Annotating Scientific Images: A Concept-based Approach

Michael Gertz[1] Kai-Uwe Sattler[1,5] Fredric Gorin[3,4] Michael Hogarth[2] Jim Stone[4]

[1]Department of Computer Science, University of California, Davis
{gertz|sattler}@cs.ucdavis.edu

[2]Department of Pathology and Internal Medicine, University of California, Davis
mahogarth@ucdavis.edu

[3]Department of Neurology, [4]Center for Neuroscience, University of California, Davis
{fagorin|jmstone}@ucdavis.edu

[5]Permanent Address: Department of Computer Science, University of Magdeburg, Germany
kus@iti.cs.uni-magdeburg.de

Abstract

Data annotations are an important kind of metadata that occur in the form of externally assigned descriptions of particular features in Web accessible documents. Such metadata are eventually used in data retrieval tasks on heterogeneous, possible distributed Web-accessible documents.

In this paper, we present the model and realization of an annotation framework that scientists can employ to semantically enrich differerent types of documents, primarily scientific images made availabe through an image respository. Although we employ ontology like structures, called concepts, for metadata schemes used in annotations, our primary focus is on how concepts are actually used to annotate images and regions of interest, respectively, that exhibit features of interest to a researcher. It turns out that the combined consideration of domain specific concepts and annotated regions in images provides interesting means to analyze the usage of metadata regarding certain correctness and plausibility criteria. We detail our annotation management framework in the context of the Human Brain Project in which Neuroscientists record their observations on specific brain structures, and share and exchange information through concept-based annotations associated with images.

1. Introduction

Since the establishment of the Human Brain Project by the US National Institute of Mental Health of the National Institute of Health in 1993, there have been tremendous advancements in brain and behavioral research [12, 16, 3]. A major factor contributing to these advancements are recent developments in imaging and visualization techniques and tools (e.g., [20, 1]). Through these, neuroscientists are now able to investigate experiments and neurobiological phenomena at an unprecedented level of detail and precision. Another contributing factor to these advancements is the current image repository technology which allows researchers to manage and query images generated at different sites in a logically centralized fashion (see, e.g., [25] for an overview). Sophisticated data retrieval methods atop such image repositories, however, are still in their infancy. Pattern recognition and feature extraction methods that operate on diverse types of images and extract certain content descriptive, text-based metadata from such images are only of limited help. There are several reasons for this. First, many features are hard to describe and are often only discovered "manually" by researchers who investigate and interpret a given image in a specific research context. This is in particular the case where the classification of features in an image, e.g., neurons, or nuclei, is based on functional or biochemical properties of these features which are not explicit in the image. Second, many features are not yet known but are discovered while an image is investigated and interpreted by a researcher for a different reason.

In general, what is needed is an extensible model that allows neuroscientists to (1) define semantic rich metadata schemes specifying features of interest for a particular application domain, (2) use such metadata schemes to describe instances of features discovered in images at different levels of granularity, and (3) use the metadata associated with such instances in different data retrieval tasks on an image repository.

In this paper, we present the components of such a model and their realization in a database framework. The core idea underlying this model is to employ domain specific concepts as metadata schemes for the description of features of interest in images. Upon the identification of a features in an image, a researchers chooses a concept providing a metadata template for the feature and then instantiates the text-based metadata for a region of interest (ROI) in the image through a *data annotation process*. Data annotations thus can be understood as well-defined, typed links between schema like metadata structures and ROIs and can easily be employed in data retrieval tasks. There are several advantages of the model we propose. First, annotations are kept separately from images and thus several users can annotate the same image using perhaps different concepts. Second, regions of interest in an image are specified as spatial structures and thus allow fine grained data annotations instead of just whole images. Third, the underlying model allows for various text-based data retrieval scenarios. A major novelty of our model is that it supports checking for the compatibility of annotations and underlying concepts.

In the following section, we present our annotation model and its realization in a specific research project of the Human Brain Project conducted at the University of California at Davis [24]. Our primary focus is on annotating images of neuroanatomical structures of the human brain. In Section 3, we present different mechanisms we realized atop of the model to check for the compatibility of annotations in images and usage of underlying concepts. A prototype application of our approach including some basic usage scenarios for annotating images is presented in Section 4. After a review of related work in Section 5, we draw some conclusions and outline future work in Section 6.

2. Representing and Managing Conceptualized Annotations

In the following, we present the model underlying our approach to annotate regions of interest in images using domain specific concepts.

2.1. Requirements and Assumptions

An annotation of an image basically can be understood as a typed link between a spatial object (so-called *region of interest* or *ROI*) in the image and a domain specific concept representing a metadata template. Associated with the annotation are values for properties that describe the feature according to the concept. In order to specify, represent, and in particular query concepts, annotations and images in a uniform fashion, these types of information need to be not only modeled appropriately, but a respective model should also be easy to implement and use in different data management and retrieval tasks.

The model should be extensible with regard to different conceptual structures adopted as metadata schemes. Conceptual structures can include simple standard vocabulary like structures, such as Neuronames [15] or UMLS [4], as well as complex (bio)ontology like structures [2], provided such structures support the notion of uniquely identifiable concepts. Since such conceptual structures are developed in a collaborative fashion and represent various domain specific aspects, not only different views on such structures and thus annotations need to be supported, but a respective infrastructure needs to be in place to negotiate concept specifications such as the naming or properties of anatomical structures. In Section 3, we will describe some mechanisms that can be employed for realizing such infrastructure.

