DESCRIBING AND CLASSIFYING MULTIMEDIA USING THE DESCRIPTION LOGIC GRAIL

C. A. Goble*, C. Haul+ and S. Bechhofer*

*Department of Computer Science, University of Manchester,
Oxford Road, Manchester, M13 9PL, UK
carole@cs.man.ac.uk, tel: +44 161 275 6195, fax: +44 161 275 6932

+Technische Universitat Braunschweig, Matrikelnummer 2333269,
Mauernstr 29, 381000 Braunschweig, Germany, c.haul@tu-bs.de

ABSTRACT

Many applications would benefit if media objects such as images could be selected and classified (or clustered) such that “conceptually similar” images are grouped together by content. This requires that image content be described by some coherent semantic domain model rather than relying on the use of keywords as in most commercial image database systems. However, a description of image contents cannot be predefined by prescribing what should be in the images but must incrementally evolve to link image instances with descriptions of what is actually there. Flexibility is required as the same image may be reused from many different application perspectives, and classified and reclassified by many different, unpredictable, and possibly contradictory interpretations of the same contents. We present preliminary work on the incremental and flexible description of image and video semantic content by the use of a description logic (DL), GRAIL, developed at the University of Manchester. GRAIL progressively bridges the gap between the uninterpreted raw image and the application’s semantic domain of ‘world’ objects by supporting the incremental specification of a schema, the automatic classification of descriptions (and hence images), the notion of “conceptual similarity” for imprecise queries, multiple granularity of views and reuse. We will then present a model for a video database system based on this approach. A primary aim is to determine if GRAIL in particular, and DLs in general, are suitable for such an application.

Keywords: semantic content based retrieval, annotation, images and video, description logics, GRAIL

1. INTRODUCTION AND BACKGROUND

If we are to retrieve and manipulate multimedia data effectively it is necessary to store not only the raw data itself, but to also hold and manipulate data about the raw data, which we can categorise into: *registration data*—media type related information such as size in pixels and compression techniques, and information about documents such as date of creation, and *metadata*—semantic and syntactic information about or extracted from documents. Registration data usually consists of simple data types that can be handled by classical database paradigms such as the relational model. The more interesting kind of data in this context is metadata, representing information about semantic content and structure (syntax) of images. Metadata is required to support content based image retrieval without reverting to the constant re-interpretation of sources: the number of documents is frequently large, the data is difficult and time-consuming to analyse for every query. Images are required to be *described* in some way; however rich the description there will always be attendant loss of information but we should seek to describe in the richest way possible. For example, the description of images, and hence their subsequent indexing and retrieval, commonly falls into two categories: image-based encoding techniques or text-based keywords. Content based retrieval (CBR) covers two broad kinds of content: syntactic and semantic.

1.1. Syntactic content

Syntactic content is typically structural information, and can be frequently acquired automatically by interpretation techniques, for example, by parsing video into individual shots[15,19] or automaton-based image examination[16]. Other examples are colour histograms, shape classifications, as well as spatial relationships between media features that are media specific and not related to real world objects. Queries are

based on: calculated features (e.g. colour distribution histograms, texture signatures); user defined functions (e.g. detecting colour distribution changes in a series of video images); non-textual descriptions (e.g. palettes of hues, retrieval by sketch, Query By Example) and structural spatial / temporal relations (e.g. a yellow object in the upper left corner). Such systems frequently support similarity based retrieval. Grosky[8] calls these "content based non-information bearing" as the systems concentrate on image-based representations where the images are not assigned any sense of *meaning* or semantics[5].

1.2. Semantic annotations

Semantic data covers the meaning of media documents or their objects, linking them to the collection of domain concepts—bridging the gap between 'world' concepts and the media itself. Systems that support the semantic annotation of, for example, images or image features can be termed *content based information-bearing*[8]. Annotations are commonly expressed as keywords associated with the image or image sets, organised into indexes and frequently using SQL as the query language. Keyword annotations do not replace image-based descriptions—describing texture or shape with words is hard or impossible—but complement them. However, as many commentators have pointed out5, if keywords are not pre-assigned to an image it will not be retrieved and there is no common vocabulary. Jain et al[11] state that "a keyword is worth a thousand images" and techniques for "power annotation" will be essential in the future. What is required is a way of annotating or describing images that is more than simple keyword association. A database of news information may have an image-independent domain database with an intension describing concepts such as politicians, events, and places, and a commensurate extension of instances of these concepts such as John Major, Serbia Peace Signing and the Paris. Images and their contents must be linked with such concepts if we are to describe and retrieve an image containing "John Major at a Serbian Peace Signing". Keyword systems for image databases have typically been viewed in isolation from any domain model, such that little structure is given to the allocation of keywords—for example some keyword are synonyms or kinds of others. A collection of ceramic artefacts may have keywords such as "vase" and "pot" where a vase is a kind of pot, so any requests for images of pots should include all vases. This kind of structure is frequently missing in cataloguing systems. Thus keyword annotations are often assigned in isolation and fail to form a structured coherent model.

1.2.1. Dynamic classification and reuse

An alternative approach is to pre-define the keyword annotations. This can lead to coherency but prescribes what *can or should be* in the images instead of *what is really there*. Prescribing keywords makes the annotations static, non-extensible and the images nonreuseable for other interpretations of content or other applications yet to come. Flexible extensible descriptions are required as the same image may be reused from many different perspectives, and dynamically classified by many different, unpredictable, and possibly contradictory interpretations of the same contents. An image of John Major in front of an aircraft may be classified as a Prime Minister, a man, a person, a British Citizen, a plane, a man and a vehicle, a British right-wing politician, an aircraft etc., depending on its use. Should we request an image of a plane, we would expect this image to be retrieved despite its primary focus on the person. Image instances are classified based on their content, requiring that image content should be described by some coherent semantic domain model rather than relying on the use of keywords. Smoliar and Zhang[19] for example, propose the modelling of relationships between real world entities and the contents of the documents. A logical consequence seems to be to use some sort of expressive knowledge base and expressive knowledge representation formalism for this purpose. Pre-definitions don't always manifest themselves as keywords but also occur as predefined types, particularly in image databases that are very application domain-specific, for example the early GIS systems which were generally not type-extensible, or only so with cumbersome reorganisation. A flexible, extensible and dynamic type system for our media instances and application domain is highly desirable.

