Pergamon

0306-4379(94)E0004-9

Information Systems Vol. 19, No. 4, pp. 33–54, 1994

0306-4379/94 $7.00 + 0.00

KMED: A KNOWLEDGE-BASED MULTIMEDIA MEDICAL DISTRIBUTED DATABASE SYSTEM

WESLEY W. CHU[1], ALFONSO F. CÁRDENAS[1] and RICKY K. TAIRA[2]

[1] Computer Science Department, UCLA
[2]Department of Radiological Sciences, UCLA School of Medicine

(Received 18 June 1993; in final revised form 1 November 1993)

Abstract — The objectives of the Knowledge-Based Multimedia Medical Distributed Database System (KMeD) are to: query medical multimedia distributed databases by both image content and alphanumeric content; model the temporal, spatial, and evolutionary nature of medical objects; formulate queries using conceptual and imprecise medical terms and support cooperative processing; develop a domain-independent, high-level query language and a medical domain user interface to support KMeD functionality; and provide analysis and presentation methods for visualization of knowledge and data models. Using rules derived from application and domain knowledge, approximate and conceptual queries may be answered. These concepts are validated in a testbed linked with radiology image databases. The joint research between the UCLA Computer Science Department and the School of Medicine assures that the prototype system is of direct interest to medical research and practice. The results of this research are extensible to other multimedia database applications.

Key words: multimedia databases, scientific databases, medical images, cooperative query answering, approximate query answering

1. INTRODUCTION

1.1. Background

Medical research and practice have several advanced database requirements. Firstly, on-line access to the large amount of multimedia data stored in various patient information systems is needed. These clinical information systems include: (1) hospital information systems (HIS), (2) radiology information systems (RIS), (3) picture archiving and communication systems (PACS) [5], and (4) various research database systems (pathology, genome mapping, brain mapping, etc.) [5]. This data includes alphanumeric structured data (e.g. demographic information), free text with imprecise medical terms and descriptions (e.g. pathology reports), images (e.g. computed tomography, magnetic resonance), and voice data. Secondly, medical research databases need to query the collected information repository using not only traditional query keys (e.g. patient hospital ID, sex, date of birth, etc.) but also using more sophisticated query predicates based on image content, evolutionary transformation of objects (e.g. organ development, disease progression, etc.), spatial relationships between objects, and temporal relationships between object events. Thirdly, the most sophisticated medical research systems require the computer to respond to queries in a manner that mimicks and assists trained medical specialists. This requires a complete physical model of the structure and dynamics of the organ system of interest.

We have developed a prototype system named the Knowledge-Based Multimedia Medical Distributed Database System (KMeD). It uses a multi-layered architecture and global data model to integrate heterogeneous information systems. We developed a temporal evolutionary data model (TEDM) [5] to account for the changing characteristics of medical objects. This allows answering of queries involving evolutionary changes in objects (e.g. organs) occurring in maturation, disease, therapeutic, and aging processes. We have also developed methods to describe the spatial semantics of medical images and new methods for knowledge-based cooperative query processing [5].

Spatial relationships are an important piece of medical knowledge. A physician's mental model of the patient includes understanding the spatial extents, orientations, boundaries, adjacent structures, etc. of organs and other structures in the human body. These mental models are adjusted depending on the maturity of the patient and the genetic, psychiatric, dietary, and demographic profiles of the patient. Medicine thus critically relies on knowing where body structures are located and their locations relative to other structures. An abnormality is defined as a gross deviation from the anatomical models. The sequencing and relationships of body structures in time and space are often critical to the diagnosis, prognosis, and mechanics of human disease.

Current database management systems with spatial query answering capabilities tend to be inefficient, time consuming, and computationally intensive. More importantly, they do not provide cooperative query answering facilities (approximate query answering, relaxation control, and associative query answering). Current methods use slow operators which work at the image or point set level to determine spatial relationships among objects. These systems are not appropriate for searching large medical databases with millions of images and with individual images comprising very large matrices (4K x 4K x 12 bits) and several tens of temporally or spatially related frames. We feel that image semantics is greatly underutilized for query processing due to a lack of definition and organization of image data. We are developing a spatial evolutionary data model (SEDM) [5] to capture, classify, and represent the semantic description of objects in an image as well as objects contained across related images (i.e. image stacks). The semantics of an image is expressed in terms of a set of basic image features, a set of operators to calculate derived features, and data hierarchies and rules that define high-level complex features. Our feature-based, set-oriented approach to query processing should greatly improve spatial query performance.

KMeD includes a cooperative query answering layer which uses domain knowledge and inferencing techniques to make intelligent decisions on localizing a query's search space, regulating a query's solution space, conceptualizing complex data entities and processes, and associating context-dependent subjects. For example, this layer provides constructs for the structured joining of a phenotypical (symptomatic) disease classification hierarchy with a genotypical (causation) disease classification hierarchy. This will allow users to search various conceptual levels of a patient's symptom space and/or underlying disease process space, and cross-correlate these symptoms with their probable underlying disease processes. This layer can assist the medical researcher's quest to understand and integrate the complex relationships between patient symptoms, imaging methods, diagnostic image features, underlying disease pathology, treatment methods, and associated treatment risks. These are questions of fundamental concern to all medical researchers.

1.2. Motivating Scenario and Objectives

This section first presents the background and overview of clinical examples used to demonstrate the important features of our work, and then the specific objectives we pursue. We present these paradigms in order of increasing sophistication and comment on the need for the TEDM and SEDM models and/or flexible query capabilities for each instance. The examples will focus on two anatomical regions: the hand and the brain. The hand is used because of its familiarity, the availability of robust hand image segmentation software, and its importance in diagnosing several musculo-skeletal diseases. The brain is used because of its complexity, its association with various body subsystems (neurological, endocrine, etc.), and its importance in the control of motor and mental functions.

1.2.1. Example Queries

Our first set of queries involves the hand. EX1 below demonstrates the need for temporal object management using a predicate to restrict an evolutionary event. EX2 refers to an abnormality called Turner's Syndrome. This is a genetic disorder in which the patient's sex chromosomal genotype is 'XO'.

Turner's Syndrome is frequently diagnosed from projectional hand X-rays and is characterized by stubby 3rd, 4th and 5th metacarpals (bones in the palm area). EX2 demonstrates: (1) query by image content and (2) cooperative query processing using type abstraction hierarchies with association.

EX1 *"Retrieve an image sequence of a patient demonstrating the fusion of the thumb metacarpal*

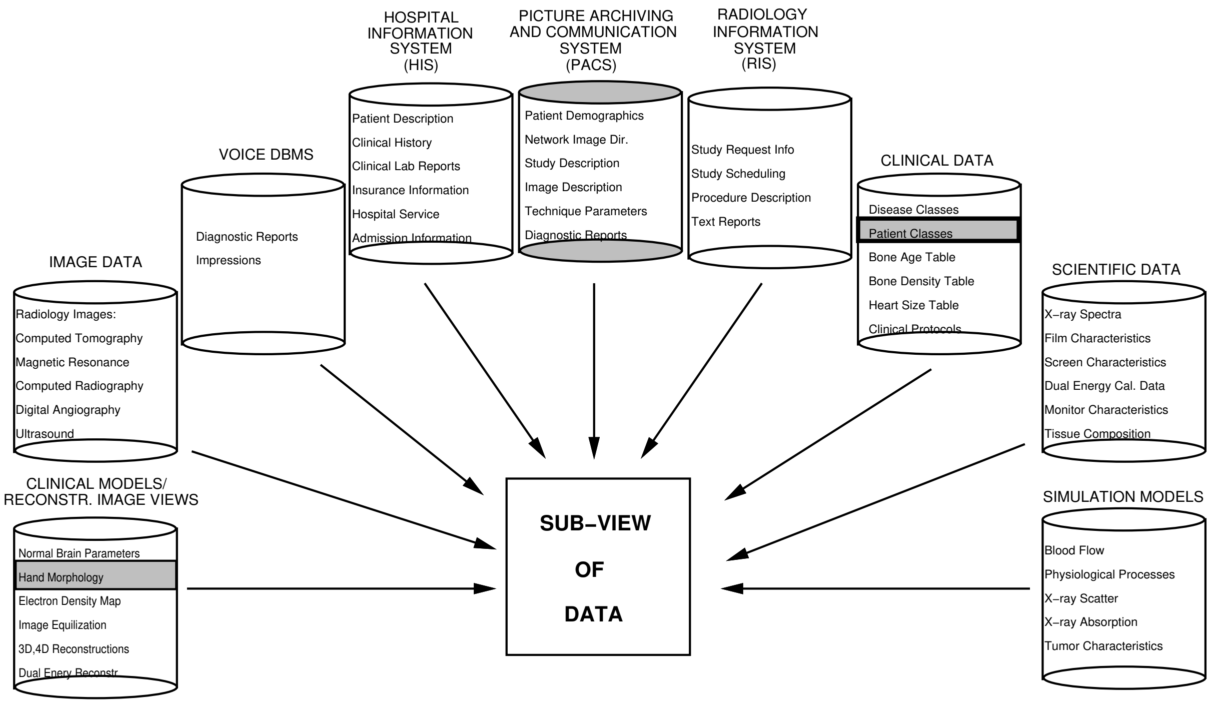

Fig. 1: UCLA Medical Center multimedia database federation. Each of the database systems shown were imlemented independently.

metaphysis with the thumb metacarpal epiphysis. Visualize the sequence in a movie loop." (see Section 2.2.3 for details)

EX2 *"Retrieve all hand X-rays of 12-year-old Korean-American patients with Turner's Syndrome."* (see Section 2.3.1 for details)

The next set of queries involves searching cases demonstrating various brain tumors. EX3 demonstrates a query that requires spatial constraints as to the tumor location and the bordering invading structures. EX4 demonstrates a complex query involving spatial constraints on an evolving brain adenoma (an adenoma is a benign glandular tumor). EX4 is presented to illustrate the ability of the system to provide additional relevant information and/or queries for a given query in order to provide more useful answers.

