around 90% from the attributes which have values from the fine type. In other words, this user is not able to use all of these attributes since the differences of interpretations between that user and the annotator who annotated that image. Despite the fact that this user can use the 90% of the attributes, the number of the retrieval images is equal to one. The average number of the attributes which have traditional values and the user was able to use is 23, and the average of the attributes which have coarse values and the user was able to use is 5. Therefore, the probability for two persons to have the same annotation in their $(23 + 5)$ attributes is $1/(2^{28})$, for that it is seldom to get the number of retrieval images larger than 1. In tables VI and VII we have the same results but with around 86% instead of 90% from the attributes which have fine values that the trained user is able to use.

In general, users are able to find their wanted images by using attributes that have coarse values, with accuracy reach the 100%, and by using attributes that have traditional values with accuracy around the 88%. Note that the query image which has number 3 has the lowest recognition degree which is 19. That because the difference of the race between the retriever and the person who is appeared in that image, which makes the description process for like these images difficult. For that, it is recommended to have annotators with different races to annotate images, where each annotator annotates images which have the same race with him.

B. Experiment 2

The goal of this experiment is to show the relation between the numbers of the attributes used to retrieve a wanted image, with the number of the retrieval images. In this experiment we choose the attributes and its related values randomly using our retrieval application. We run the experiment first on the attributes that have fine values. After that, we run the same experiment but this time using the attributes that have coarse values. The statistical results of this experiment are displayed in tables IX and X. Fig. 8 and 9 display these statistical results in charts.

Table IX: Relationship between the number of the used attributes that have fine values with the number of retrieval Images.

Number of the used attributes	Number of the retrieval Images	Attributes were used (Accumulative)
0	449	Not Used
1	321	Face shape = round
2	126	Chin Prominence = very
3	97	Chin shape = round
4	15	Cleft Chin = exist
5	9	Eyebrow Thickness = fine
6	9	Eyebrow Placement = not connected
7	6	Eye distance = average
8	4	Eye Size = average
9	3	Eye shape = almond
10	1	Mouth Size = average

Table. X Relationship between the number of the used attributes that have coarse values with the number of retrieval Images.

Number of the used attributes	Number of the retrieval Images	Attributes were used (Accumulative)
0	499	Not Used
1	284	Skin Color = white
2	217	Hair Type = straight
3	54	Hair Color = Blonde
4	32	Eyebrows Color = light
5	26	Eye Color = Blue

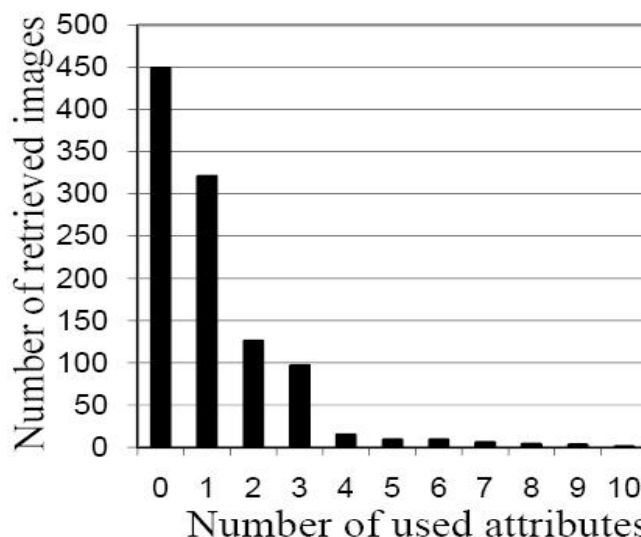


Fig. 8 Relationship between the number of the used attributes that have fine values with the number of retrieval Images.