1.2.2. Incremental elaboration, reclassification and incompleteness

Zdonik[23] proposes that image databases fall into the category of incremental 'bottom up' databases, where a description of the images cannot be predefined to fit with a

prescriptive database schema as in conventional databases, but must incrementally evolve to link image instances with a schema, or even evolve the schema from describing the image instances. When image data is captured it has little or no form though a good deal of substructure. Applications determine the appropriate substructure and additional substructure generates more schema. The schema could exist before connections between instances are made or not. For example, the concept of a UK Prime Minister could exist before images with instances of a prime minister are connected to these concepts. Instances can begin with no annotations linking them to the schema and be incrementally elaborated as more of the content is revealed or is required to be retrieved. So we might have an image of a man that we describe as such, and later elaborate on that description to include an aircraft, and still later further elaborate upon it to name the man as John Major. Hence our instances will always have incomplete or varying completeness of description, and as their descriptions are extended so they are reclassified, (e.g. as an image portraying a vehicle *and* a politician). Images collect annotations[11], so this kind of incremental support is essential.

1.3. Similarity retrieval based on annotations

How we have described the images determines their possible retrieval and indexing strategies. Imprecise or incomplete descriptions of image content will naturally mean imprecise and inexact matching of queries. Query by example, “retrieve all images whose content is similar to this one”, is common amongst image-based content retrieval systems[5]. Semantically, we may want to query on relations of entities in the real world (retrieve the names of all UK politicians) or exact matches (retrieve all videos of John Major and no substitutions will do). More interestingly, many applications would benefit if images could be selected because their semantic contents are “conceptually similar to” another image, for example, “retrieve videos of John Major” but actually the user is really interested in all kinds of politicians or if there are no videos of John Major, a video of a UK politician or an image of John Major may be a sufficient alternative. Likewise, should we ask for an image of a man, we would expect the image of John Major to be retrieved as he *isa* man. Such applications are particularly applicable to news or sports archives where there has been active research in classifying video sequences. We should also be able to request images using varying degrees of genericity in our description. Although we have concentrated in this motivation section on semantic annotations, much the same could also be said of syntactic annotations. We should aim to support:

- image reuse;
- unpredictability of use;
- automatic classification (or clustering) of images;
- incremental elaboration and reclassification of images and domain schema, and
- imprecise retrieval based on related concepts.

In summary, keyword systems tend to provide us with coherent annotations that are inflexible and difficult to extend and flexible annotations without structure or coherency. We require an extensible and coherent structure. We take the view that as keywords are language terms, we should look at coherent methods of representing terminologies, notably terminology logics, now usually known as Description Logics (DLs). In this paper we present preliminary work on the incremental and flexible description of image semantic content by the use of a description logic, GRAIL, developed at the University of Manchester. GRAIL progressively bridges the gap between the uninterpreted raw image and the application’s semantic domain of ‘world’ objects by supporting the incremental specification of a schema, the automatic classification of descriptions (and hence images or videos), the notion of “conceptual similarity” for imprecise queries, multiple granularity of views and reuse. A primary aim is to determine if GRAIL in particular, and DLs in general, are suitable for such an application.

The rest of this paper is organised as follows. Section 2 gives a brief introduction to description logics and GRAIL in particular. Section 3 presents the modelling and querying issues when applying GRAIL to a video database case study. Section 4 discusses our experiences and we conclude in section 5 with related and future work.

2. DESCRIPTION LOGICS AND GRAIL

Description logics, otherwise known as terminology logics, are descendants of the KL-ONE language. They have been studied extensively in Artificial Intelligence, especially in Natural Language Processing. There are a number of well known prototype DLs, including CANDIDE, BACK, CLASSIC, and LOOM, that all share common characteristics. We only briefly introduce DLs here—for an overview see[4].

All DLs define complex entities in terms of composite descriptions made up of a limited set of elementary concepts and assembled according to explicit rules. DLs can be viewed as languages obtained by term composition using recursive *term constructors*, where some terms are *concepts* (denoting a collection of individuals or instances) and *roles* (relationships between, or attributes of, concepts or individuals). Instances of concepts are called individuals and cannot themselves have instances. DLs distinguish between *primitive* concepts and *defined* concepts. A primitive concept has no characterising attributes and memberships are asserted for every individual object. As an example say 'Organisation' is a primitive concept. A defined concept is characterised by a set of attributes or role-fillers whose presence makes an object belong to this concept. A corresponding defined concept could be 'PoliticalOrganisation' which is an Organisation with a PoliticalStatus as a role-filler for the attribute hasState[†] :

```
Topic newSub [Person Thing Organisation State].        ........primitive concepts;
State newSub PoliticalStatus.
Attribute newAttribute hasState isStateOf manyOne.     ........attribute and its
                                                       inverse definition;
Organisation hasState†† State.                         ........sanctioning attribute
                                                       between concepts;
(Organisation which hasState PoliticalStatus)          ........defined concept, with
name PoliticalOrganisation.                            expansion name;
PoliticalOrganisation newSub PoliticalParty.
```

The concepts define a subsumption lattice (or semi-lattice): primitive concepts are placed there by the system designer whereas defined concepts are placed automatically by a classifier based on their compositional structure, or definition, and hence are a form of implied and more specific subtype of their base supertype, inheriting the properties of their supertype. Hence PoliticalOrganisation is a subtype of Organisation inheriting its properties. A further example relates the composite description PoliticalParty to a primitive concept Person to form new defined concept of "Person who works for a PoliticalParty" (i.e. a politician). This concept is subsumed by, and hence inherits properties of, Person.

```
Person worksFor Organisation.
(Person which worksFor PoliticalParty)   ........sanctioned because PoliticalParty
name Politican.                          is_a_kind_of Organisation
```

If the worksFor relationship is bi-directional then the inverse of this description for Politician (i.e. PoliticalParty which employs Person) is subsumed by Organisation.