EX3 *"Retrieve all image cases demonstrating the invasion of an adenoma into the sphenoid sinus in pre-adolescent patients."* (see Section 2.2.4 for details)

EX4 *"Retrieve image cases demonstrating a pituitary gland microadenoma that evolved into a macroadenoma with suprasellar extension pressing against the optic chiasm."* (see Sections 2.2.4, 2.3.2, and 2.4.1 for details)

Medical research and practice is hindered by the fact that multimedia data is distributed and managed under different hardware and software systems. Furthermore, medical researchers investigating similar areas of study are often unaware of the existence of databases which contain relevant information. Figure 1 shows the major medical data repositories present in our institution, most of which are developed independently of each other by different organizational units.

1.2.2. Objectives

The focus of our research is to support general medical research and clinical treatment which involve understanding image and alphanumeric features and correlating patient symptoms with underlying disease processes for various patient profiles.

The major objectives of KMeD are:

1. to query medical multimedia distributed databases by both image and alphanumeric content,

2. to model the temporal, evolutionary, and spatial natures of objects (e.g. features such as bone growth) and to enable queries based on this modeling,

3. to formulate and answer conceptual and imprecise queries by relaxation, and to provide relaxation control for satisfying user constraints and regulating the size of returned answers from a cooperative query system,

4. to provide relevant "value-added" information as part of the query answer even though it is not explicitly requested (an ability called *associative query answering*),

5. to provide a domain-independent high level query language and a medical domain-oriented, graphically interactive user interface, and

6. to provide analysis and presentation methods for visualization of data and knowledge models.

In this paper, we shall first present the system's architecture and functionality, consisting of four layers: the autonomous database layer, the canonical/distribution layer, the cooperative query answering layer, and the presentation layer. Such a layered architecture allows system extensibility and scalability. Next, we describe our testbed facilities then finally outline our conclusions.

2. SYSTEM ARCHITECTURE AND FUNCTIONALITY

The Knowledge-Based Multimedia Medical Distributed Database System (KMeD) which we have been developing at UCLA aims to provide transparent access and integration over many diverse and distributed types of data, including alphanumeric values, pictures or images, and voice samples. The system's multilayer architecture is shown in Figure 2. This architecture is an evolution of our earlier heterogeneous distributed database architecture [5], adding full cooperative functionality and spatial/evolutionary semantics for query processing.

The general working approach of our current system consists of the following procedural steps:

1. Images and associated alphanumeric and text data are transferred to KMeD from the Radiology Department's PACS (Picture Archiving and Communication System) and other medical data sources.

2. High-level features are extracted from each image either through special image analysis software or through the intervention of an expert in the field.

3. The extracted features are integrated into the database by way of the data models which we have been/are developing.

4. The database can now be queried and/or processed by image and alphanumeric contents.

Since medical images are almost never modified, it is efficient to preprocess them in order to extract their constituent objects.

We shall now detail the layers, the data models, and the access facilities of KMeD in the following order: the autonomous database layer, the canonical/distribution layer, the cooperative query answering layer, and the presentation layer.

2.1. Autonomous Database Layer (ADL)

The ADL consists of two further sublayers: the *local DBMS layer* and the *local canonical layer*.

The local DBMS layer is essentially the federation of autonomous databases, consisting of commercial DBMSs and other specialized data management systems running within their own data models and data manipulation/query languages. The local canonical layer handles any translation

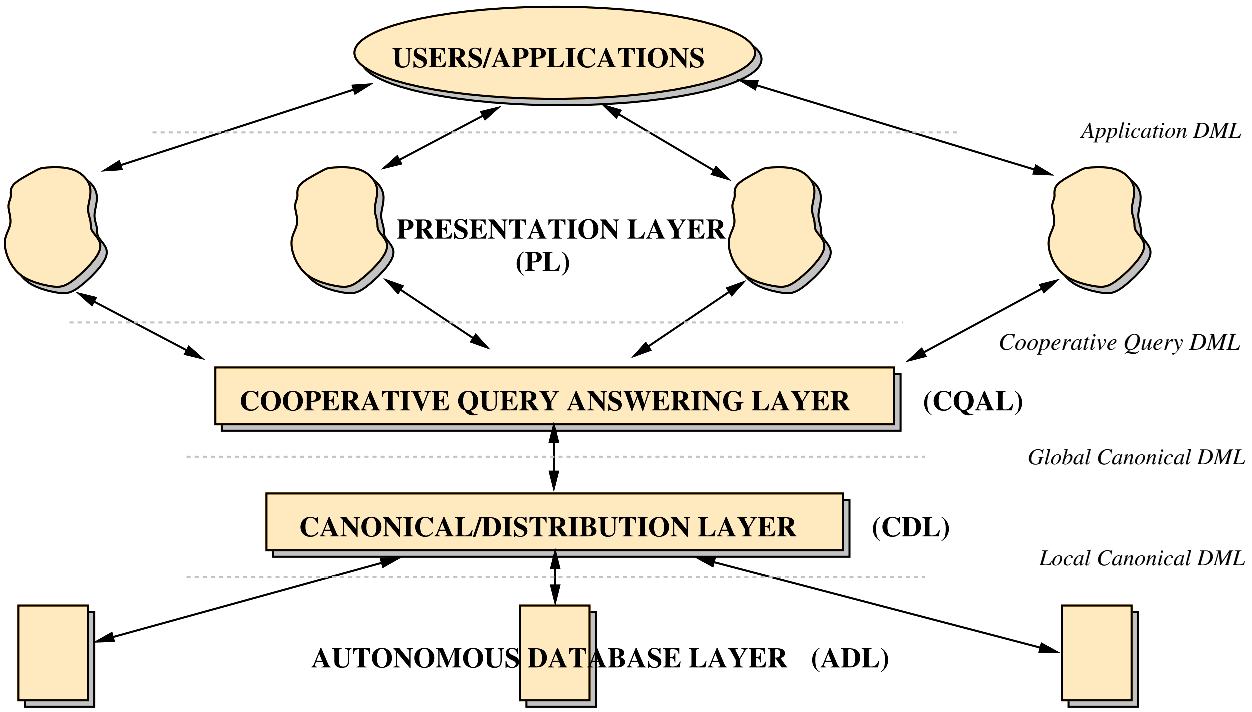

Fig. 2: *KMeD multilayer database architecture.*

or data manipulation needs for linking the local and autonomous data models and languages with the *local canonical data model*, which is the first step toward full heterogeneous integration [5].

At the highest level, the KMeD user can query the database without regard for the true source of the resulting data.

2.2. Canonical/Distribution Layer (CDL)

The CDL manages the aggregation of the local canonical database views into collected conceptual database descriptions and the decomposition of these global views back into local databases. Thus, we provide powerful constructs by which a conceptual object's attributes can be a sum of features from multiple underlying heterogeneous data models.

As shown in Figure 3, our image/picture data model consists of two layers. In the lower layer, related images are stored in a dynamic stacked data structure. In the upper layer, the semantics and complex relationships among the image objects are modeled at various conceptual levels, using aggregation, generalization/specialization, temporal, evolutionary, and spatial relationships.

2.2.1. Feature Extraction

The Picture Archival and Communication System (PACS) stores a variety of medical images, including X-rays and MRI. Each image type has its own specialized image segmentation methods which take advantage of its image characteristics; they range from fully automatic segmentation software to semi-automatic computer-assisted tools. Additional information is available through diagnosis by physicians. Advances in image segmentation techniques greatly influenced the design strategy of the image management system. This strategy involves extracting an image's features selectively, then organizing and storing these features in a manner that provides efficient content-based retrieval.