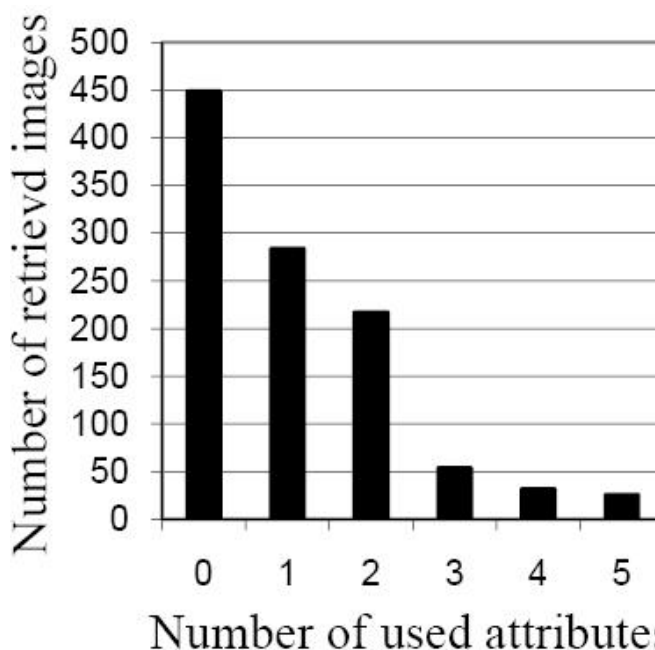


Fig. 9 Relationship between the number of the used attributes that have coarse values with the number of retrieval Images.

As we see from the fig. 8, the relationship between the number of the attributes used and the number of the retrieval images are inverse relationship in general. The number of images is decrease sharply from 0 to 2 and become less sharp between 2 and 3, and get back to its sharp state between 3 and 4, and finally become almost steady from 4 to 10. There are two reasons for this behavior; the first one depends on the number of permutations for the values of the attributes, and the second one depends on the fact that we are dealing with inheritance characters. As a consequence, there are dominate characters and recessive characters which are govern by the inheritance rules where the probability that the dominate character appears is 3/4 of the probability of the recessive character. This reason explains what happened between 0 and

1, where we used the round value (dominate character) for the shape of the face and we got 321 images. In contrast, if we were use the square value (recessive), we would get 128 images.

C. Experiment 3

The goal of this experiment is to show how the order of performing a query may affect the performance of the application, and why we choose to execute the query first on the external annotation and then on the internal annotation. For this experiment we build a program that perform a simple query on the external annotation and register the time period that is needed to extract the result from these annotations, and make the same experiment on the internal annotations too. Fig. 10 shows the query which is used on the External annotation and the query which is used on the internal annotation. And fig. 11 shows a comparison between the execution time for these statistical tables.

```
//SPARQL Query to extract the ssn from the
External (Relative) annotation
SELECT ?ssn
WHERE {
  ?descriptor <
  http://www.ju.edu.jo/cs/thesis/2007/owl/face#hasFaceShape>
  <
  http://www.ju.edu.jo/cs/thesis/2007/owl/range#round> .
  ?humanFace <
  http://www.ju.edu.jo/cs/thesis/2007/owl/face#hasDescriptor>
  ?descriptor .
  ?humanFace <
  http://www.ju.edu.jo/cs/thesis/2007/owl/face#hasMajorSuper>
  ?human .
  ?human <
  http://www.ju.edu.jo/cs/thesis/2007/owl/face#hasSSN> ?ssn
}

//SPARQL Query to extract the ssn from the
Internal (Absolute) annotation
SELECT ?ssn
WHERE {
  ?person
  <http://www.ju.edu.jo/cs/thesis/2007/owl/person#hasGender>
  <http://www.ju.edu.jo/cs/thesis/2007/owl/person#male> .
  ?person
  <http://www.ju.edu.jo/cs/thesis/2007/owl/person#hasSSN> ?ssn .
}
```

Fig. 10 The query which is used in the External annotation and the query which is used in the Internal annotation. These queries are built by using the SPARQL syntax.