From this paper's point of view the two most important characteristics are:

- elaborate (recursively) compositional descriptions of individuals (and concepts);
- automatic classification where the meaning of concepts are determined by their structure, concepts or individuals are classified by their descriptions and relationships exist between concepts by virtue of their definition. Child descriptions are more specific, and more elaborate, versions of their parents.

The power of DLs relies on the exploitation of the automatic subsumption of compositional descriptions with respect to one another. If descriptive models make use of highly compositional defined concepts wherever possible, we can navigate the lattice

---

[†] In the style of GRAIL

[††] In GRAIL attributes are bi-directional and have a two level sanctioning semantics discussed in 2.1, and omitted here for clarity.

to give us: for one description its direct subsumers and its direct subsumees, and for two descriptions, their greatest common subsumee (meet) and their least common subsumer (join).

Although DLs have been used primarily as a way of expressing terminologies for natural language processing, they are increasingly being used for data modelling[4]. DLs are much more expressive than semantic data models such as the EER. The subsumption ordering on descriptions corresponds to type refinement, and hence subsumption algorithms can be used for type checking. If a description fails to be classifiable with respect to the model's concepts and relationships it is possible to test the coherency of a concept, supporting the verification of a schema's consistency.

2.1. GRAIL

GRAIL has its origins in the GALEN project, which aims to build a terminology server for the medical domain[7]. The Terminology Server presents clinical concepts to applications which may use the concepts to drive content-sensitive user interfaces, to mediate between medical classification systems or as types for medical records. GRAIL has been specifically devised for medical terminologies, which has influenced its range of term constructors. In contrast to some representations certain constructs are excluded (notably existential quantification) and others restricted, in particular universal quantification, disjunction and negation. Individuals were present in an early version of GRAIL and they have recently been re-instated as a consequence of GRAIL's application to image and hypermedia document description.

The function of GRAIL is to represent statements that allow the expression and validation of all and only semantically correct descriptions. GRAIL describes a subsumption network consisting of simple, elementary entities, bi-directional binary relationships linking concepts and 'particularizations' which are new composite entities implied by the descriptive relationship. Particularizations are placed in the subsumption hierarchy by a classifier based on their structure. GRAIL can be viewed primarily as a representation model for supporting the creation of conceptual schemas or ontologies as collections of semantic constraints. GRAIL differs from its KL-ONE relatives in that it has:

- a *sanctioning* constraint mechanism such that only semantically valid concepts can be combined into descriptions and act as, currently, a two-level type system: grammatical and sensible (in the medical scenario 'fractured livers' grammatically correct but semantically nonsensical). This allows us to *generate* concepts implied by the model and guarantee their semantic correctness. This is instead of the role restriction mechanisms found in other DLs;
- *essential assertions*, called *necessary* statements, at the concepts level: e.g.: 'Metastases are always malignant' so a benin metastases is a nonsensical concept. Hence the GALEN concepts model is not just a model of pure terminology. These essential assertions take part in classification;
- a *canonisation* mechanism for ensuring that all equivalent, tautologous and redundant concepts are identified and reduced to a unique canonical form (the 'left hand' and the 'hand attached to the left arm' are the same object). This is instead of the SAME-AS designation in other DLs;
- the *co-ordination of partitive hierarchies*, transitive relationships and subsumption (classification) hierarchies (e.g. 'the shaft of the femur' is a division of the femur, not a kind of femur', but a 'fracture of the shaft of the femur' is a 'fracture of the femur');
- *limited expressions of cardinality*: one:one, many:many and one:many are supported but 'at-least' and 'at-most' with numeric cardinalities are not. Cardinality is defined as part of the attribute rather than in association with a defined concept or individual.

3. DESCRIBING MULTIMEDIA DOCUMENTS

We based our experimental investigations on a sample application designed to serve a similar task to the one described in[10] for the VideoSTAR system. The desired task is the management of a large video archive with an emphasis on news videos such as that found in a broadcasting company or news broker. The contents of such an archive are

frequently reused for assembling new reports. Hence it is necessary to provide comprehensive query facilities and support for reuse. VideoSTAR models a video from several different granularity levels: shots, scenes, sequence, and compound units which again can contain an arbitrary number of other compound units. Each of these structural components can have an annotation of persons, places, keywords and objects. The annotations are classical indexes on these classes. Semantic queries are handled by finding sets of shots or scenes connected to one of the desired keywords, and constructing intersections until only shots or scenes are left that have all the specified annotations.

We aim to build an index based on textual descriptions of the stored documents, concentrating on registration data and metadata, chiefly semantic annotations. The assumption is that the descriptions are made either at the time of capturing a document, incrementally when documents are authored or incrementally when browsing through them. We must describe two type spaces: that of the media themselves and that of the real world concepts they cover, both in an extensible way.