Preprocessing of the images consists of the following steps:

1. Identify the objects in the images.

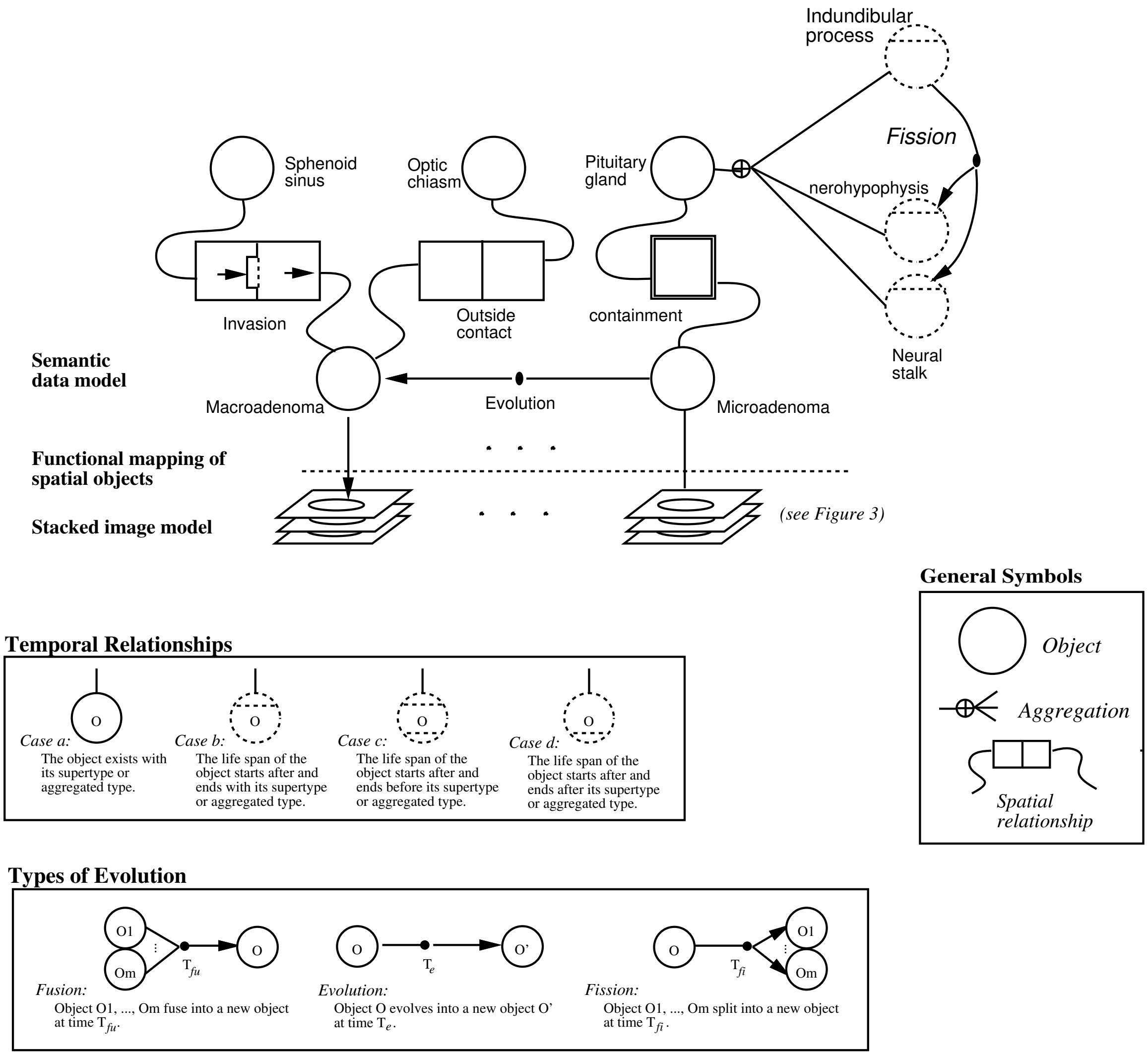

Fig. 3: A two-layered data model for modeling the semantics of KMeD image data.

2. Detect the boundaries of the objects.

3. Obtain the spatial features and relations of the image objects.

Current approaches can be broadly categorized as based on intensity discrimination [5] or edge detection [5]. The intensity approach assumes that a fundamental relationship exists between pixel intensity and the physical substrate. However, numerous studies have shown that factors such as field inhomogeneity, instrumentation noise and partial volume effects all contribute to pixel intensity fluctuation and create uncertainty in discrimination [5].

Current edge detection work shows some degree of success [5], but in general require user interaction. Visual interpretation plays an important part in correlating edges in different scales as well as in making decisions on true vs. false edges. Lack of rigorous means of performing multiscale edge analysis as well as parsing have seriously limited the capability of these methods to achieve automated segmentation. Most current work in segmenting anatomic objects in the brain center around wavelet transforms [5] and model-guided methods.

Because of fewer objects and better contrast in hand X-ray images, we use digitized hand radiographs in our studies. Several investigators are currently developing algorithms to segment these images either automatically or semi-automatically [5, 5, 5, 5, 5, 5]. Feature extraction was performed automatically for patients between the ages of 3 and 11 years using computer vision software developed by Pietka et. al. [5, 5]. For this age group, feature identification was about 90% in agreement with radiologists' assessments. In patients over 11 years of age, we manually measured bone features using simple caliper tools available from a 2048 x 2560 high resolution PACS workstation [5]. Manual methods were used due to difficulties of the current algorithm in reliably estimating bone margins in regions that demonstrate epiphyseal-metaphyseal fusion (i.e. in the phylange inter-joint spaces).

To accommodate different types of images with selected segmentation software in a system, we need to provide guidelines on the types of information to be extracted from the images and the structures to organize them so that they can be effectively retrieved. The types of information extracted from the images for spatial query answering are:

1. object contour

   Object contours are stored as bit maps in the system. A number of spatial characteristics on the object such as area, volume, circumference, bounding box, and 3D-volume rendering [5] can be derived from the contour.

2. spatial features

   Spatial features such as type, shape, area, volume, diameter, length, and circumference describe the spatial characteristics of an image object. For example, to measure the circumference on a cross section of a macroadenoma, the contour showing its boundary is detected as shown in Figure 4. The number of pixels of the contour is counted and the circumference of the macroadenoma is obtained.

3. spatial relations

   Spatial relations between a pair of spatial objects include:

   - orthogonal relations, which describe directional relationships between objects (i.e. `East`, `South`, `SouthEast`, etc.), and
   - containment relations, which describe the relative position and the locations of contact between a pair of objects (i.e. `Invades`, `Contains`, etc.).

   Spatial relations among objects such as containment can be determined through computation with object contours [5, 5]. This derived information is used for answering symbolic spatial queries efficiently. Object contours are also essential in answering direct spatial queries since they provide detailed object boundary information at the pixel level.

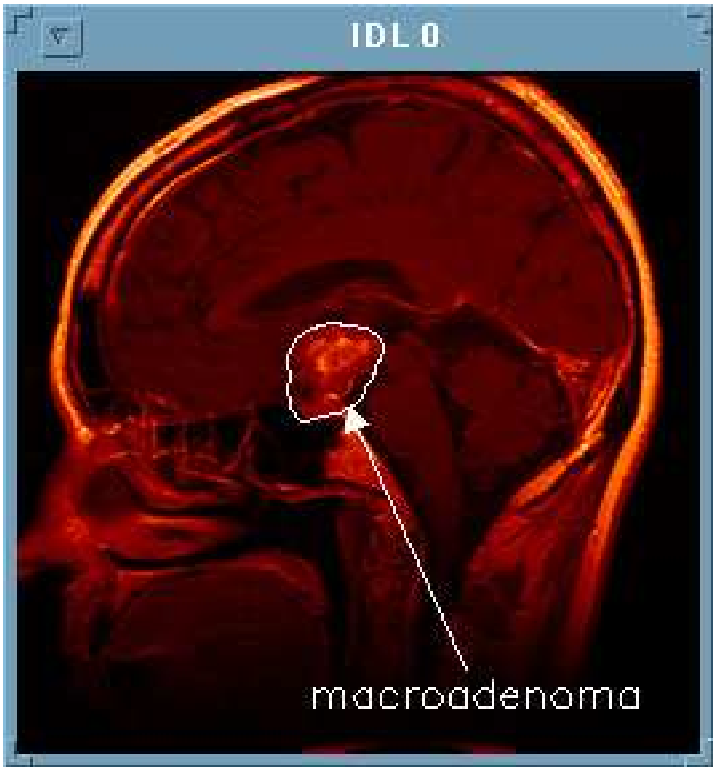

Fig. 4: A sagittal image on a cross section of the brain. The boundary of a macroadenoma (pointed by an arrow) is outlined.

To support spatial queries which directly manipulate image pixels and determine infrequently-referenced spatial relations, we need the pixel-oriented object boundary information provided by object contours. Although this information requires a large amount of storage, it is essential to answer these queries. Not storing object contours has the following problems:

- Using an approximation of object contours such as the bounding box cannot provide accurate spatial query answers.
- Performing image segmentation on-the-fly is very time consuming. For example, extracting features on an X-ray image of the hand takes four minutes on a Sun Sparc II [5, 5]. It is infeasible to scan through all the images and extract features during query execution. Further, certain image segmentation applications require human expert guidance and interpretation.

Therefore, storing high-level semantic spatial information in the database enables us to efficiently answer symbolic spatial queries. Further, such symbolic spatial information and object contours can also improve the processing efficiency of direct spatial queries.

In our system, related images and object contours are stored in stacks [5] based on temporal or spatial correlation among images.

2.2.2. Stacked Image Model

Our underlying data structure for images is the dynamic stacked image model [5, 5]. A stack consists of two-dimensional variables or images registered to the same coordinate system. It employs a gridded data representation scheme as opposed to a topological or polygonal subdivision scheme, providing a natural way to store a set of related images based on time and image development. Shown in Figure 5 is a set of cross-sectional saggital magnetic resonance images stored in the database using the stacked image model. Thus, dynamic image stacks capture not only the structure of multiple-image sequences, but also any behavior and operations that apply to them such as superimposition, translation, rotation, subimage extraction, and others. Geometric operators such as distances among points, lines, and regions are also supported. Note that there are functional mappings from the image objects on the upper level to the corresponding points and segments in the stacked data model on the lower level. The spatial representation of an object in a given space is modeled by a function that maps the object to one or more points in the image space stored in a stacked data structure.