We assume that images are managed by an image repository which can be used by individual researchers and research groups in a collaborative fashion. Images can be registered and Dublin Core like creational metadata are associated with images and describe authorship, experiment and a very basic content description. Such metadata, different from the concepts used to annotate images, provide researchers an entry point to image data of interest. Such images are further investigated and interpreted and perhaps annotated using concepts. We assume high resolution image data that cover a wide variety of neuroanatomical and biological phenomena of the human brain, ranging from photographed slices of sections of the brain up to images of individual cells, cell structures, and nuclei, perhaps in different stages of function and/or behavior.

2.2. Annotation Graph Model

In the following, we detail an *annotation graph model* that addresses the above requirements in terms of expressiveness, extensibility, and ease in implementation. In this model, concepts, annotations, and images are represented as different types of nodes. Edges between nodes describe respective relationships such as how concepts are related and images are annotated using concepts.

Assume a set $T = \{String, Int, Date, \ldots\}$ of simple data types. Let $\mathrm{dom}(T)$ denotes the domain, i.e., the set of all possible values for T. In the annotation graph model, a property of a node is defined by an identifier and a type $PDef = String \times T$. An instantiation of a

property consists of an identifier and a value $PVal = String \times \mathrm{dom}(T)$.

As outlined in the previous sections, concepts provide templates for annotations that are associated with ROIs in images. In our model, concept are represented by a simple yet extensible form of *concept nodes*. We assume that each concept node has the following components: (1) a concept identifier *cid*, (2) a natural language definition that associates an agreed upon, well-defined meaning with the concept (*def*), (3) a set *terms* of phrases or words that are typically used to name the concept (e.g., synonyms), (4) and a set $pdefs \subset \mathbb{P}\, PDef$ of property definitions. A concept thus is similar to a class definition used in the context of object-oriented modeling. In the following, we denote the set of all concepts by $\mathcal{C}$.

The second type of node in our graph model represents *images*, which are assumed to be Web accessible, either through a direct URL or a query against the image repository. Images thus are identified by a URI (Uniform Resource Identifier). The set of all image nodes is denoted by $\mathcal{I}$. Finally, *annotation nodes* provide the basis to specify links between concepts and ROIs in images. An annotation node has an identifier and a set *PVal* of property instantiations induced by the concept underlying the annotation. The set of all annotation nodes in a graph is denoted as $\mathcal{A}$.

With each of the above nodes, further creational properties are associated, including author information, date of creation etc, and are not specified explicitly. Note that from an operational point of view creational properties of image nodes can be provided by the image repository.

The set of all nodes $\mathcal{V}$ in an annotation graph is defined as the union of the component sets $\mathcal{A}$, $\mathcal{C}$ and $\mathcal{I}$: $\mathcal{V} = \mathcal{A} \cup \mathcal{C} \cup \mathcal{I}$. Links between nodes are represented as directed, typed edges to which optional property instantiations are assigned. The types of edges are drawn from concepts (see below) and property instantiations are determined by the concepts underlying the edges. Finally, the set $\mathcal{E}$ of all edges is define as $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \times \mathcal{C} \times \mathbb{P}\, PVal$. The meaning of the components of an edge $e = (from, to, type, pvals) \in \mathcal{E}$, with *from*, *to*, and *type* being nodes (or rather node identifiers), is as follows. The edge e connects the node with id *from* with the node *to* (in this direction). With the edge e the concept with the id *type* is associated, and *pvals* is a set property instantiations induced by the concept *type*.

Based on these definitions, our annotation graph model comprises both metadata template components (concepts) and metadata instance components (annotations). An instance of the model defined by one or more users is then represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

Naturally, nodes can be connected via edges of arbitrary types (concepts). However, in most cases only edges of certain types are reasonable. In order to deal with the specific meaning of the different kinds of nodes and how they can be connected via edges, we introduce the following *default concepts*, which are assumed to be contained in any specification of a collection of concepts:

- *annotates* is used to represent edges from annotations to images. Since a basic requirement in our model is to allow for fine-grained annotations, that is, regions of interest in an image, we assume a set of *ROI descriptions* as properties of this default concept. An ROI description comprises information about the region in an image in form of a spatial object. Currently, our model supports polygons, rectangles, circles, as well as single points as spatial objects.[1]

- *annotatedBy* is a concept for representing the inverse of *annotates*, thus supporting edges from images to annotations.

- The concept *ofConcept* represents the fact that an annotation is based on a certain concept, i.e., it assigns the concept to the annotated image and instantiates the properties specified by this concept. Each annotation $a \in \mathcal{A}$ must be related to a concept, i.e., $\forall a \in \mathcal{A} : \exists c \in \mathcal{C} \wedge (a, c, ofConcept) \in \mathcal{E}$.

- *hasAnnotation* describes the inverse relationship of *ofConcept*.