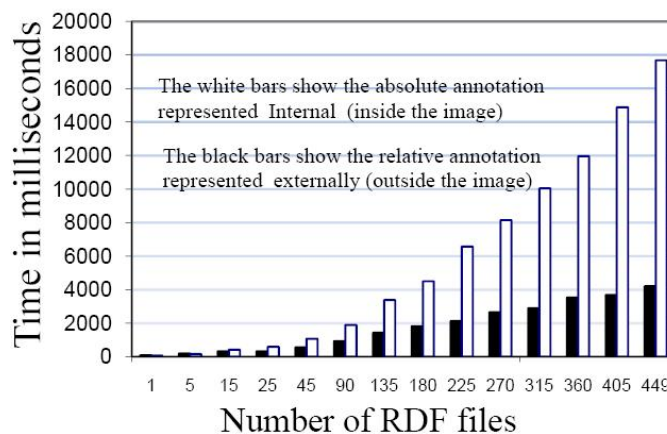


Fig. 11 Comparison between external annotation and the internal annotation based on the query execution time.

As we see from fig. 11, the time period that is needed to extract the result when we execute the query on the internal annotation is less than when we execute it on the external annotation for the number of RDF files is less than 5 files. That is because the complexity of Human ontology (Appendix B) where its instances are stored in the external annotation. Whilst the internal annotation that has instances from the Person ontology (Appendix A) considers simple comparing to Human ontology. The effectiveness of the complexity of the ontology is almost disappeared when the number of the RDF files exceeded 25 files. Where the effectiveness of the factor of the number of steps that is needed to extract the RDF files from the images before running the query on them appears. In the other side, we can run the query on the external annotation directly without needing further steps. Fig. 12 displays these steps for both the external and the internal annotations.

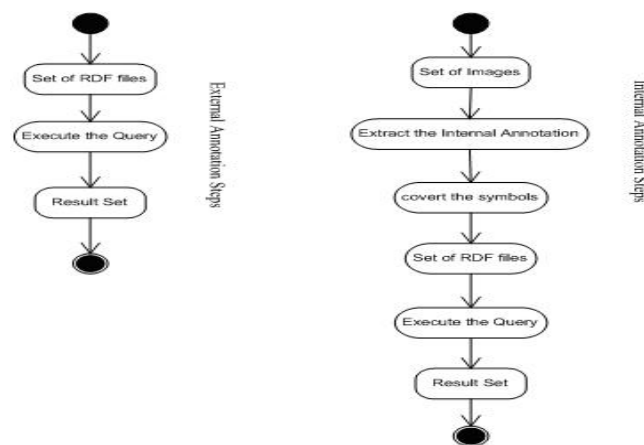


Fig. 12 The external and the internal annotations steps that are needed to extract the result by executing the query on them.

For this reason we run the query on the external annotation, which will give us a set of images. After that, we run the query that is related to the internal annotation on this set of images which represent a subset of the whole images. By doing this we improve the performance of the application.

D. Experiment 4

The goal of this experiment is to mitigate the difference of execution speed time between the external annotation and the internal annotation, when a query is executed on them. To fulfill this goal we cache the internal annotation of the whole images, by doing that we reduce the number of steps that are needed to execute a query on the internal annotation into three steps, fig. 13 displays these steps. For that, these internal annotations become external annotations. To differ between these two annotations we called the annotation which is originally internal an Absolute annotation, and the other annotation which is originally external a Relative annotation. For this experiment we build a program that perform a simple query on the relative annotation and register the time period that is needed to extract the result from these annotations, and make the same experiment on the absolute annotations too. Fig. 14 shows a comparison between the execution time for these statistical tables.

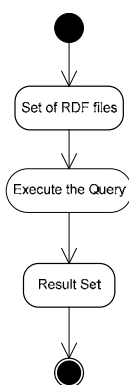


Fig 13 The absolute annotations steps that are needed to extract the result by executing the query on them.

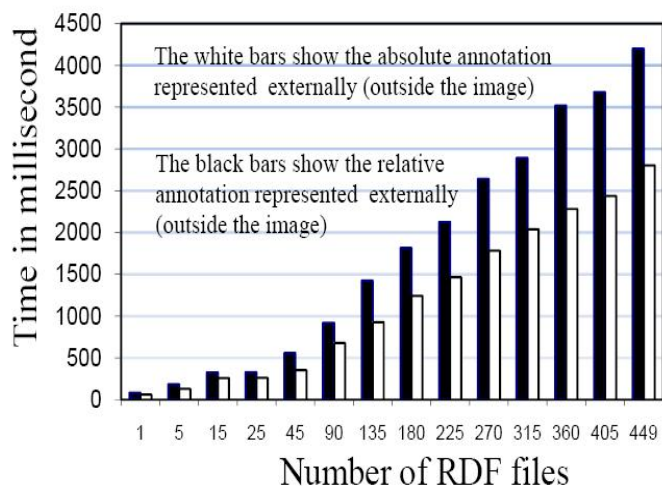


Fig. 14 Comparison between relative annotation and the absolute annotation based on the query execution time.