3.1. Describing the semantic contents

In Hjelsvold's approach[10], categories of interest are created such as persons, locations and keywords. However, we get a richer model capable of sustaining more sophisticated queries if we model the categories of a video in GRAIL. This requires that we (a) describe the world concepts type-space and progressively link the media instances with these concepts and (b) describe the media instances,which may mean extending the world concepts type-space. Let us describe an image of a male politician. We have simplified the sections of the GRAIL model, by omitting definitions of attributes, domain values etc. when these aren't necessary to make the point. In our world model we need to describe the concepts people, politicians, news topics etc., in our media model we need to introduce the notion of a video and relate the two:

```
MediaObject newSub [Video Image Audio].
Attribute newAttribute covers isCoveredBy manyMany.
MediaObject covers Topic.
Video newIndividual V0001.                                  .....media instance assertion
V0001 really covers                                         .....describing a media object instance
(Person which hasSex male, worksFor PoliticalParty).            with a composite description
```

This definition classifies the video V0001 as being an instance of the concept (or class) Video, and described as covering a male politician. This means that the description attributed to V0001 is classified as

```
Video which covers (Person which hasSex male, worksFor PoliticalParty).
```

hence V0001 is multiply classified as one portraying a person, a male, politician, and a male politician, and is subsumed by concepts:

```
Video which covers (Person which hasSex male).
Video which covers (Person worksFor PoliticalParty).
Video which covers (Person worksFor PoliticalOrganisation).
Video which covers (Person worksFor Organisation).
Video which covers Person.
Video which covers Topic.
```

If we now describe the video as also having an aircraft in it:

```
Thing newSub Vehicle.
(Vehicle which travelsOn air) name Aircraft.
V0001 really covers Aircraft.
```

Then the video V0001 is reclassified as also being a:

```
Video which covers ((Person which hasSex male, worksFor PoliticalParty), Vehicle
which travelsOn air).
```

which is subsumed by all the concepts as before plus:

```
Video which covers (Vehicle which travelsOn air).
Video which covers Vehicle.
Video which covers Thing.
```

We have been able to ascribe to an instance of a video a description of a man without knowing who he is and we have made assertions about individuals and concepts in combination. If we develop our world model further and create the individual of JohnMajor as

(Person which hasSex male, worksFor PoliticalParty) newIndividual JohnMajor).

We can go on to describe V0001 as

V0001 really covers JohnMajor.

and although the individual V0001 has more information asserted about it, it doesn't change its conceptual classification.

3.2. Extensibility and evolution: extending the type system

We have to cater for a complex domain. As it is difficult to predefine all eventualities and model them in advance there should be a possibility to evolve the schema and allow the creation of new media types, new methods to combine documents, new attributes for both media types and real world entities, additional real world concepts etc. Perhaps, when the video archive was built it was not considered important to annotate clips with keywords that described the political persuasion of the people in those clips. It could be that this becomes important later and the category of conservative political parties is created. GRAIL supports this view of an evolving type system or schema and consequently there are no differences between introducing extensions and the initial creation of the schema. If we go on to assert that:

PoliticalParty newIndividual [ConservativeParty LabourParty].
PoliticalColour newIndividual [conservative liberal socialist].
ConservativeParty really hasState conservative.

classifies the ConservativeParty as a

PoliticalParty which hasState conservative. ....as well as....
PoliticalOrganisation which hasState conservative. ....and....
Organisation which hasState conservative.

if

JohnMajor really worksFor ConservativeParty.

then John Major can be classified as the type

Person which worksFor (PoliticalParty which (hasState conservative)).

hence automatically reclassifying V0001 as a:

Video which covers (Person which hasSex male, worksFor (PoliticalParty which (hasState conservative))).

Instances can begin without any annotations linking them to the schema and be incrementally elaborated as more of the content is described. We also continue to elaborate dynamically our world model, though care must be taken so that past descriptions remain classifiable. As media objects incrementally collect annotations, which may include new concepts hitherto unknown, we can go some way to driving the schema creation process from the instance description process. As we saw before JohnMajor is classified as a series of concepts forming a type space capable of homing in on collections of instances and testing instance descriptions for coherency w.r.t the model. Thus the concept lattice can be viewed also as a series of types for the individual that becomes more general the further up the generalisation hierarchy you go and an elaborate semantic index on the individual's description. To be useful it is only required to be a coarse index capable of indicating which videos would be good starting places to look: pruning the search space by restricting the type space.

GRAIL doesn't just support terminological concepts: it is also possible to make assertions about concepts that aren't definitional but are true, and for these to take part in the classification process. Such assertions are called *necessary statements*. For example, if the definition of a politician is (Person which worksFor PoliticalParty) name Politician and we can assert that Politician necessarily hasCharacteristic[†††] publicPerson. then John Major is now classified as a (Person which hasCharacteristic publicPerson). Hence video V0001 is now also classified as a video about a public person.

[†††] hasCharacteristic is an attribute that has already been defined as grammatical and sensible for some more general parent concepts of the two entities it links.

3.3. Imprecise and Precise Queries

In DLs the query language and the description language are unified, such that you describe the object you want to find (be it concept or instance) and classify it. If it classifies correctly then it is a coherent description w.r.t. the model, and we can use this classification and the subsumption lattice to explore potential answers to this and more general questions. We can present the following semantic queries:

ι *relations of entities in the real world*: e.g. retrieve instances of politicians.
(Person which worksFor PoliticalParty) allIndividuals. will return a collection of Person individuals described as being politicians.

ι *intensional enquiries and descriptive answers*: e.g. given the concept Person, what can be said about a Person. The generative aspect of GRAIL is such that all the attributes that can be attributed to Person are explored and all the possible combinations of new concepts can be created, for example:
(Person which worksFor PoliticalParty).
(Person which worksFor PoliticalParty, hasSex male). etc.

This has been extensively used in GALEN for generating concept driven contextual user interfaces, or for new users who wish to explore the concept space. Because we can associate arbitrary descriptions with individuals we can also return descriptive answers rather than extensional values. The answer to the question "who works for the Conservative Party" could be (Person which worksFor PoliticalParty which (hasState conservative)). This is useful when we have incomplete information or when it is appropriate to produce abstract answers by finding the least common subsumer of a set of individuals' descriptions.

ι *exact matches*: retrieve videos of John Major, and no substitutions will do:
(Video which covers John Major) allIndividuals. which returns a collection of Video individuals described as covering John Major, or the empty set.