It is important to note that besides these default *relationship type concepts*, other such concepts can be introduced to specify relationships among base concepts. In the context of our current research, these include in particular concepts that define spatial (e.g., *contains* and the inverse relationship *containedIn*) and type-based is-a relationships (*isA*/*hasSubtype*). Type-based is-a relationships naturally involved the inheritance of property definitions among concepts related through such a relationship, i.e., $\forall e \in \mathcal{E} : e.type = isA \rightarrow e.from, e.to \in \mathcal{C} \wedge e.from.pdefs \supseteq e.to.pdefs$. In should also be noted that the concept part of an annotation graph as it typically occurs in our project basically consists of a collection of classification hierarchies. Individual concepts can occur in one or more such hierarchies, depending on whether the focus of

[1] It should be noted that the same principles can be applied to text documents where a document is viewed as a tree like structure. In this case, XPath expressions can be used as ROI descriptions, assuming that documents are represented in X(HT)ML.

the classification is based on functional, biochemical, physiological etc. aspect. Figure 1 illustrates a typical subgraph of an annotation graph. It represents a hierarchy of base concepts related through a spatial containment type relationship concept (left side) and a type-based hierarchy (right side). Among certain base concepts in these two hierarchies, there is another relationship type concept, here representing the fact that cells of the type A typically occur in the ephithalamus.

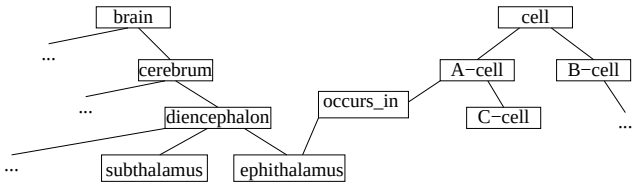

Figure 1. Concept Classification Hierarchies

In our current application for annotating images showing neuroanatomical phenomena, specifications of base and relationship type concepts as part of an annotation graph actually turn out to be very similar to cross-linked Yahoo-like hierarchies, thus providing users with an intuitive and easy to employ entry point to annotated images.

2.3. Querying an Annotation Graph

Querying and navigating an annotation graph is supported by two kinds of operations: selection and path traversal. The input for a selection operation is either one of the basic sets $\mathcal{A}$, $\mathcal{C}$ or $\mathcal{I}$ (but not a union of them) or a derived set resulting from a prior operation. Let S be one of the sets $\mathcal{A}$, $\mathcal{C}$ or $\mathcal{I}$, and $P(s)$ a predicate on $s \in S$. Then a selection operation σ_P is defined as

$$\sigma_P(S) = \{s \mid s \in S \wedge P(s)\}$$

A predicate P is a boolean expression made up of a number of clauses like *prop <op> value* which can be connected by logical connectives. In addition, path expressions of the form $rel_1.rel_2 \ldots rel_n.prop$ indicating the traversal of edges of concepts rel_1, rel_2 etc. from the current node to the property *prop* of the target node are allowed as long the result is a single-valued expression. Accessing non-existing properties always evaluates to false.

Path traversal enables following links between nodes of the graph. Given a start node v_s and a relationship type concept *rel*, the operation ϕ_{rel} returns the set of target nodes based on respective edges:

$$\phi_{rel}(v_s) = \{v_t \mid (v_s, v_t, rel) \in \mathcal{E}\}$$

Since we mainly have to deal with sets of nodes in query expressions, this operation is easily extended to set of nodes $V \in \mathcal{V}$ as $\Phi_{rel}(V) = \{v_t \mid \forall v_s \in V : (v_s, v_t, rel) \in \mathcal{E}\}$. A special kind of the path traversal operation is the computation of the transitive closure. It extends ϕ_{rel} by traversing the path indicated by the relationship as long as edges can be found that have not already been visited. The result set of nodes visited during the traversal thus is

$$\begin{aligned}\phi^+_{rel}(v_s) \quad = \quad & \{v_t \mid (v_s, v_t, rel) \in \mathcal{E} \vee \exists v_i \in \phi^+_{rel}(v_s) : \\ & (v_i, v_t, rel) \in \mathcal{E}\}\end{aligned}$$

As for ϕ_{rel}, this operation is defined on a set of nodes:

$$\begin{aligned}\Phi^+_{rel}(V) \quad = \quad & \{v_t \mid \forall v_s \in V : (v_s, v_t, rel) \in \mathcal{E} \vee \\ & \exists v_i \in \Phi^+_{rel}(v_s) : (v_i, v_t, rel) \in \mathcal{E}\}\end{aligned}$$

Using these operations, query expressions containing node selections and edge traversal can easily be formulated. The initial set of nodes for a traversal always has to be obtained by applying a selection on one of the basic sets $\mathcal{A}$, $\mathcal{C}$ or $\mathcal{I}$. Then, following edges specified by special relationship-type concepts *ofConcept*, *annotates* etc. allows to go to another type of node.

In order to provide for easy specification and implementation of services on top of the model, we have developed a simple language in the spirit of XPath. In this language, the sets $\mathcal{A}$ (*annotation*), $\mathcal{C}$ (*concept*), and $\mathcal{I}$ (*images*) are valid root elements. If views as filters on these sets are defined, they can be used as root elements as well (see also Section 2.4). Selections on nodes are formulated by appending a `[`*condition*`]` clause to a term. In *condition*, the properties of the nodes can be accessed and – in combination with the standard logical connectives – used for formulating predicates. The Φ-operator is expressed by appending */relship* to the term. *relship* denotes a relationship type concept that has to be used for following the links. The optional `+` indicates that the transitive closure has to be computed.