As we see from fig. 14, the period of time that is needed to extract the result when we execute the query on the absolute annotation is less than when we execute it on the relative annotation. That is because the complexity of Human

ontology where its instances are stored in the relative annotation. Whilst the absolute annotation that has instances from the Person ontology considers simple comparing to Human ontology.

Despite this result, we adopt on our proposed idea to run a query on the relative annotation before running it on the absolute annotation, because in general the number of attributes in the relative annotation is larger than the number of attributes in absolute annotation. Thus, when we run the query on the relative annotation the number of the retrieved RDF files will be less than the number of retrieved RDF files if we run the query first on the absolute annotation. For that, we reduce the set of images, which we need to check, to the number of RDF files that are retrieved when we ran a query on relative annotation which may equal to 1.

VIII. CONCLUSION

The recent years have witnessed a huge growth of digital image collections, which motivates the research of image retrieval. Several methods were developed to retrieve these images. Some of these methods use the low level features of images and others use the high level features of them. Due to the fact that most ordinary users are interested in retrieving their images according to the high level feature of image; one of the solutions to retrieve the images based on the high level features is to tag these images using the natural text language. The main problem with this approach is the ambiguity, and the inaccessibility of its semantic to machines. To enable information sharing and reusing it across application, computers need access to the semantic of the content. Semantic Web emerges to overcome this limitation; Tim Berner-Lee proposed it as an extension of the current Web, in which the information is given a well defined meaning. The meaning of the information on a web page is formalized using semantic metadata that is based on concepts defined in ontology.

In this work, we introduce a search engine for the Human face pictures, which aims at using the Semantic Web notion in the image retrieval by annotating these images with semantic metadata based on concepts defined in ontology. To do this, we divide our approach into two steps; the first one for the annotation and the second one for the retrieval. To be able to retrieve images, we should first annotate them. We divide the annotation step into two stages, dynamic stage and manual stage. The dynamic stage uses API to read the related data from the database and store it inside an image in the metadata part. The candidate data that should be stored inside the image is preferred to have the stability and to describe the logical and absolute facts about the image like the name of the person who appeared in this image. This candidate data is not dependent on the annotator's point of view. The purpose of storing this type of data inside the image is to give the image the portability characteristic, especially when the owner wants to share his images with his friends. The second stage is the Manual stage; this stage is done by the annotator who uses a

web application to annotate images. The resulted annotation, which is on the Semantic Web recommendations, is stored in a separate file from the image file. The purpose of this separation is to improve the performance of the retrieval step, and to keep the size of the images as small as possible, because this annotation is based on the annotator's point of view, thus the other applications may not be interested to use this type of annotations. One of the major problems with the annotation is the differences of interpretations among the annotators. To reduce the impact of this problem, we use coarse values for the attributes, where the annotator uses the leaf node from this hierarchy, whilst the retriever uses the parent of these leaves in the retrieval step.

The retrieval step is also divided into two stages; Query Based Retrieval and the Image Based Retrieval. The user who wants to find a required image, he will use the web application that we have built for this purpose. The user describes his wanted image using this web application. After completing the description, he presses on the submit button to submit his description to a component which converts these description into a query according to the Semantic Web recommendation. This query is first applied to the external annotations which were created by the annotators in the previous step. Each resource from the resulted set contains a reference to the related image; from these images we extract the internal annotation. The part that is related to the internal annotation from that query is applied with the extracted internal annotation. After that, we get a subset of images from that set. This subset is displayed for the retriever with brief information, which is extracted from the internal annotation, about each image in that set. When the retriever selects an image from the displayed set of images, this image will be a seed for the next stage which is the Image Based stage. This stage uses the external annotation of the seed image to build a query, this query is applied to all image to find the images that have the same annotation with that image.