Coregistration is a challenging process that is assumed to be done by each image specialty (e.g. brain imaging, etc.) before loading an image stack and applying various operations that require coregistration.

One image stack usually contains many 2D images of the same modality, such as MRI, CT, etc. Such stacks of 2D images comprise the database provided to specialized software used for 3D rendering and visualization.

2.2.3. Temporal Evolutionary Data Model (TEDM)

We have developed a temporal evolutionary data model and are able to address images by content. TEDM [5] extends the traditional object-oriented data model by introducing a new set of evolutionary and temporal constructs to represent the relationships among different objects. They include:

1. evolutionary object constructs for evolution, fusion, and fission, and
2. temporal relation object constructs that represent temporal relationships between peer objects and between an object and its super- or aggregated type.

One of the contributions of our research is in the modeling of temporal inheritance, which deals with how time-dependent characteristics of a supertype are inherited by its subtypes. The general rule is that an object may only inherit characteristics from other objects which exist in its own space-time domain.

With our temporal evolutionary constructs, we can model the fusion process of the epiphysis and the unfused tabular bone at the metacarpal level (Figure 6). Thumb metacarpal evolutionary stages B through G inherit object characteristics of the metacarpal epiphysis and unfused tabular bone since these stages exist before time T_{fu}. Stages H and I inherit object characteristics of the fused tabular bone since these exist after T_{fu}.

After capturing the evolutionary semantics in the data model for objects of interest, evolutionary queries such as EX1: *"Retrieve an image sequence of a patient demonstrating the fusion of the thumb metacarpal metaphysis with the thumb metacarpal epiphysis. Visualize the sequence in a movie loop."* can be solved by searching the various objects along the path representing the evolution of the thumb metacarpal and retrieving the images demonstrating the progression of fusion. We have implemented this data model and the query language to display the evolutionary processes.

2.2.4. Spatial Evolutionary Data Model (SEDM)

Image objects possess two kinds of spatial properties: their absolute spatial characteristics with respect to a fixed reference frame and their relative spatial characteristics with respect to other objects. Spatial relationships among objects include three components:

1. *spatial relationship:* objects are positioned, described, and modeled spatially using an orthogonal coordinate system in three dimensions.
2. *containment relationships:* an object is either fully contained, partially contained, or not contained within another object.
3. *bordering relationships:* an object may either touch, invade, connect with, or be separated from another object at one or more contact locations.

Fig. 5: Stacked image organization: a set of cross-sectional saggital magnetic resonance images (MRIs), as retrieved by a PICQUERY+ query (also shown). The query returns a list of MRIs for patients with brain tumors. An MRI can then be selected for viewing, as seen in the figure (the *contains* relation operator shown in this query indicates a substring search, not the spatial containment that is mentioned elsewhere in this paper).

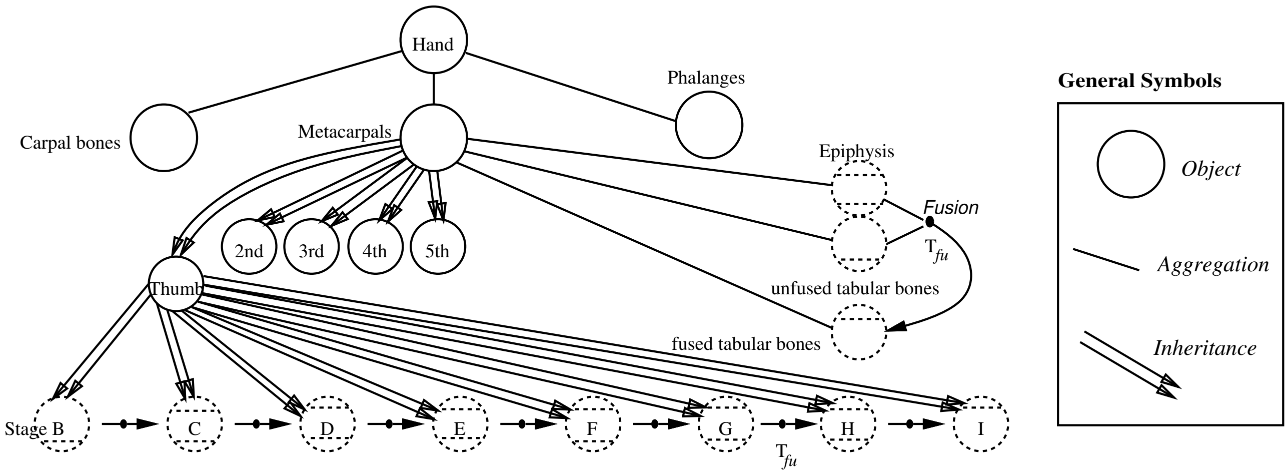

Fig. 6: Modeling the skeletal development of the hand's metacarpal bone. The evolutionary stages progress from stage B to I as defined by [5].

As shown in Figure 3, the indundibular process splits into a nerohypophysis and a neural stalk as the pituitary gland develops. A microadenoma (a tumor smaller than 10 mm) contained within the pituitary gland may evolve into a macroadenoma (a tumor 10 mm or larger). Once evolved, it may invade the sphenoid sinus and/or press against the optic chiasm.

In our approach, the temporal, evolutionary, and spatial features extracted from the images are classified and captured in the data model and stored in tables. Spatial queries can be answered by searching the appropriate tables and selecting the instances that match the query constraints [5]. Using the feature-based, set-oriented approach is superior to the existing time consuming and often inefficient techniques which operate at the image level [5, 5] or the point set level [5] to determine the spatial relationships among objects. For example, query EX3: *"Retrieve all image cases demonstrating the invasion of an adenoma into the sphenoid sinus in pre-adolescent patients."* can be solved by searching an invasion relationship table between the adenoma and the sphenoid sinus and selecting the instances showing the invasion for pre-adolescent patients (see Figure 5). More complex spatial evolutionary queries such as EX4: *"Retrieve image cases demonstrating a pituitary gland microadenoma which evolved into a macroadenoma with suprasellar extension pressing against the optic chiasm."* can be answered by searching a pituitary gland – microadenoma containment relationship table, then following the evolutionary path that leads to a macroadenoma and selecting the instances from the outside contact relationship table between a macroadenoma and the optic chiasm.

Current indexing methods apply to only one aspect of the information embedded in the images. For example, spatial data structures such as R-trees index on the spatial characteristics of the image object; time indices [5] focus only on the temporal aspect of the data; hierarchical data structures such as B-trees index only on attribute values. In our approach, spatial, temporal, and evolutionary features are all extracted from the images into a data model. We therefore have a solid foundation on which to investigate a logical indexing scheme that can utilize spatial, temporal, and evolutionary information to retrieve images and speed up spatial evolutionary query processing [5].

Our index incorporates evolution information along with spatial characteristics. "Nearby" objects, both logically and spatially related, will be clustered together at the evolution points in the database.

2.3. Cooperative Query Answering Layer (CQAL)

This layer of the architecture incorporates the database's ability to reason about its underlying data. A database that is knowledgeable about its underlying data provides the end-user with powerful querying constructs such as conceptual, approximate, neighborhood, and associative queries (see Figure 8a).

Traditional query processing accepts precisely specified queries, provides only exact answers, requires users to fully understand the problem domain and the database schema, and returns limited or even null information if the exact answer is not available. To remedy such a restriction, extending the classical notion of query answering to *cooperative query answering* (CQA) has been explored [5, 5]. CQA possesses two aspects: *neighborhood query answering* (NQA), which provides neighborhood or generalized information relevant to the original query within a certain semantic distance to the desired answer, and *associative query answering* (AQA), which provides information conceptually related to but not explicitly asked for by the query.

An NQA process relaxes a query scope to enlarge the search range, or relaxes an answer scope to include additional information. Enlarging and shrinking a query scope can be accomplished by viewing the queried objects at different conceptual levels, since an object representation has wider coverage at a higher level, and, inversely, narrower coverage at a lower level. Although linking different-level object representations can be made in terms of explicit rules [5, 5, 5], such linking lacks a systematic organization to guide the query transformation process. To remedy this problem, we use the notion of a *type abstraction hierarchy* (TAH) [5] for providing an efficient and organized framework for NQA.

Type abstraction hierarchies override conventional subsumption-based type hierarchies by introducing abstract values in supertype objects. In the usual semantic network or object-oriented database approach, information is dealt with only at two general levels: the meta-level and the instance level. However, in the type abstraction hierarchy, multilevel object instance representations can be introduced.

For example, in the type abstraction in Figure 7, the *third, fourth, and fifth metacarpal are stubby* can be relaxed to *the third and fourth metacarpals are stubby or thin*. As a result, we can handle objects with different abstractions. However, in order to achieve efficient NQA, we need to control the relaxation process; this relaxation control is discussed in the next section.