In the following example, we start from an image with a given URI and then retrieve the annotations associated with that image. If now the *ofConcept* relationship type concept is followed, we are able to obtain the concepts underlying these annotations, and by traversing to the annotations and images, we obtain "similar" images (ROIs), i.e., images that are annotated using the same concepts. This query can be extended further by considering a relationship type concepts, say *is-of-cell-type*. We are then able to find images that have been annotated based on more general concepts:

image[uri=...]/annotatedBy/ofConcept/

is-of-cell-type+/hasAnnotation/annotates

As another example, the query below returns the image(s) and ROI(s), respectively, that is/are linked to a given image by an annotation of a certain concept C:

image[uri=...]/annotatedBy[ofConcept.cname='C']/annotates

In Section 4, we will outline how queries expressed in this language are managed and evaluated against a database storing an instance of an annotation graph.

2.4. View Mechanism

In many emerging areas of the biosciences and in particular in Neuroscience, new domain concepts and knowledge are acquired almost every day and thus general, fully agreed upon conceptual structures among research communities in form of, e.g., standard vocabularies, do not exist. Typically, such structures and vocabularies are developed over time in individual research projects and later homogenized and made available to specific research communities. Thus, for associating concept-based metadata with images as proposed in this work, there is a strong need to support different vocabularies or conceptual structures as the basis for metadata schemes.

We support these requirements by providing *view mechanisms* on annotation graphs. On each of the base component sets, a view can be defined. More precisely, a view specifies a (virtual) sub-graph $\mathcal{G}'$ of the annotation graph: $\mathcal{G}' = (\mathcal{V}', \mathcal{E})$ with $\mathcal{V}' \subset \mathcal{V}$. $\mathcal{V}'$ is specified by formulating queries using operations presented in Section 2.3 and which restrict the set of annotations, concepts, and images to be considered in selection and graph traversal.

For example, if we want to provide a view containing only (1) concepts that have been introduced by a certain author and (2) annotations made this year, we could define this as follows[2]

```
define view my_view as
  annotation := annotation[created>='01/01/02']
  concept    := concept[author='Jim Smith']
```

If a base component set is not involved in the view definition, e.g., the set $\mathcal{I}$ of images as in the above case, the complete base set is used by default. For restricting queries in views, any valid query expression is allowed as long as it returns a proper subset of a base set, for example, a set $\mathcal{C}' \subset \mathcal{C}$ for the concept set. Since the set $\mathcal{C}$ of concepts contains default relationship type concepts such as *annotates*, *ofConcept* etc, this set is handled in a special way, guaranteeing the inclusion of such default concepts in each view.

Views are used in queries by simply giving the name of the view as an additional parameter of the query service invocation. For example, the invocation

```
query("image['uri=...']/annotatedBy",
      my_view)
```

is evaluated based on the annotation sub-graph defined by `my_view`.

It should be noted that from a practical perspective, views are a viable approach for protecting researchers from concept or annotation "overload". When a single image annotation service is used by researchers from different domains, it will probably contain large portions of concepts and annotations that are not of interest for all researchers in all these domains.

3. Analysis and Synthesis of Annotations and Concepts

In order for concepts and annotations created by different researcher to be useful in data retrieval tasks, mechanisms need to be devised that ensure a certain degree of compatibility among concepts and annotations. In this section, we present the basic principles underlying the realization of such mechanisms in the context of annotating images in the Neuroscience.

3.1. Overview

As indicated in the introduction, in order for metadata to be useful, it is essential to devise mechanisms that guard against inconsistent or incompatible metadata and metadata schemes. There have been major advancements in the development and usage of metadata schemes for Web-accessible data, but there has been little attention given to the correctness and plausibility of metadata associated with data. The problem obviously is that it is hard to precisely define what consistent metadata are and thus to develop respective mechanisms.

In the context of annotating images using concepts specific to a Neuroscience application domain, we have devised such mechanisms. The core idea behind these mechanisms is to (1) exploit region information associated with annotations, and (2) investigate relationships between the concepts underlying these annotations. Depending on what spatial properties such regions have and what relationships exist among the annotations describing ROIs, the user can be provided with feedback about possible incompatibilities. It should be noted

[2]Currently, we do not provide a view definition language. Instead, a view is defined in context of a dedicated service.

that no precise definition of consistent annotations and concepts is possible in this context since annotations and concept specifications typically are based on a specific perception and interpretation a neuroscientist has regarding a real-world concept or image representing some data specific to an application domain. A respective framework thus has to provide the user with mechanisms that allow her to specify (1) what is considered to be possibly incompatible and (2) how to react to an incompatibility. The latter aspect necessitates certain annotation policies the user can specify and which describe actions to be performed in case incompatibilities have been discovered.

In the following, we discuss a framework in which mechanisms checking for the compatibility of annotations and concepts has been devised. In order to have a workable but still useful setting, our mechanisms are based on two common types of concept classification hierarchies, namely those based on spatial containment (i.e., the spatial containment of one brain structure in another brain structure), and the type-based classification of brain structures (with a particular focus on cells and nuclei). An excerpt of two such hierarchies is shown in Figure 1 where the left hierarchy is based on spatial containment and the right hierarchy is based on type-based cell classification.