ι *conceptual similarity*: retrieve videos of John Major but we are really interested in all kinds of politicians. DLs are useful for querying knowledge bases in circumstances where the user is not familiar with the contents or structure of the data, aren't sure what question to ask or use as an example of the kind of object they are looking for as a 'template' to start with. Because the descriptions can be classified into a subsumption hierarchy if a query description returns the empty set it is reasonable to consider generalising the query until a non-empty set is obtained. For example, we ask for a video of female conservative politicians but there are none. If we generalise the query we may have videos of conservative politicians or female socialist politicians or just politicians, which we could term as "conceptually similar" and which may be adequate for our purposes. So we can generalise queries first by finding the class an individual belongs to and then "climb the generalisation hierarchy" to find the super classes of that class, and that classes superclasses etc. This may also lead to discovering of "new" knowledge that is implicitly contained in the knowledge base and wasn't realised before. Hence we can cater for imprecise queries by a process of query generalisation, by relaxing the query attribute's value. The pattern of query, classify, test for instances, generalise, classify etc. is described in1. The lattice of subsuming descriptions provide us with a search space for such generalisations, and can effectively support iterative query refinement. Let us define:
(Person which worksFor PoliticalParty) newIndividual GlendaJackson.
GlendaJackson really worksFor LabourParty.
Video newIndividual V0002. V0002 really covers GlendaJackson.
John Major is classified as:
Person which (hasSex male, worksFor PoliticalParty).
So by relaxing the individual in the query Video which covers John Major by its conceptual description, we get
Video which covers (Person which (hasSex male, worksFor PoliticalParty)).
As we generalise the Person further, moving up the subsumption hierarchy, we get
Video which covers (Person which worksFor PoliticalParty) allIndividuals.
This query will result in a collection of Video individuals that includes V0001 and V0002. We can also relax the Video class to be the more general MediaObject, and hence request all images that describe John Major. V0001 actually describes more than just a

politician, so once this individual is selected its (more elaborate) description might well trigger further queries.

3.4. Syntactic content: composition, decomposition and the inheritance of annotations

Although syntactic modelling hasn't been a focus of the work so far, we have made some investigations with regard to the inheritance of semantic annotations between compositional media objects. The role of the data model is to allow any number of composition levels without any maximum number of iterations an incremental specification may have. For example, let us take a Video that is composed of a series of Clips.

MediaObject newSub [Video Clip].
Clip newIndividual C01. Clip newIndividual C02. Video newIndividual V01.
(Vehicle which travelsOn air) newIndividual Boeing747.
C01 really covers GlendaJackson. C02 really covers [JohnMajor Boeing747].
V01 really contains [C01 C02].

If we wish to recover all topics covered by C01 we pose the query

Topic which isCoveredBy C01.

which returns a collection of individuals with two members: JohnMajor and Boeing747. If we wish to recover all topics covered by V01 we pose the query

Topic which isCoveredBy V01.

which returns a collection of individuals with three members: GlendaJackson, JohnMajor & Boeing747. This happens because the covers attribute has been declared to be *refined along* contains, and contains has been declared to be *transitive*.

3.4.1. Transitivity

An essential characteristic of documents is the importance of part-whole relationships. GRAIL co-ordinates partitive hierarchies and subsumption hierarchies, and can endow partitive or containment relationships with special transitive properties. Subsumption is an obvious example of a transitive relation*. Other relations, particularly partitive ones, also behave in this way. If Frame is part of Shot, which is part of Sequence, then Frame is part of Sequence. This results in an extended classification search, and it supports a similar functionality to the isa relationship but without the attribute inheritance. It is useful for expressing recursive attributes:

Attribute newAttribute contains isContainedIn manyMany.
contains transitiveDown. isContainedIn transitiveDown.
MediaObject contains MediaObject.

3.4.2. Refinement

This is a similar property which describes how relations can interact with one another. For example, Topics are covered by a clip which can be part of another clip. Covering is a refined attribute across the part attribute, then topic T, covered by clip C1, which is a part of clip C2, is also covered by clip C2. Hence a query for topics covered by C2 returns also T. The constructed document inherits "bottom-up" descriptions attributed to its parts. The attributes are defined as:

Attribute newAttribute covers isCoveredBy manyMany.
isCoveredBy refinedAlong contains.
MediaObject covers Topic. (hence Topic isCoveredBy MediaObject)

GRAIL provides facilities for specifying which relations are transitive or refined along one another which the subsumption algorithms and tests take into account. The mechanisms to deal with this are bound up in the subsumption rules for criteria. Although C01 is not *a_kind_of* V01, and a Clip is not *a_kind_of* Video:

Topic which isCoveredBy (Clip which isContainIn Video). ....is a kind of ....
Topic which isCoveredBy Video.

V01 will be classified as the concept:

Video which (contains (Clip which covers Person),

---

* If a Minister is a kind of Politician, and a Politician is a kind of Person, then a Minister is a kind of Person.

contains (Clip which (covers Person, covers (Vehicle which travelsOn air)))

If media objects can be composed over arbitrary levels then the same goes for decomposition. While composition is relatively easy to define in respect of participating objects' attributes, decomposition is without deeper understanding of the media data not possible. Oomoto and Tanaka[17] suggest that *interval inclusion inheritance* defines the relation between the whole and its new parts, that not all attributes should be inherited and the ones that should, have to be specified in advance by the user. It is difficult to define which attributes of the whole should be inherited by the parts, if any. Strictly speaking this is not really an inheritance because of the double nature of this association: in terms of a complex data structure are parts more special than the whole (from a classification approach are parts more general because they are less specified[**]). Consequently, parts cannot inherit by being subconcepts, they have to be created on the same conceptual level as the data they are part of. As a result the application needs to support logical decomposition by offering certain attributes to be "inherited".