2.3.1. Relaxation Control

When a query has a null or empty answer, some of its attribute values can be relaxed to provide a wider search scope. As a result, approximate answers are derived. However, without control over relaxation, we may generate more approximations than the user can handle. For example, consider EX2, the case in which the user asks for images of Turner's syndrome, and assume that the exact answer is not available. There are many ways to relax the query, as shown in the TAH in Figure 7: for example, we can generalize from "stubby 3rd, 4th, 5th metacarpal with hand/wrist bone normal," to "positive metacarpal sign with hand/wrist bone normal," to "cases with abnormal hand/wrist and metacarpal bones." We can also generalize "Korean American" to "Asian," then specialize to "Chinese," "Japanese," etc., or generalize "12 years old" to "pre-adolescent" then specialize "pre-adolescent" to other appropriate ages. These generalization/specialization operations effectively broaden the search scope of the original query.

Suppose we know, in addition, that the user is usually interested in cases in which the hand/wrist bone is normal, does not wish to relax the patient's age, and desires limited relaxation on the patient's ethnicity. With this extra information, we can limit the query relaxation to a smaller scope by trimming the type abstraction hierarchies — thus improving the system's performance and the relevance of the results, as shown in Figure 7.

The full relaxation control policy depends on many factors: user profile, query context, attribute relaxation order (e.g. relax age first, then ethnic group), unrelaxable attributes, unacceptable conditions, preference order, relaxation level, and desired number of answers. Relaxation control will also involve rules for combining these factors over multiple attributes. The *Relaxation Manager (RM)* applies these rules based on certain metrics or objectives (minimizing search time or semantic distance, for example) to restrict the search for approximate answers.

Figure 8b depicts the data flow within the relaxation manager and explanation system. When a query is presented to the cooperative database system, the system first relaxes any explicit cooperative operators in the query. The modified query is then presented to the underlying database

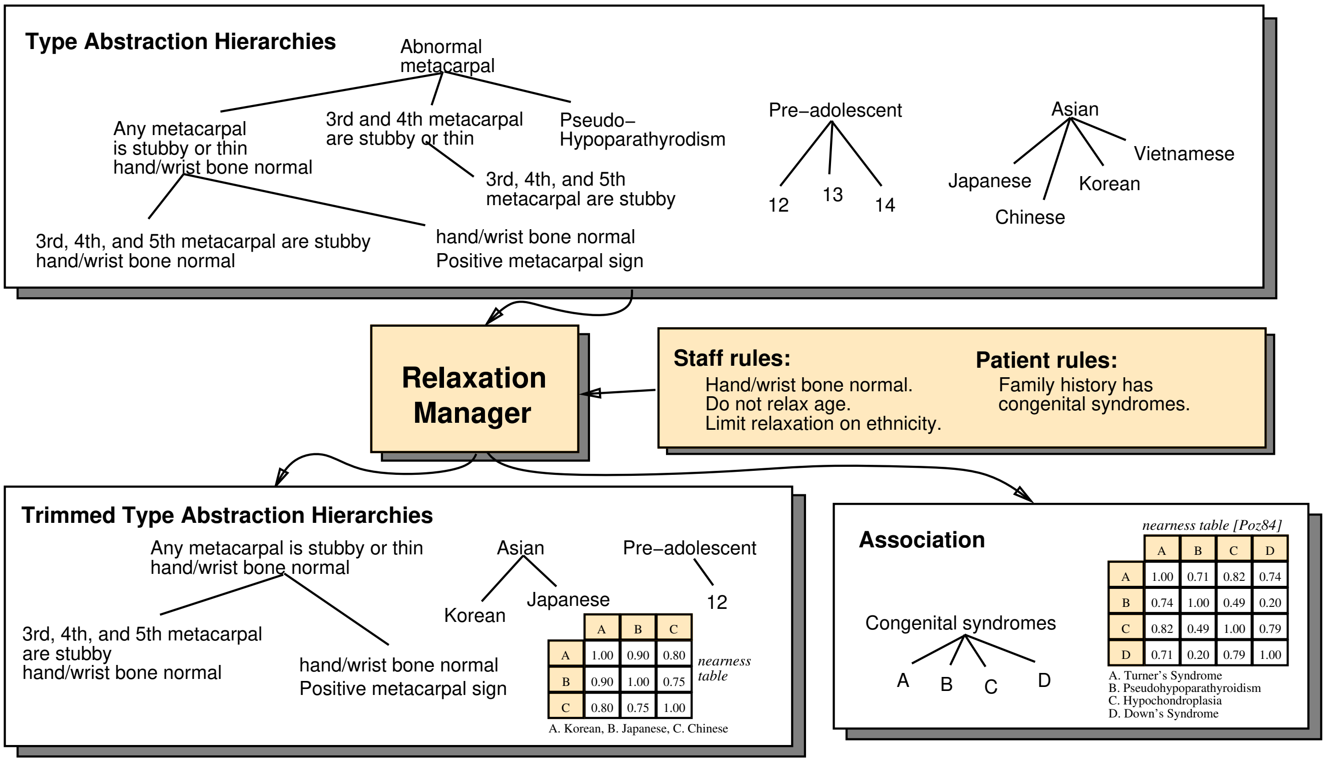

Fig. 7: Relaxation control and subject association via the Relaxation Manager.

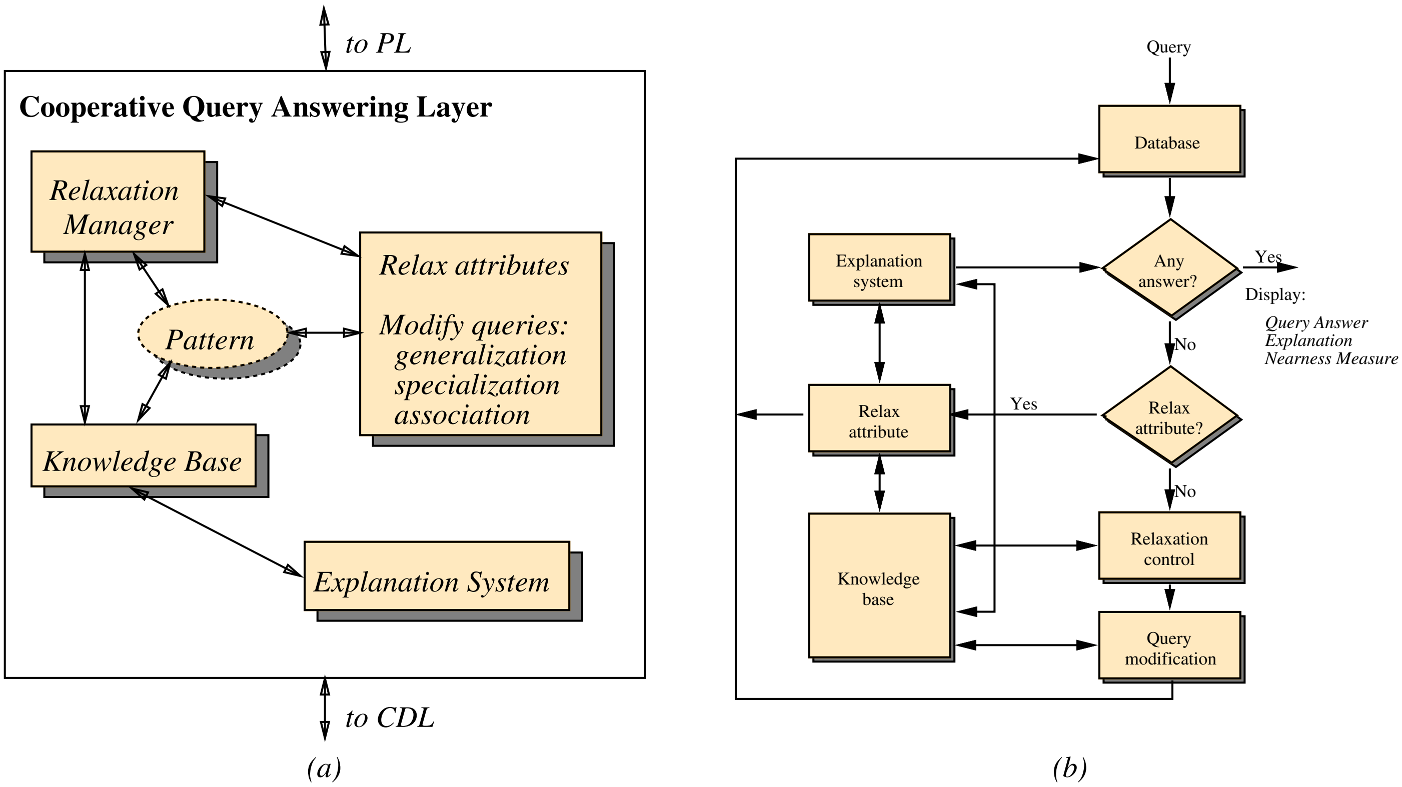

Fig. 8: (a) The CQAL. (b) Flowchart for cooperative queries.