3.2. Annotation-level Mechanisms

Assume the scenario where a user interprets an image $i \in \mathcal{I}$ and that she chooses concept c underlying the new annotation a of a particular region r in i. Based on this information, the task of annotation level mechanisms is to determine other annotations in i that might be incompatible with the new annotation a. Let $A_i = \{a_1, \ldots, a_n\}$ denote all annotations that already exist for the image i (under the user specified view). For each annotation $a_k \in A_i$, let c_k be the concept underlying a_k, and let r_k and $pval_k$ be the region information and properties, respectively, associated with a_k. This information can easily be obtained through querying the annotation graph instance.

Based on A_i, there is one procedure that partitions A_i into three disjoint sets A_{same}, A_{over} and A_{disj}, based on the spatial relationships the existing annotations have with respect to the new annotation a. The set $A_{same} \subseteq A_i$ contains all annotations that employ the same region as a. The sets A_{over} and A_{disj} are defined similarly for overlapping and disjoint regions associated with annotations.

A subsequent set of procedures then checks for each annotation in such a set how the concept underlying the annotation is related to the new annotation a. Before we detail these procedures, we first describe the partitioning of the set A_i.

Since regions in an image can be free drawings (circles, rectangles, or polygons), there is no precise and general definition of what "same region" actually means. In our framework, we thus employ user-defined predicates, which are typically detailed by a group of researchers and are based on some agreed upon criteria. For our annotation level mechanisms, we provide the user with three types of predicates, which are defined on pairs of regions and determine the spatial relationship among these regions (see also Figure 2).

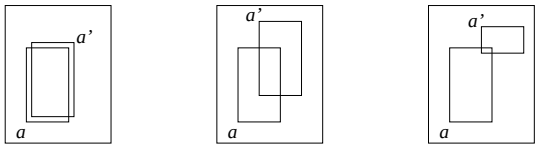

Figure 2. Spatial relationships among regions (ROIs) underlying annotations in an image

same_region(r_1, r_2, sr_threshold) evaluates to true if there is an overlap among the two regions r_1 and r_2 of more than *sr_threshold* percent. In this case, the two regions are considered to be equal. The predicate *overlap_region(r_1, r_2, or_threshold, sr_threshold)* evaluates to true if there the two regions r_1 and r_2 overlap more than *or_threshold* percent but less than *sr_threshold* percent. The predicate *disj_region(r_1, r_2,dr_threshold, or_threshold)* used to determine whether two regions are disjoint is devised similarly. Checking these predicates for each annotation $a_k \in A_i$ and the new annotation a results in a partitioning of A_i into A_{same}, A_{over}, and A_{disj}. For each set, now individual mechanisms are applied that check for possible incompatibilities among the concepts underlying the annotations. Our main focus will be on the case where two annotations are based on the same regions. Mechanisms for cases where two regions are overlapping or disjoint can be devised in an analogous fashion and are only outlined.

Same Region. Assume an annotation $a_k \in A_{same}$ based on concept c_k and the new annotation a based on concept c. There are three cases to consider:

- $c = c_i$: Both annotation are based on the same concept. If they also have the same properties (*pval*, see Section 2), then the annotation a is redundant (case 1). The equivalence of properties is checked based on another function *match_properties*: $A \times A \rightarrow [0, 1]$ in order to account for similarities among values users choose to describe instances for concepts. If the value determined by the function is below a certain user-

specified threshold, the properties are considered to be different and thus a *data conflict* is determined (case 2). In this case, the mechanism triggers a respective action, e.g., a negotiation process with the user who specified the annotation a_k.

- $c \neq c_i$: The two annotations are based on different concepts. We refer to such a situation as *concept reference conflict* whose handling will be detailed in the following.

As indicated in Section 3.1, our main focus is on concept classification hierarchies that are based on spatial containment and sub-type/super-type relationships. Assume two annotations a and a_k, based on concepts c and c_k, respectively. If there is no (direct) relationship between c and c_k, then the two annotations are likely to reflect different views on the same ROI. The user annotating the image i then can initiate respective actions through the specification of annotation policies, as indicated in Section 3.1. The more interesting cases are when c and c_k belong to the same classification hierarchy.

Assume a hierarchy based on spatial containment. If c is a (direct) sub-concept of c_i, then the annotation a is either to fine-grained or the annotation a_k is to coarse-grained. That is, for either annotation a different region needs to be specified. The analogous case holds if c is a (direct) super-concept of c_k. In both cases, a review process is triggered by the annotation level mechanisms which then allow the two users who made the annotations a and a_k to review and negotiate a correct annotation. A similar scenario holds when both concepts belong to the same classification hierarchy but there is no super/sub-concept relationship between the two concepts. This case indicates that either annotation is based on a misclassification of the feature described by the annotations. The above scenario is adopted in an equivalent fashion where the two concepts c and c_k belong to the same type-based classification hierarchy.

Overlapping and Disjoint Regions. Mechanisms that are applied to the two sets A_{over} and A_{disj} are based on the same principles adopted for annotations that are associated with the same ROI. In particular, cases where the two concepts are specified in a concept hierarchy based on spatial containment can be handled in exactly the same fashion. For example, if two concepts c and c_k are associated with two disjoint regions r and r_k and there is a spatial containment among the two concepts in terms of a super/sub-concept relationship, then this clearly indicates a possible misclassification. A simple example, based on Figure 1, is when a user associated a region with the concept diencephalon and another user associates an overlapping (or disjoint) region with the concept cerebrum. In order for the two annotations to be compatible, there must be no overlap among the two regions, but containment since the concept diencephalon is a super-concept of ephithalamus in the spatial containment hierarchy.