Hjelsvold[10] differentiates queries for their VideoSTAR system into five types of common tasks:

Browsing: GRAIL's browsing tool follows the conceptual lattice at an adjustable granularity level and with an adjustable view: the user can move between the subsumption hierarchy and compositional based view, for example.

Content Queries: In order to create a new video clip the user needs to find suitable sequences in old clips by specifying the contents.

Complex Content Queries**:** Hjelsvold considers finding a report on the European Union which has a component that covers the illegal gathering of eggs from protected birds. A simple system based on textual descriptions linked to individual documents relies on the fact that all relevant information is contained in the keywords annotation *illegal, egg gathering* and *protected bird*—annotations *egg gathering* and *stork* make it impossible to infer that it is a case of illegal egg gathering, a stork is a bird and a stork is a protected bird without a knowledge base that states which birds are protected and all egg gathering from protected birds is illegal. In our approach the query could be formulated as ((MediaObject which covers EggGathering, Bird which hasState protected) which isContainedIn (MediaObject which covers EU) allIndividuals. If illegal egg gathering is defined as an environmental crime then the query ((MediaObject which covers EnvironmentalCrime) which isContainedIn (MediaObject which covers EU)) allIndividuals. retrieves the clip without asking for every possible environmental crime or adding to the document's original annotations.

Clip List Generation: The GRAIL browsing tool can be used for inspecting particular composition branches to supply lists that detail how a video is composed from others.

Contents Report Generation: List the topics covered by a given clip, essential when reusing a piece as the author must be aware of the context the sequence is taken from. A contents report can be generated by the query (Topic which isCoveredBy clip) allIndividuals. resulting in a complete list of all topics a (complex) video clip is about by virtue of the refinement characteristics of isCoveredBy as described earlier. The GRAIL browser could be used on the result of this query, which would also show all other annotations or generate a clip list.

3.5. Describing Media Types

We need to describe the characteristics of video as against image, for example. We can use the composition and descriptive aspects of GRAIL to define the media type-space as well as the domain type-space. If we define one recursive and transitive concept called *MediaObject* media on every level could be treated equally and be available to an arbitrary level of composition and decomposition, with registration data and metadata attributed to every media object regardless of granularity. In GRAIL, using the sanctioning mechanism on attributes, we are able to define a two layered type-space,

[**] Is a video about 'John Major' and 'Helmut Kohl' more specific than a video about 'John Major' even if the latter is part of the former.

separating out those attributes that can generally be applied to a type and those that are semantically sensible to apply. For example,

MediaObject grammatically contains MediaObject.
MediaObject newSub [Video Clip Frame Image].
Video sensibly contains Clip.
Clip sensibly contains Clip.
Clip sensibly contains Frame.

means that any MediaObject can have a topic annotation, and in general a MediaObject can contain another, but really it is only sensible for a Clip to contain Frames rather than Images, and by recursion a Video can sensibly contain Frames. This doesn't, by the way, mean that a Video instance *has* to contain a Frame instance, it is just permissible. Sensible statements between concepts must be sanctioned by an appropriate grammatical statement, and really statements for individuals must be sanctioned by an appropriate sensible statement[***]. Any Topic annotations on an individual of type Frame will be inherited "bottom-up" to the appropriate Clip and Video individuals, as described above. Let us define some registration data:

FormatType newSub [VideoType ImageType]. VideoType newSub [VHS PAL].
MediaObject grammaticallyandSensibly hasCreationDate Date.
MediaObject grammatically hasFormat FormatType.
Video sensibly hasFormat VideoType.

This states that all MediaObjects can have creation dates and formats, only Videos can have video formats, like VHS, and a length, and an Image can't have a length. The individual:

Image newIndividual IM1111. IM1111 really hasFormat VHS.

will fail the type definitions as its concept Image which hasFormat VideoType. cannot be classified. Media types have their defining characteristics described. This makes it easier to specialise them into new types and to develop the type specification incrementally.

4. DISCUSSION

Our aim isn't to model exhaustively every aspect of metadata but to support the coherent and incremental development of a coarse index on the semantic annotations of media documents such that we can easily find documents that either match or are conceptually similar to, query descriptions. The application of description logics for describing semantic metadata annotations for multimedia documents, and forming a coherent and extensible knowledge base of domain and media types appear to have some promise, providing all basic functionality for handling registration data and querying on an exact-match basis as well as imprecise queries. Returning to our original wish list:

- *Automatic classification, imprecise retrieval based on related concepts and complex content queries.* Queries can be answered based on conceptual similarity. This drastically reduces the effort made for document research and eases the annotation process.

- *Incremental specification of schema, elaboration and reclassification of media instances and unpredictability of use*. Annotations to documents can be added at anytime which is especially important as no description could ever be complete and hence needs to be extended. This is achieved through an expressive compositional model

- *Reuse*. The prototype system does not impose restrictions for reusing documents in other documents which was one of the major design intentions.

- *Granularity of views.* The data can be viewed on different granularity levels and from different view points. This is particularly useful for the view of the contents space or when investigating compositions. E.g. one could switch from the standard isAKindOf inheritance structure to a contains view to see all parts and, after identifying a particular one, go to an isPartOf view that shows all complex documents this part is contained in.

- *Mapping between document contents and real world.* Not only can the data stored about the documents be utilised in queries but also the information stored in the

[***] as must necessary statements between concepts

annotations. A knowledge base is built up that improves the search for suitable documents immensely. The chosen approach provides the functionality necessary to create relationship links between documents and between documents and real world entities. Relationships can be introduced on instance-, concept-, or mixed-level.

We believe that DLs offer a principled and powerful way of expressing, indexing and retrieving annotations. However there are limitations in both GRAIL itself and the use of DLs in general. The major ones are outlined below.