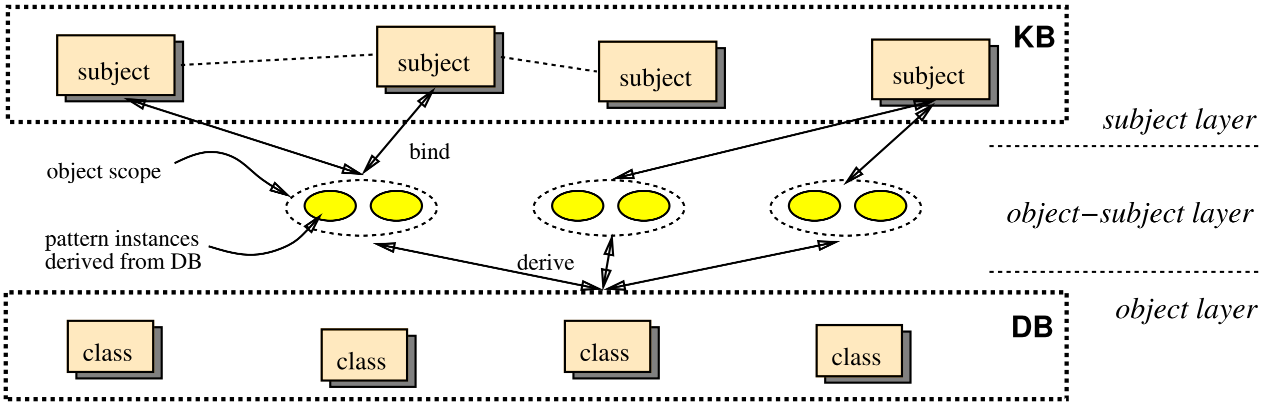

Fig. 9: The DB-pattern-KB architecture.

system for execution. If no answers are returned, then the cooperative query system, under the direction of the Relaxation Manager, relaxes the queries by query modification using the trimmed type abstraction hierarchies. The relaxed query is executed, and, if there is no answer due to over-trimmed TAHs, the Relaxation Manager will deactivate certain relaxation rules, restoring part of a trimmed TAH to broaden the search scope until an answer is found.

When an approximate answer is returned, the user may wish to be presented with an explanation of how the answer was derived or with an annotated relaxation path. A context-based semantic nearness will be provided, allowing the system to rank the approximate answers (in order of nearness) against the specified query. A menu-based graphical user interface (GUI) is provided to control over explanations of the relaxation choices.

2.3.2. Associative Query Answering (AQA)

AQA finds information that is relevant to a user's query even if it was not explicitly requested. Thus, AQA needs to identify the dependencies and relationships between objects distributed in multiple classes or even multiple databases. Since objects have different relationships in different problem domains, it is necessary to focus on the localized contexts in which the objects participate. Domain knowledge is then used to interpret the relationships between these objects. As an object can have multiple super-types, there exist different views of type abstraction for different problem contexts. Therefore, TAHs express relationships at the context level.

Linking the Database and the Knowledge Base

To facilitate knowledge-based AQA, we introduce a *DB-pattern-KB* architecture as the framework for supporting both *static grouping* and *dynamic linkage* of data and knowledge [5]. Based on such a framework, data grouping will be class-oriented and knowledge grouping will be subject-oriented. Classes and subjects are coupled via virtual *pattern instances* containing the objects participating in the subjects. These objects are derived from the data classes.

The framework has three layers, as shown in Figure 9: *object layer*, *object-subject layer*, and *subject layer*. The subject layer and the object-subject layer reside in the CQAL (see Figure 8a) and can be described as follows:

- *object layer: object-oriented database.* This layer contains the usual object-oriented database with a reasonable granularity of object classification. In general, knowledge about particular subjects is not maintained at this layer.
- *object-subject layer:* At this layer, the notions of *pattern* and *pattern-instance* are introduced to provide an *interface between the database and the knowledge base.* A pattern is just a query on the objects that qualifies a set of objects satisfying its conditions. For example, based on the object schema

macroadenoma(*size, shape, volume, cell type, grade, secretion, extra sellar extension*),

a query condition such as *size* $\geq$ *10 mm, shape = round with rough contour, and extra sellar extension = positive* can be used to define a pattern on a macroadenoma (a tumor with size greater than 10 mm).

The objects satisfying a pattern condition *match the pattern*. Deriving object instances based on a known pattern is a *deduction* process, and finding patterns that match a given object is a *dynamic classification* process. In fact, patterns are just *views* rather than actual classes of the database, which are used to specify the participants of certain subjects, and are not instantiated until needed. Therefore, patterns provide a flexible interface between data and knowledge that allows data and knowledge to be maintained and updated independently of each other. Like views, patterns are specified by the user but generated by the system.

- *subject layer: subject-oriented knowledge base.* This layer is characterized by subject-oriented knowledge organization. Rules and facts in a knowledge base system are classified under subjects. A subject provides a localized scope for organizing the knowledge related to a particular problem domain. Knowledge about a subject describes cross-class inter-object relationships which are not generally specified in any single data class.

To support system scalibility and maintainability, we organize the database and the knowledge base independently and link them dynamically. To facilitate such a link it is necessary to specify the objects that participate in a subject. Further, to provide a high-level and abstract specification on these objects, we specify the patterns that contain them instead of specifying the objects themselves.

Knowledge-Based Associative Query Answering

Subject knowledge represents behavioral associations among the participating objects. To determine such associations, it is necessary for a subject to identify the participating objects and for an object to identify the subjects that object is involved in. Based on our DB-pattern-KB framework, determining the participating objects of a subject involves two steps: identifying the participating patterns based on their object scope specification, and determining the database objects matching these patterns. Inversely, identifying the subjects involving a given object also includes two steps: dynamically classifying the object into one or multiple patterns, and identifying the subjects whose object scopes involve these patterns.

We shall now show how we can provide AQA to query EX4: *"Retrieve image cases demonstrating a pituitary gland microadenoma that has evolved into a macroadenoma with suprasellar extension pressing against the optic chiasm."* In this case, the DB contains the objects Sella Turcica, Pituitary Gland and Optic Chiasm. The KB consists of the subjects "Adenoma" and "Vision." The pattern is "macroadenoma with suprasellar extension," as shown in Figure 10. Since we know that an adenoma growing inside the pituitary gland with suprasellar (above the sella turcica) extension pressing against the optic chiasm causes bitemporal hemianopic scotomata and produces an island of visual loss, then in the subject "Adenoma," we summarize this knowledge into a rule, "If an adenoma with suprasellar extension invades the optic chiasm, then the adenoma can impair vision." This rule causes the pattern to *dynamically* link with the subject "Vision." Since different sizes of adenomas can cause different degrees of vision loss, we have the following rules:

- "Microadenoma invading the optic chiasm can cause sieve vision loss."
- "Macroadenoma pressing against the optic chiasm can cause partial vision loss."

The knowledge in the subject "Vision" provides additional information and/or queries for this pattern. As a result, the pattern derives answers from both Sella Turcica and Optic Chiasm. The pieces of additional information — "adenoma with suprasellar extension may impair vision" and

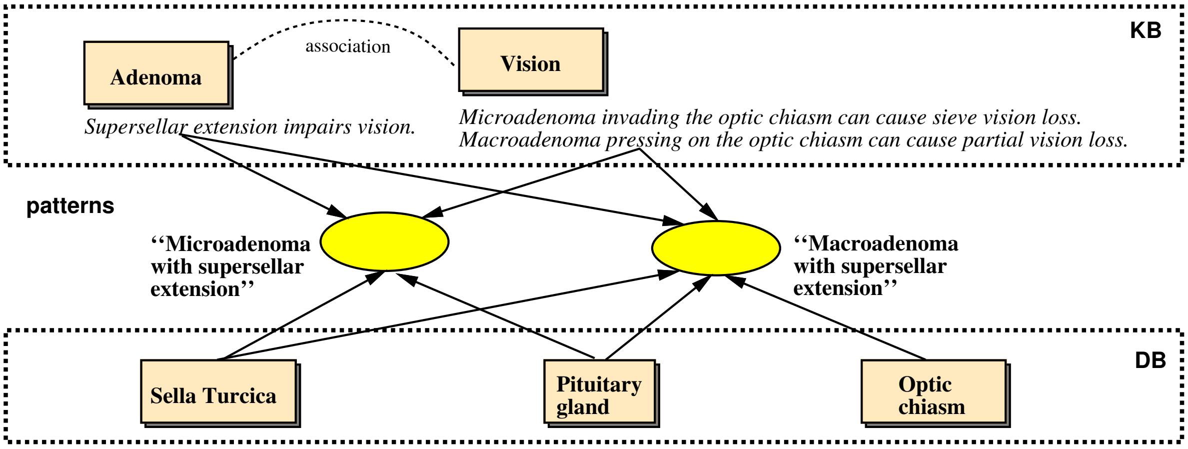

Fig. 10: A DB-pattern-KB approach of associative query answering for query EX4.

"vision test results can provide additional confirmation of evolution from microadenoma to macroadenoma" — are appended to the query. Further, association operations can provide statistics of the risk factor on the treatment of similar tumors in the past, as in EX5. As a result, AQA provides significant diagnostic and clinical value.

We will implement this DB-pattern-KB framework to provide associative query answering in the spatial evolutionary environment. This structure provides static grouping and dynamic linking of KB and DB. Thus, they can be maintained relatively independently of each other. As a result, our methodology can be scaled to large systems.