We implemented this search engine using two different ontologies; one for the internal annotation and the other for the external annotation. Then, we build a complete web application using Java language and the Apache Tomcat as a web server for this system. After that, we run several experiments on this application and analyze the result, where we find that we can retrieve images in a shorter time and a higher accuracy.

APPENDIX

A. Person Ontology Concept Structure

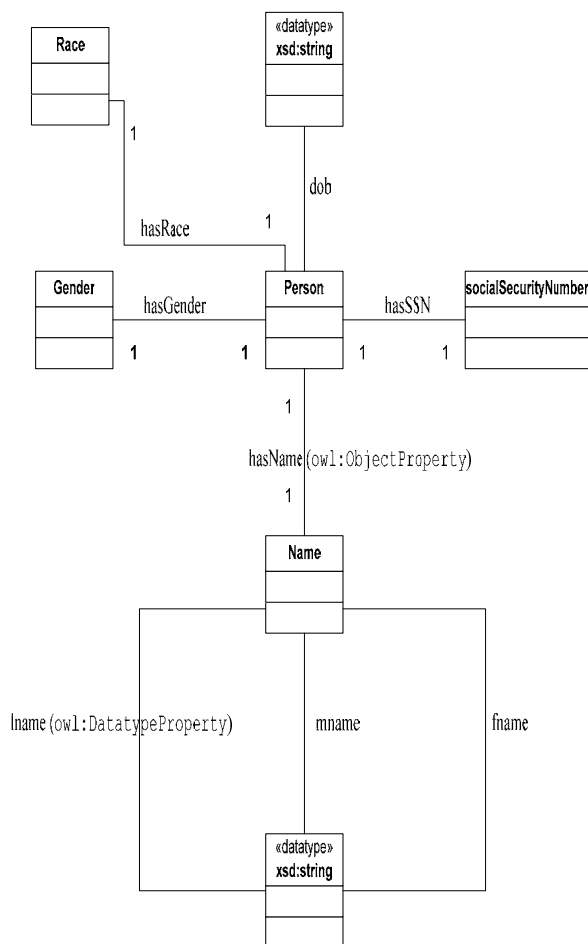


Fig. 15 Abstract person ontology concept structure.