3.3. Concept-level Mechanisms

Concept-level mechanisms are invoked whenever a user creates or modifies a concept or introduces a new relationship type concept. Achieving the desired functionality of such mechanisms is much more critical for the overall approach since concepts provide the basis for annotations and thus require a high degree of compatibility in terms of consistency and non-redundancy.

For the specification of a concept, we employ a function that determines the similarity among a new and existing concepts and which is automatically invoked by the mechanism. Basis for this function, named *similar_concept*, are two components that individually check the similarity among terms and properties between the new and an existing concept using user-defined weights. Formally, the function is defined as

$$\begin{aligned} \mathit{similar\_concept}(c_1, c_2) := t * \mathit{sim\_terms}(c_1, c_2) + \\ p * \mathit{sim\_prop}(c_1, c_2) \in \mathcal{R}[0, 1] \end{aligned}$$

with t, p being weights such that $t + p = 1$. The functions *sim_terms* and *sim_prop* determine the similarity among individual components of two concepts c_1 and c_2 using word and phrase similarity measures as they are used in, e.g., schema matching approaches in database integration [19], natural language processing techniques [14], or approaches in consolidating clinical terminology [17]. Each function returns a value between 0 and 1, which is then passed to the function *similar_concept*. Upon the creation of a concept, the mechanism realizing the above function provides the user with a list of similar concepts, ranked based on their computed similarity value. Besides comparing the new concept to the existing concept, our approach in particular provides the user with means to investigate how similar existing concepts have been used in annotating images. In that respect, our framework provides the user with more functionality than just simply checking similarity among terms and properties used for specifying concepts.

The latter aspect is also of particular concern for mechanisms that check for possible incompatibilities among concepts *after* concepts have been specified and used for annotating images. The basic idea for this is that for pairs of concepts the similarity value is recorded. This is only done for pairs of concepts that are not be considered similar but whose similarity value is above a certain threshold. For these pairs of con-

cepts, at user specified times, automated mechanisms check how these concepts have been used to annotate data. The invocation of the mechanism can either occur on a regular basis (e.g., weekly) or based on how many annotations have been made using these concepts. If it turns out that the two concepts have been used to basically annotate the image (or rather ROIs), then these two concepts are likely to be similar. The realization of such checks again utilizes the notion of *same_region* as introduced in the previous section for analyzing the compatibility of annotations.

4. Prototype Application

In this section, we briefly describe the application of the presented framework as part of a collaborative environment for annotating images in the context of the Human Brain Project. In this project, brain slices are digitally photographed under microscope and utilized by researchers who mark specific regions (e.g. cell structures) and assign concepts (e.g., a certain cell type) to these regions. The annotation graph model is used to represent concepts, annotations and images as well as their relationships. Furthermore, the query operations allow to formulate declarative queries for retrieving elements and traversing the graph.

The implemented annotation system follows the typical client/server paradigm. Basic services for defining concepts and assigning annotations as well as formulating queries are provided by an annotation server, whose architecture is shown in Figure 3.

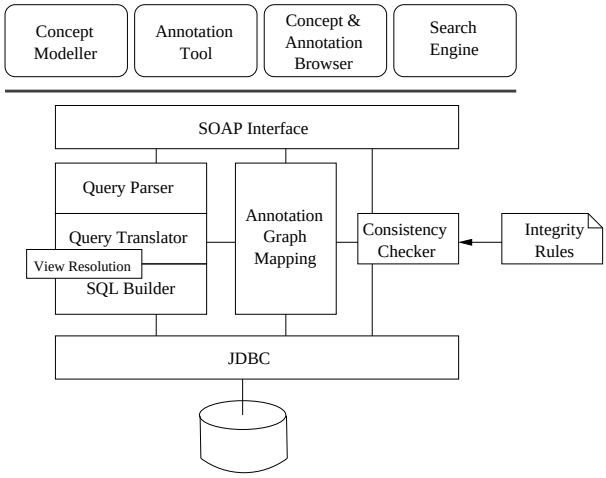

Figure 3. System architecture

The main component is the graph mapping module, which represents an annotation graph by mapping its nodes and edges to relations stored in a relational database. Tightly related to this is the query component consisting of a parser for the language described in Section 2.3, the translator which transforms a query into a relational algebra expression by applying a set of transformation rules [9] and the SQL builder for deriving SQL queries from such expressions. The translator also implements the view mechanism by substituting all references to the basic sets (concept, annotation, images) in a query by the restricting expressions of the according view definition. Finally, the SQL query is sent to the DBMS for evaluation. For an efficient evaluation of similarity predicates both as part of queries as well as for consistency checking we are currently investigating the usage of DBMS cartridges for text and spatial data.

The components of the system are implemented in Java using JDBC for accessing the DBMS. The interface to the services of the system is realized using SOAP. In this way, the annotation server can be used as a Web Service by different (possibly Web-based) tools as shown in Figure 3.