2.3.3. Knowledge Acquisition and Maintenance

Knowledge Acquisition

Medical knowledge is crucial to the design of our cooperative query answering layer. Two methods are used to incorporate domain knowledge:

1. *Domain experts.* Currently, the knowledge in our system is specified by medical specialists. It is organized using a hybrid object-oriented/knowledge-base entity-relationship data model [5], which provides the mechanisms to represent and integrate the various objects, attributes, generalization and specialization hierarchies, object relationships, association links between different scopes, rules for the definition of fuzzy terms, and nearness correlation coefficients between objects. Future developments will include the organization of data and knowledge into different object, object-subject, and subject layers as described in Section 2.3.2. For example, this layered approach will allow the linking of two subjects related to skeletal abnormalities: the first one is a disease classification based on the underlying cause, and the second one is based on the disease symptoms and measurable features. Thus, a disease such as Turner's Syndrome can be relaxed to include other diseases which have similar symptoms but possibly have very different underlying disease mechanisms, or alternatively, to include other diseases which pathologically originate from similar causations but which have different symptoms. The PICQUERY+ language graphically allows the user to visualize the knowledge base (including type abstraction hierarchies, rules, etc.) to assist in the formulation of the query (see Section 2.4.1).

2. *Knowledge discovery from databases*

   (a) Data instances with non-numerical attribute values Certain knowledge may not be easily specified by experts. Our hybrid data model will incorporate machine learning methods that employ inductive learning to discover semantic intra-object and inter-object knowledge implicit in the database. We follow the approach for knowledge induction as presented in [5] as a supplementary way to acquire knowledge. The notion of patterns as defined in Section 2.3.2 is used to study the relationships between correlated patterns. For example, the following disease–symptom patterns can be automatically deduced:

$$\textit{Disease} = \textit{Turner's Syndrome} \Rightarrow \textit{Metacarpal Sign} = \textit{Positive},\ \alpha = 0.87$$

The inferential correlation α is a probabilistic coefficient deduced for each relationship. It is a measure of "how common" the inference is based on the total population space, and the usefulness/accuracy of the discovered knowledge rule. Rules can be further generalized to produce more abstract knowledge. The following inferential relationships:

$$\textit{Ethnicity} = \textit{``Chinese'' or Ethnicity} = \textit{``Japanese''} \Rightarrow \textit{Height} < 1.75m,\ \alpha = 0.70$$
$$\textit{Ethnicity} = \textit{``Swedish'' or Ethnicity} = \textit{``Dutch''} \Rightarrow \textit{Height} \geq 1.75m,\ \alpha = 0.67$$

can be represented in a more abstract fashion as:

$$\textit{Ethnicity} = \textit{``Asian''} \Rightarrow \textit{Height is short},\ \alpha = 0.70$$
$$\textit{Ethnicity} = \textit{``Northern European''} \Rightarrow \textit{Height is tall},\ \alpha = 0.67$$

Summary knowledge such as "In general Northern Europeans are taller than Asians" can then be derived.

We have used this method to acquire knowledge and generate TAH from database instances with non-numerical values.

(b) Data instances with numerical attribute values A conceptual clustering method is used for discovering high-level concepts of numerical attribute values from databases. The method considers both frequency and value distributions of data, and thus is able to discover relevant concepts from numerical attributes [5]. The discovered knowledge can be used for representing data semantically and constructing the type abstraction hierarchy for providing approximate answers when exact ones are not available. The knowledge discovery partitions the data set of one or more attributes into clusters that minimize relaxation errors (a nearness measure for clustering numerical values). For a more detailed discussion on this subject, the interested reader should refer to [5].

Knowledge Organization and Maintenance

Organizing knowledge based on subjects allows knowledge inheritance among inter-subject relationships. Rules specified in a super-subject are applicable to its sub-subjects. The hierarchical knowledge organization provides a structured way to maintain domain knowledge, thus allowing the system to scale up. We introduce three types of subject knowledge hierarchies as shown in Figure 11:

- *generalization:* Knowledge about a subject may be generalized to include a broader subject scope or specialized to narrow the subject scope. For example, the subject Hypoplasia includes rules about the non-specific hand disease hypoplasia, while its sub-subject Turner's Syndrome inherits rules from it and includes additional features fully characterizing Turner's Syndrome.

- *subject composition:* Subject composition provides a way to specify more complex subjects in terms of simpler ones. For example, let subject "Structural_Abnormalities" be a subject covering all the knowledge about hand malformations. A subject about abnormal bone orientation or a subject about the absence of bones can be viewed as a component of the subject "Structural_Abnormalities."

- *subject abstraction:* In a subject-abstraction hierarchy, a rule in the super-subject has a more abstract representation than that in the sub-subject. This feature allows specifying subjects at higher levels and in wider ranges than we described in subject generalization.

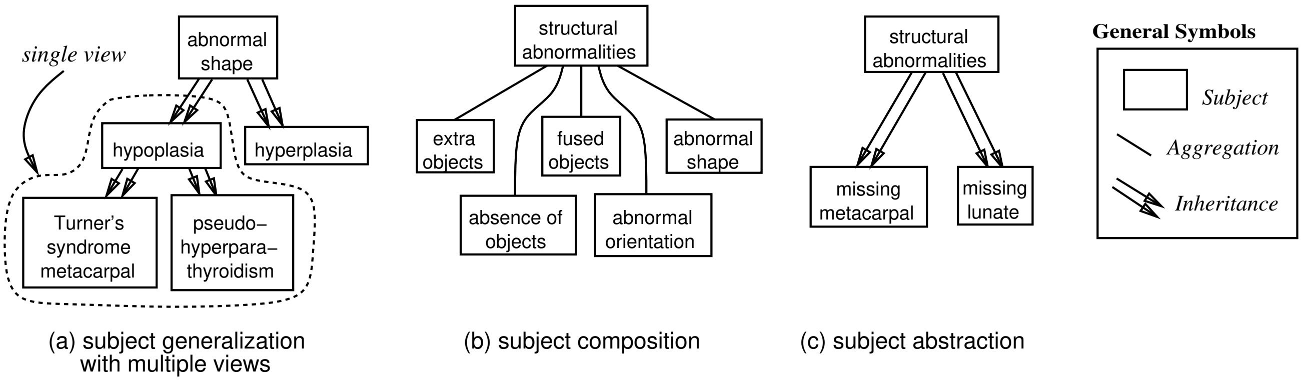

Fig. 11: Subject generalization, composition, and abstraction hierarchies.

Knowledge Editor

Various knowledge structures are used as "defaults" by the query relaxation facility. The user can interact with the facility and edit its default knowledge. The editable knowledge structures are: the TAHs to be used in query relaxation and the default values for unspecified control parameters, The knowledge editor provides an interactive interface for the user to examine and modify the relaxation control parameters.

Editing TAHs The automatically generated knowledge structures (TAHs) from database may not be perfect for all applications. Therefore, a user may want to change the knowledge bases to better suit his applications. A knowledge editor allows the user to browse and edit *TAHs*. The editor provides the following editing operations.

delete: to improve search efficiency, any irrelevant part of a hierarchy for a given application should be deleted.

move: if a TAH is improperly formed for its application, then a domain expert can adjust it by moving sub-hierarchies and attaching them to the appropriate places.

add: if a TAH does not provide enough details for the particular application, then additional information can be added to the hierarchy.

Editing Relaxation Control Parameters In addition to editing TAHs, the knowledge editor allows the user to browse and edit the relaxation control parameters to better suit his applications. The relaxation control parameters include attribute weights for *TAHs*, the relaxation range for *approximately-equal* operator, the default radius for *near-to*, the number of returned tuples for *similar-to*, etc. By using the knowledge editor, the user can make the query relaxation more context and user sensitive.

2.4. Presentation Layer (PL)

The primary interface between a user or application and the other layers in the system is the presentation layer (PL), which provides, at the user's level, the query constructs and capabilities that are eventually filtered to the rest of the system. In addition to the standard querying facilities that can be found in today's commercial databases, our system adds the ability to query image data from their spatial, evolutionary, and temporal content. We integrate this new querying functionality into a single query language.

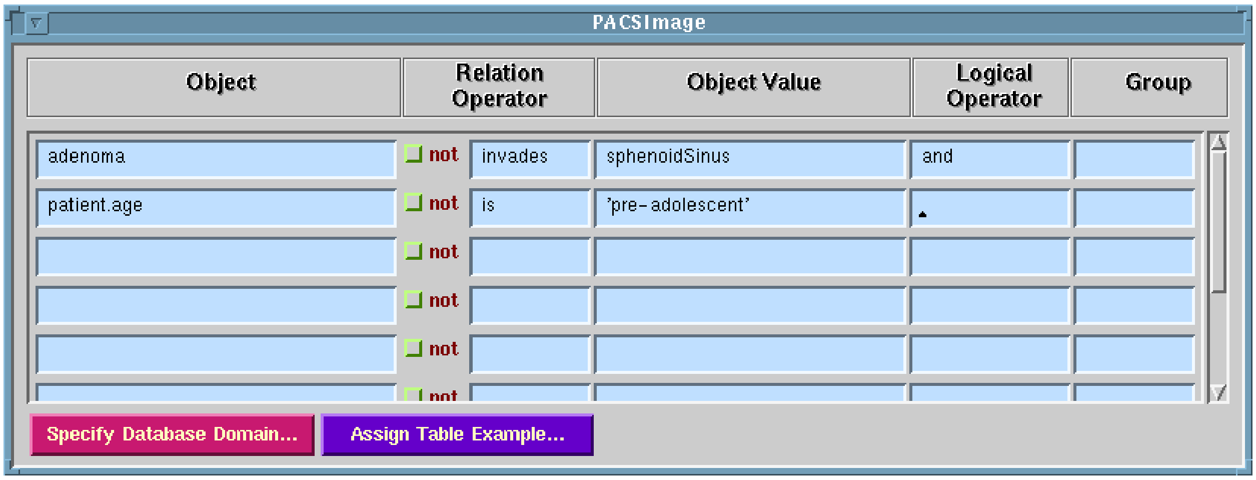

Fig. 12: A sample PICQUERY+ table expressing the query EX3: "Retrieve all image cases demonstrating the invasion of an adenoma into the sphenoid sinus in pre-adolescent patients".