B. Human Ontology Concept Structure

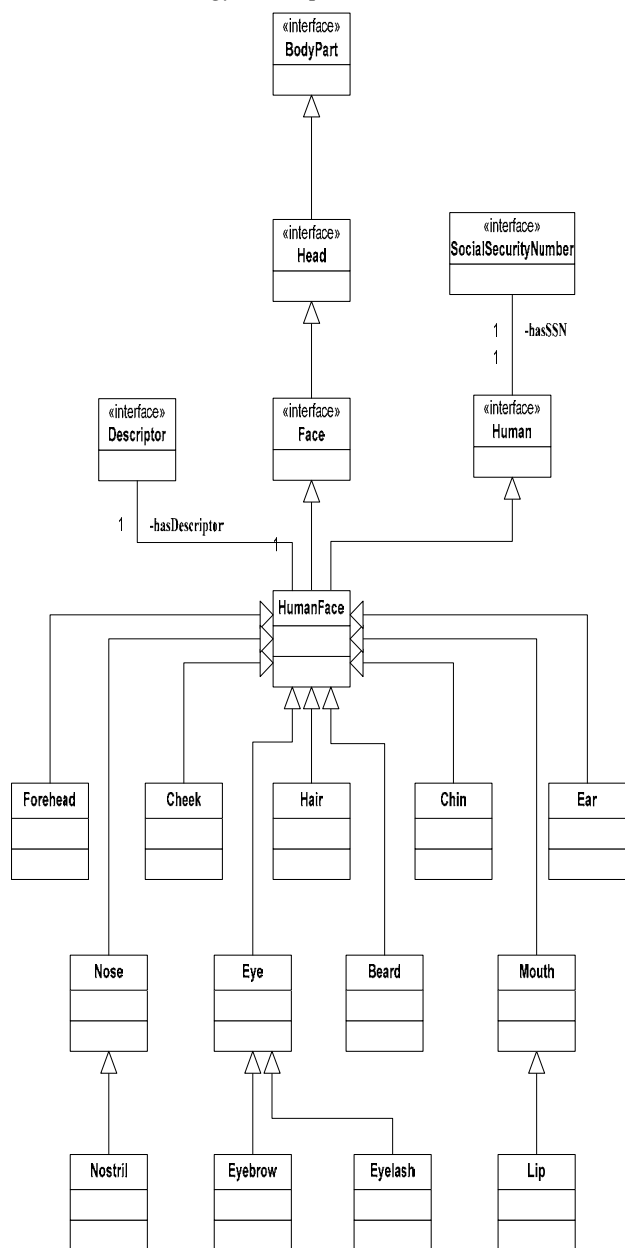


Fig. 16: Abstract human ontology concept structure.

REFERENCES

[1] M. L. Pawlak, "Image Analysis by Moments: Reconstruction and Computational Aspects", Oficyna Wydawnicza Politechniki WrocLawskiej WrocLaw 2006.

[2] A. Chavez-Aragon, and O. Strostenko, "Image Retrieval by Ontological Description of Shapes (IRONS)", in *Proc, 1st Canadian Conferences on Computer and Robot Vision, 2004*.

[3] I Cox, M. Miller, T. Papatthomas, and P. Yianilos, "The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical experiments", *IEEE Transactions and Image Processing, 2000*.

[4] J. Ossenbruggen, G. Stamou, and J. Pan, "Multimedia Annotations and the Semantic Web", *IEEE Multimedia, 2006*.

[5] J. Martínez, "MPEG-7 Overview (version 10) [Online]". Available: <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>.

[6] J. Bromans, and K. Hill, "MPEG-21 Overview v.5 [Online]". Available: <http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm>

[7] Dublin Core Community, Dublin Core Element Set, Version 1.1, <http://www.niso.org/international/SC4/n515.pdf>, visited October 2007.

[8] T. Berners-Lee, J. Handler, and O. Lassila, "A new form of web content that is meaningful to computers will unleash a revolution of new possibilities", *Scientific American, 2001*.

[9] E. Hyvonen, S. Saarela, A. Styrmm, and K. Viljanen, "Ontology-Based Image Retrieval [Online]". Available: <http://www2003.org/cdrom/papers/poster/p199/p199-hyvonen.html>

[10] A. Hliaoutakis, G. Varelas, E. Voutsakis, E. Petrakis, and E. Milio, "Information Retrieval by Semantic Similarity", *International Journal on Semantic Web & Information Systems, 2006*.

[11] Human Face Lab. Available: <http://www.lampstras.k12.pa.us/hschool/>.

[12] Java Image I/O API Guide. Retrieved August 2007/ Available: <http://www.java.sun.com>.

[13] Image IO Java Package, J2SDK documentation. (August 2007). Available: from <http://java.sun.com/j2se/1.4.2/index.jsp>.

[14] Jena - a Semantic web framework for java. Available: <http://jena.sourceforge.net>.

[15] Pere P. CVL face database. (October 2007). Available: <http://www.lrv.fri.unilj.si/facedb.html>.

[16] Jain V, and Mukherjee A. Indian face database. (October 2007). Available: <http://vis-www.cs.umass.edu/~vidit/IndianFaceDatabase/>.

[17] A. Kulkarni, "Content-based image retrieval using associative memories", in *Proc, 6th WSEAS Conf. Telecommunication and Informatics, Texas, March 2007*.

[18] P. Kidambi, S. Narayanan, "A Human Computer Integrated approach for Content Image Retrieval", in *Proc, 12th WSEAS Conf. Computers, Greece, July 2008*.

[19] T. Yoshida, "Representing implicit term relationship for Information Retrieval", in *Proc, 12th WSEAS Conf. Computers, Greece, July 2008*.