A screenshot with two of these tools is given in Figure 4. On the lower left side, the concept browser is shown, which is used for querying and browsing the concepts. It offers different views, e.g., a simple tabular presentation of query results as well as tree presentations, where the primary relationship can be chosen by the user (for example, *contains* for browsing containment hierarchies and *hasSubtype* for specialization hierarchies). Beside visualizing the concept space, a second function of this browser is to select a concept for creating a new annotation. The latter step can be performed with the help of the annotation tool shown behind the browser. It allows to mark regions of interest in an image and to assign an annotation based on the previously selected concept.

This requires to chose a certain concept from the available set or sometimes – if new structures are discovered – to define a new concept, possibly as a specialization of another one. In the latter case, the consistency/compatibility checking mechanisms (cf. Section 3) are involved to notify the user about possible conflicts or redundancies. In addition, concepts act as a kind of template for annotations by defining a set of properties which have to be instantiated, i.e., by specifying values for the annotation.

A collection of annotated images can later be used to visualize and explain various structures of the brain. For this purpose, images are shown together with their annotations, which not only give an explanation on important regions of the image, but allow also to follow links to the concept underlying the annotation, to re-

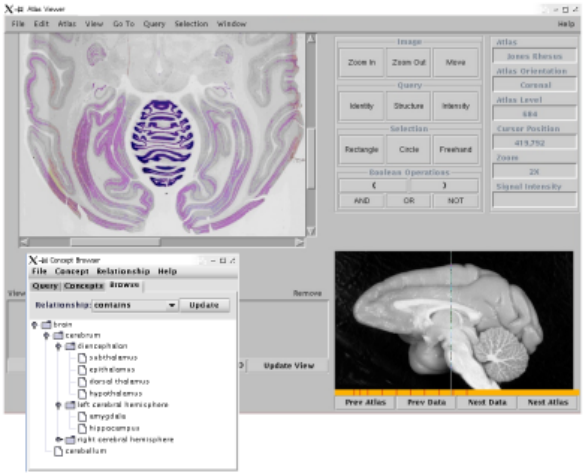

Figure 4. Tools from the annotation system

lated concepts and finally to images annotated with these concepts. In this way, an atlas of the human brain can be built, consisting of marked regions in images which are linked by concepts.

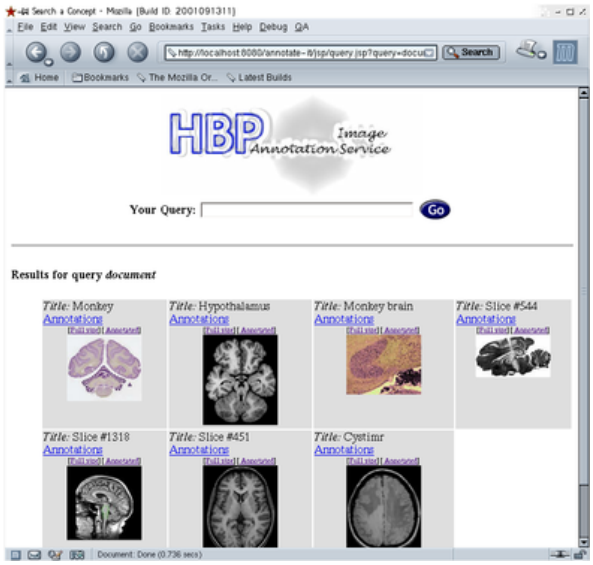

Figure 5. The search engine

Finally, Figure 5 shows a screenshot of the search engine for image annotations. It simply evaluates queries formulated in our query language using the query engine of the annotation service and displays the results. In addition, with each image the associated annotations and concepts are shown as links referring to a page with details about them including further links to related concepts, images etc.

5. Related Work

There is an increasing amount of work on models and methodologies to semantically enrich the Web (see www.semanticweb.org for an extensive overview). The major focus in these works is on building semantic rich and expressive ontology models that allow users to specify domain knowledge. The most prominent approaches in this area the Ontobroker project [6, 7], SHOE [10], the Topic Maps standard [23] as general ontology frameworks, and TAMBIS [21] and OIL [22] as specific ontology frameworks tailored to the biological domain. We consider these ontology-centric works as orthogonal to our annotation-centric approach. Also, most of these work do not put much emphasis on how remote Web documents or images can be annotated by different users at a fine-level of granularity. We consider the need for external and fine-grained annotations as essential and appropriately include these aspects in our model for annotating scientific images. Furthermore, whereas the above approaches concentrate on querying ontologies using, e.g., RDF-based languages, our focus is to have a simple, expressive, and easy to implement language that allows to query all three components, concepts, annotations, and Web accessible images in a uniform fashion.

At the other end of the spectrum, several systems have been proposed that provide users with means to annotate data. This includes the multivalent document approach [18], the SLIMPad approach [5], the Annotea project [13] as well as some commercial systems (see, e.g., [8, 11] for an overview). While none of these approaches supports a query framework for annotations, only [5] support the notion of concept like structures underlying annotations. Finally, to the best of our knowledge, there has been no work that considers the aspects of the consistent usage of metadata in annotating or enriching Web accessible documents.

6. Conclusions

In many computational sciences, the association of different types of metadata with heterogeneous and distributed collections of scientific data play a crucial role in order to facilitate data retrieval tasks in an integrated and uniform way. In this paper, we have presented an approach that allows researchers to associate well-defined metadata in form of concept instances with image data. Although our focus primarily is on image data as they typically occur in the Neurosciences, the underlying model of data annotations and concept-based metadata templates is applicable to a wide variety of different forms of scientific data. The model

and its realization provide all features researchers in collaborative research environments deem necessary to enrich (possibly remote) data and thus to "semantically index" data that is not easy to classify or analyze otherwise. In particular, we have shown how properties of concepts and annotated regions in images can actually be used to investigate the compatibility or consistency of metadata associated with images.