4.1. Model and annotation acquisition

Retrieval by annotations will only be as good as the annotations themselves. Although our approach caters for the incremental development of models and the incremental elaboration of instances, this doesn't solve the problem that the development of a model and the annotation of media instances is a time consuming and difficult business. Annotations are acquired from user interaction at the time of media capture, when the instance is being authored and also at query time. Automatic annotation techniques are much harder to achieve; the identification of image features will not be precise and depends on the application, the image analysis and image interpretation algorithms, and the state of any models or domain knowledge known at input or retrieval time. In PICTION[20] the semantic information index is generated from the document's textual descriptions by natural language processing using a description logic, LOOM, to interpret newspaper captions of images and drive automatic image interpretation and attribution of concepts to image features. MMIS6 used a semantic net representation for its knowledge base and attempted to link automatically image features with world concepts. If we are to achieve "power annotation" we need to unify terminological annotation techniques such as those outlined in this paper with techniques to annotate automatically the instances.

4.2. Reclassification

We are using GRAIL as an index or query language into a database of individuals. In order to answer questions such as "which videos cover politicians" we simply classify the appropriate concept and then return the collection of individuals that are instances of that concept. In this way we are using GRAIL as a sophisticated index. The power of this is that once we have our individuals installed we have effectively pre-compiled the answers to any questions that we may wish to ask. If a new question is asked (i.e. we request a concept description) this is classified and that classification corresponds to a collection of individuals that forms the answer. One of the benefits of our approach is the automatic reclassification of instance descriptions. It is also one of the drawbacks. As an individual changes its description it is reclassified with respect to the concept type system we have developed and this can lead to the reclassification of other individuals which in certain circumstances can cause a ripple throughout the lattice. Although the terminology is dynamic—questions and answers are built on the fly—the actual storage and classification of individuals is not. As we change the instances description we must ensure that references to the entity are consistently maintained. The integrity issues (ensuring the coherency of the lattice) becomes an issue when individuals are reclassified. This suggests that this kind of approach is best suited to applications where unpredictable and complex queries are made but the addition of new descriptions to individuals is restricted ,or tightly controlled, or it is understood that the indexing is not always exhaustive or complete.

4.3. Anonymous individuals

A further problem is the relationship of concepts and individuals. If we assert that

```
Video newIndividual V0004. V0004 really covers (Person which hasSex Male).
```

The query

```
Topic which isCoveredBy V0004.
```

is, for consistency's sake, neither an individual nor a concept but describes a collection of individuals. In order to retrieve the conceptual description we have an "anonymous" individual that carries that description that appears in our query result. Implementation maintenance issues occur when we later assert an individual for the person and have to unify this with the (temporary) anonymous individual.

4.4. Issues with GRAIL

Composition should be realised by an aggregation with additional attributes that specify the ordering and manner of the composition, for example parameters for an over blending effect. GRAIL does not support attributes on attribute relationships: attributes have to be promoted to being an entity. this leads to rather opaque and clumsy models, and complicates the classification process. Extensions to GRAIL to extend the attribute hierarchy to include attributes and full cardinalities are under review. GRAIL currently views numbers as symbols and does not understand the notions of counting or ordering. This information can be represented in the model but takes no part in the classification, which is the only form of inference possible. A query for a complex video individual whose *first* clip is about politicians cannot be answered but one with *a* clip about politicians can. A query for a video covering four (different) topics cannot be answered as currently cardinalities on attributes are only expressed as one:many or many:many, making expressions such as “retrieve a frame with two people” impossible to formulate. This is not a restriction found in other description logics. Modelling in an expressive and flexible DL is difficult: the key aspect being the identification of what are the terms that define a concept and what terms do not, but are merely true of the concept. It is easy to make false assumptions, for example:

Attribute newAttribute covers manyMany.
(Person which hasSex male) name Man.
(Person which worksFor PoliticalParty) name Politician.
V0001 really covers [Man, Politician].

does not mean that V0001 covers a male politician, but that it covers a politician and a man. As GRAIL has no form of shared variables we cannot assert that the Man and the Politician are the same.

5. RELATED AND FUTURE WORK

A common approach is to code the semantic information in some type structure, for example by using object-oriented methods[12]. Although object oriented systems provide many suitable features for multimedia database systems, Oomoto and Tanaka[17] in particular make the criticism that OODB type systems are generally static and do not support schema evolution well. They propose a descriptive schema that is evolutionary but within the framework of a conventional OO approach that doesn’t support automatic classification. Lahlou[13] shares many of our aims and uses a Semantic Data Model to describe images; however his model doesn’t appear to support automatic class classification. MORE[22] supports multiple views on the same instances by using domain knowledge to enhance the media instances’ OO type system with pseudo objects that are media-specific and derivable from the instances, including content analysis functions. The inferencing is not terminologically based and the type hierarchy, including the pseudo attributes, needs to be explicitly asserted. Our approach would extend MORE with a more sophisticated concept model. Frame-based systems[19], of which DLs are a principled form, are more flexible. Many authors[6,20] have used some form of knowledge base, usually based on semantic nets or frames, to describe images, drive image interpretation systems or to automatically label features with a semantic description, but without directly exploiting the imprecise querying and automatic classification possible through the use of a DL or using the knowledge descriptions directly as an instance annotation mechanism. DLs have been used in the field of Information Retrieval to describe and classify documents[14].

From these early experiences, DLs appear to be promising with regard to describing, annotating and retrieving semantically media documents. We plan to extend GRAIL to cope with some of the deficiencies described in section 4 (e.g. first class support for numbers) and experiment with other DLs, in particular we shall investigate the support that DLs can offer structural annotations such as spatial and temporal relationships.

We have examined DLs as a way of developing evolutionary semantic schemas for instances and describing domain knowledge. In a fully fledged multimedia database we should also support calculated features, user defined functions, automatic feature extraction and more conventional content-based similarity retrieval through the use of non-textual descriptions such as histograms and signatures. We plan an image workbench prototype in the field of medical images using an object oriented database for the document space (supporting large data objects, concurrency, reliability and media type specification),

GRAIL for the clinical terminology contents space and conventional CBR and feature extraction to support automatic image annotation along the lines of PICTION[20].

The use of terminologies to describe images is not new and predates computers. The principle of descriptive analysis of fine art was expounded by the Prague School in the 1920s[21] in the creation and application of a structured semiotic terminology used to describe the content of works of art. This terminology could be used to describe and automatically classify works of art by their content, and moreover identify patterns of change with reference to the social and environmental contexts of the artists and identifying common influences. Such work using picture description languages is highly active in museums[3]. Manipulating descriptions of works of art that is scalable and flexible requires the use of a terminology such as that provided by the Prague School coupled with a knowledge based computerised terminology able to support the automatic classification of works of art by description of aesthetic function and communication function, and the transient relationships between such terms. PaVE [18] uses a simple and static terminology for art; what we propose is complex and dynamic and we will be using subject specialists to build and determine the terminology. On a practical level this would provide a source for graphic designers, organisers of themed exhibitions, artists seeking source material with common image content and the general public with interests in kinds of images rather than specific artists or genre.

ACKNOWLEDGEMENTS

We would like to acknowledge the work of the whole of the Medical Informatics Group, but in particular Dr. Pam Pole and Dr. Jeremy Rogers for their helpful discussions and support, and Ian Cottam for his useful comments on this paper.

REFERENCES

1 Anwar TW, Beck H and Navathe S, “Knowledge mining by imprecise querying: a classification based approach” *Proc 8th Conf. on Data Engineering*, Tempe, Arizona, USA, Feb 1992 pp: 622-630.
2 Bechhofer S, and Solomon D. 1994. A Tutorial Introduction to the GRAIL Kernel. *GALEN Documentation*, vol. C. University of Manchester, UK
3 Besser H, Visual access to visual images: The UC Berkeley image database project, *Library Trends* vol 38, no 4, 1990, pp. 787-798
4 Borgida A, Description Logics in Data Management, *IEEE Trans Knowledge and Data Engineering* 7(5) pp: 671-782, 1995
5 Flickner M, Sawhney H et al Query by Image and Video Contents: The QBIC System *IEEE Computer* 28(9), pp: 23-32, Sept 1995
6 Goble CA, O’Docherty MH, Crowther PJ, Ireton MA, Oakley J, Xydeas CS, The Manchester Multimedia Information System *Advances in Database Technology EDBT’92, Third International Conference on Extending Database Technology*, Vienna March 1992, Springer-Verlag pp 39-55
7 Goble CA, Bechhofer S, Solomon WD, Rector AL, Nowlan WA and Glowinski AJ, Conceptual, Semantic And Information Models for Medicine, *Proceedings of 4th European-Japanese Seminar on Information Modelling and Knowledge Bases.* 31st May-3rd June 1994, Sweden, pp. 257-286. IOS publishing 1995.
8 Grosky W.I. Multimedia Information Systems. *IEEE Multimedia,* 1(1) pp. 12--24, 1994
9 Gupta A., Weymouth T., and Jain R. (30th Sept -- 3rd Oct). Semantic Queries In Image Databases. Knuth, E., and Wegener, L.M. (eds), Proceedings of the IFIP TC 2/WG 2.6 Working Second Conference on Visual Database Systems, pp:201--215, 1991
10 Hjelsvold R and Midtstraum R. Modelling and Querying Video Data. Proceedings of the 20th VLDB Conference, Chile, pp: 686--694 1994
11 Jain R, Pentland AP and Petkovic D. Workshop Report: NSF-ARPA Workshop on Visual Information Systems. available from UC at San Diego, USA, 1995
12 Klas W, Neuhold EJ and Schrefl M. Using an object-oriented approach to multimedia data. Computer Communications, 13(4), 204--216, 1990.
13 Lahlou Y, Modelling complex objects in content-based image retrieval, *Proc Storage and Retrieval for Image and Video Databases III*, SPIE Vol 2420, San Jose, CA, USA, pp: 104-115, 1995.
14 Meghini C, Sebastiani F, Straccia U and Thanos C A model of information retrieval based on a terminology logic *Proc ACM SIGIR93*, Pittsburg, USA, pp: 298-307, 1993

15 Nagasaka A., and Tanaka Y. Automatic Video Indexing and Full-Video Search for Object Appearances in (Knuth, E., and Wegener, L.M. (eds)), *Proceedings of the IFIP TC 2/WG 2.6 Working Second Conference on Visual Database Systems* pp:113--127, 1991.
16 Oommen, BJ and Fothergill C. Fast Learning Automaton-Based Image Examination and Retrieval. *The Computer Journal*, 36(6), 542--553, 1993.
17 Oomoto E., and Tanaka K. OVID: Design and Implementation of a Video-Object Database System. *IEEE Transactions on Knowledge and Data Engineering*, 5(4), 629--643, 1993.
18 Rostek L and Mohr W An Editor's Workbench for an Art History Reference Work *Proc ECHT'94*, Edinburgh, UK, Sept 1994, pp: 233-238
19 Smoliar SW and Zhang H. Content-Based Video Indexing and Retrieval. *IEEE Multimedia* 1(2), pp: 62--72, 1994.
20 Srihari RK, Automatic Indexing and Content-Based Retrieval of Captioned Images in *IEEE Computer* 28(9), pp: 49-56, Sept 1995
21 Titunik (ed) *Semiotics of Art*, MIT Press, 1973.
22 Yoshitaka A, Kishida S, Hirakawa M and Ichikawa T Knowledge Assisted Content-Based Retrieval for Multimedia Databases pp: 12-21, *IEEE Multimedia* 1(4) 1994.
23 Zdonik SB, Incremental Database Systems: Databases from the Ground Up *Proc ACM SIGMOD93*, pp: 408-412, 1993.