While the usage of the first prototype of our system confirms this hypothesis, several new challenges come up. These include aspects of scalability of the system as well as efficiency and effectiveness of user interfaces. Due to the centralized graph storage, the presented architecture and its services are appropriate only for smaller communities. A distributed approach using multiple instances of an annotation graph and a distribution of data retrieval service alleviate such problems.

References

[1] I.N. Bankman (Editor-in-Chief): *Handbook of Medical Imaging – Processing and Analysis*, Academic Press, 2000.

[2] 3rd annual Bio-Ontolgies Workshop – Sharing Experiences and Spreading Best Practice. La Joalla, CA, www.ingenuity.com, August 2000.

[3] M. Chicurel: Databasing the brain. *Nature* 406:822-825, 2000.

[4] K.E. Campbell, D.E. Oliver, E.H. Shortliffe: The unified medical language system: towards a collaborative approach for solving terminology problems. *Journal of the American Medical Informatics Association*, Volume 8, 12–16, 1998.

[5] L.M. Delcambre, D. Maier, S. Bowers, M. Weaver, L. Deng, P. Gorman, J. Ash, M. Lavelle, J. Lyman: Bundles in Captivity: An Application of Superimposed Information. In *Proc. of the 17th International Conference on Data Engineering (ICDE 2001)*, IEEE Computer Society, 111-120, 2001.

[6] S. Decker, M. Erdmann, D. Fensel, R. Studer: Ontobroker: Ontology based Access to Distributed and Semi-Structured Information. In *Database Semantics - Semantic Issues in Multimedia Systems, IFIP TC2/WG2.6 Eighth Working Conference on Database Semantics (DS-8)*, 351–369. Kluwer, 1999.

[7] D. Fensel, J. Angele, S. Decker, M. Erdmann, H.-P. Schnurr, S. Staab, R. Studer, A. Witt. On2broker: Semantic-based access to information sources at the WWW, 1999. In *Proceedings of the World Conference on the WWW and Internet (WebNet 99)*, 1999.

[8] J. Garfunkel: Web Annotation Technologies. look.boston.ma.us/garf/webdev/annote/software.html

[9] M. Gertz, K. Sattler: A Model and Architecture for Conceptualized Data Annotations. Technical Report, Department of Computer Science, University of California, Davis, 2001.

[10] J. Heflin, J. Hendler: Dynamic Ontologies on the Web. In *Proc. of the 17th National Conference on Artificial Intelligence (AAAI 2000)*, 443–449, AAAI/MIT Press, 2000.

[11] R.M. Heck, S.M. Luebke, C.H. Obermark: A Survey of Web Annotation Systems, www.math.grin.edu/∼luebke/Research/Summer1999 /survey_paper.html.

[12] S. Koslow, M. Huerta (eds.): *Neuroinformatics: An Overview of the Human Brain Project.* Lawrence Erlbaum Associates, NJ, 1997.

[13] J. Kahan, M.-R. Koivunen, E. P. Hommeaux, R. R. Swick: Annotea: An Open RDF Infrastructure for Shared Web Annotations. In *Proceedings of the 10th International World Wide Web Conference (WWW10)*, 623–632, ACM, 2001.

[14] C.D. Manning, H. Schütze: *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.

[15] braininfo.rprc.washington.edu/mainmenu.html, Neuroscience Division, Regional Primate Research Center, University of Washington.

[16] Neuroinformatics – The Human Brain Project. www.nimh.nih.gov/neuroinformatics/index.cfm.

[17] D.E. Oliver: Synchronization of Diverging Versions of a Controlled Medical Terminology. In *Proceedings of the 1998 AMIA Annual Fall Symposium*, 850–854, 1998.

[18] T. A. Phelps, R. Wilensky: Multivalent Annotations. In *Research and Advanced Technology for Digital Libraries – First European Conference*, 287–303, LNCS 1324, Springer, 1997.

[19] E. Rahm, P.A. Bernstein: A Survey of Approaches to Automatic Schema Matching. *VLDB Journal* 10(4):334–350, 2001.

[20] R.A. Robb: *Biomedical Imaging, Visualization, and Analysis.* Wiley-Liss, 2000.

[21] R. Stevens, P. Baker, S. Bechhofer, G. Ng, A. Jacoby, N.W. Paton, C.A. Goble, A. Brass: TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources. *Bioinformatics* 16(2):184–186, 2000.

[22] R. Stevens, C. Goble, I. Harrocks, S. Bechhofer: Building a Bioinformatics Ontology using OIL. To appear in a special issue of IEEE Information Technology in Biomedicine on Bioinformatics, 2001.

[23] Topic Maps. www.topicmaps.org

[24] UC Davis/UC San Diego Human Brain Project Informatics of the Human and Monkey Brain, neuroscience.ucdavis.edu/HBP.

[25] A. Wong, S. Lou: Medical Image Archive and Retrieval. In *Handbook of Medical Imaging – Processing and Analysis*, Academic Press, 771–783, 2000.