2.4.1. Language Requirements for the PL

To fulfill the querying needs of KMeD, we require a front-end language that is capable of expressing the many new constructs introduced in this document while maintaining clarity and ease-of-use for its users. We augmented the original PICQUERY language [5] with significant advances and now refer to it as PICQUERY+. PICQUERY+ offers the following major additional features over PICQUERY [5]:

1. specification of the data domain space from a multimedia database federation,
2. visualization of underlying data models, knowledge-based hierarchies, and domain rules,
3. query processing of temporal and object evolutionary events,
4. the ability to perform queries using imprecise ("fuzzy") descriptors and relational correlators,
5. the ability to specify alphanumeric, image processing, and visualization operations on any query result.

A subset of the entire PICQUERY+ design has been implemented including the stacked-image data model and the temporal evolutionary data model (TEDM). Object-oriented and graphical interface technology served as an implementation base for our PICQUERY+ subset. We are in the process of developing the spatial querying facilities provided by SEDM to our prototype system, as well as language/user interface constructs to support full relaxation control and associative query answering.

PICQUERY+ Examples

Figure 5 shows the PICQUERY+ example for the query *"Retrieve patient images in the PACS database domain with brain tumors."* The query returns a list of cross-sectional saggital magnetic resonance images (MRI) for such patients. The user can then select the patient MRI thus displaying the image stack in a window of the PICQUERY+ screen. The PACS, in day-to-day operational use, does not support this type of access by image content features.

Figure 12 presents another sample PICQUERY+ table which expresses EX3:

EX3 *Retrieve all image cases demonstrating the invasion of an adenoma into the sphenoid sinus in pre-adolescent patients.*

These queries illustrate PICQUERY+'s object orientation, visualization capabilities, and proposed spatial constructs. Note how objects can be referred to by name in the query; the list of all valid objects is stored in the KMeD global directory/dictionary which contains the database's schema. A user who is unfamiliar with the schema may, at any time, view it in graphical form, including any knowledge or type abstraction hierarchies that accompany it to facilitate cooperative query answering. In addition, objects need not be objects in the database per se — they can be derived objects which are generated "on the fly" through predefined methods. For example, the object `patientAge` may not be available as a stored attribute. Instead, it is generated dynamically using the patient's birthday and the image's date stamp.

Once the requested objects have been retrieved — in this case it is the set of all images which contain the requested adenoma (Figure 5) — they are sent to a visualization subsystem that can present the results in various ways. Since the result of this query is a set of unrelated images from different patients, simple `imageDisplay` is sufficient for visualization. Images which are related (through evolution in time or position in space, for example) can be visualized differently, perhaps using a `movieLoop` or `threeDimensionalView` instead.

Menu-Icon PICQUERY+ and Data Model Visualization

The PICQUERY+ language uses a tabular user interface. However, a more menu- and application domain-oriented interface which employs a hierarchical window system and graphical representation of objects (e.g. icons, etc.) has been developed for our medical domain, and specifically for each application domain. Thus, PICQUERY+ will include features for tailoring such menu–icon shells to specific domains with specific knowledge bases. Buttons are included at the bottom of a high-level query window to perform such functions as schema display, rule display, or query confirmation.

Our architecture also connects KMeD image stacks to and commercial 3D rendering and visualization products to provide 3D volume visualization, 2D sectioning of 3D images, etc.

3. TESTBED FACILITIES AND IMPLEMENTATION

A testbed that provides the KMeD functionality while maintaining a high degree of extensibility has been developed at UCLA. Our primary software development platform is VisualWorks, from ParcPlace Systems; database management operations are handled primarily by GemStone from Servio Corporation, with other databases participating as described in the autonomous database layer. The fully object-oriented development base makes for powerful software constructs and promotes portability across platforms. This portability is one of our objectives; we should be able to distribute the developed software to other interested research parties as long as their platform can run the VisualWorks and GemStone environments.

Our testbed currently consists of the following aspects and support modules:

- *Image and object database support.* We have built a fully functional image class hierarchy in GemStone; in addition, we use VisualWorks and other third party software such as ImageMagick and IDL to provide basic visualization functionality, including image display, scaling, and rendering. If the GemStone image is a dynamic stack, a movie loop option has been made available to display each image in the stack. The object-oriented design of our system makes it easy to add or to connect to further visualization functionality as the needs of the project dictate.

  3D rendering and visualization can be a time-consuming process for higher image resolutions. Specialized software and hardware are used to reduce the processing time. However, queries requiring the system to assemble volumes of data for the first time result in long wait periods.

- *Language support.* The data manipulation languages for each layer in the database architecture have been defined, and compilers/translators for subsets of these designs have been implemented. We have developed the methods and classes necessary for implementing temporal and evolutionary queries, and we are extending them to cover spatial predicates.

- *Integration.* Images from the UCLA Department of Radiological Sciences are stored as regular Unix files under a specific predefined PACS format (PACS is the official repository of digitized images in the UCLA Center for Health Science). The PACS format includes not only the actual image data, but standard alphanumeric information such as the patient's name, ID, the date of the examination, and others. We have extended our installation of GemStone so that it is capable of reading PACS files directly, without further intervention by the user, then integrating the PACS data right into the KMeD object-oriented database. Users of our testbed can thus access PACS image files from any location in the network that has access to our GemStone server.

- *Cooperative query support.* Together with the radiology group and medical experts, we have been gathering the knowledge necessary to provide cooperative query answering capabilities. This knowledge is stored in GemStone for representing type abstraction hierarchies. The object-orientation of GemStone makes it a very natural implementation base for a type abstraction hierarchy. We have implemented the relaxation manager with fuller functionality. We are currently in the process of implementing the explanation system, knowledge editors, and facilities for associative query answering [5].

- *Hardware/software platform.* We use a Unix platform with Sun SPARCstations to develop our system. One of these stations acts as a server for GemStone, our primary database platform, and is also used for actual software and user interface development. GemStone allows transparent concurrent access to the database server from remote nodes, and provides bridges to other commercial database systems such as the relational Sybase (this allows access to heterogeneous databases).

4. CONCLUSION

Our effort embodies a unique and powerful conceptualization of heterogeneous scientific databases. Not only do we solve the merger and management of observed data from multiple sources with spatial and evolutionary constructs, but our system strives to aid the scientist in *extracting the science from the data.*

We have developed a semantic, spatial, evolutionary data model to allow us to retrieve images by features and contents. We have also developed a knowledge-based cooperative query facility with control policies to answer imprecise queries and also provide additional relevant information even if it is not asked for in the query. Further, we have specified the requirements for a query language with spatial, temporal, evolutionary and cooperative constructs, and have implemented a subset of that language. A KMeD testbed and prototype has been constructed to validate these concepts. Our future directions will provide: more flexible user input options such as point-and-click query input or freehand sketches and figures, more effective visualization techniques to present results to the user, and incorporation of sound and video data into our system.

The joint research between investigators from the UCLA Computer Science Department and the Center for Health Sciences assures that the prototype system and experiments used are of direct interest to medical research. The new methodology enables access to the vast storehouse of images by content and features with spatial and evolutionary constructs, association with relevant subjects, and relaxation control for efficient cooperative query processing. Our capabilities enhance the image databases by characterizing disease behavior, radiographic features, and effects of diagnosis techniques. The results of this research should be extensible to other large-scale image and multimedia database applications.

Acknowledgements — This work is supported by the National Science Foundation Scientific Database Initiative, Grant IRI9116849. In addition, the authors would like to thank John David N. Dionisio, Ion Tim Ieong, Chih-Cheng Hsu, Timothy Plattner, Christina Chu, Zuad Oropeza, Hyeonjoon Shim, Hing Ming Chan, and Roger Barker of the UCLA KMeD project for their contributions to KMeD's development. Also, our appreciation to our colleague collaborators: Drs. Claudine Breant, Robert Lufkin, Gary Duckwiler, Daniel Valentino, Fernando Vinuela, and Ted Hall at the UCLA Center for Health Sciences, and Drs. Bernie Huang and Katherine Andriole at the UCSF Medical Center, whose requirements and involvement established the direction and success of KMeD.

REFERENCES

[1] E. Angel. *Computer Graphics*. Addison-Wesley (1990).

[2] M. Bart, H. Romeny, L. Florack, J. Koenderink, and M. Viergever. Scale space: Its natural operators and differential invariants. In *Information Processing in Medical Imaging, Proceedings of the 12th International Conference, IPMI*, page 239 (1991).

[3] M. Brummer, A. Van Est, and W. Menhardt. The accuracy of volume measurements from MR imaging data. In *Proceedings of the 8th Annual Meeting SMRM*, page 610, Amsterdam (1989).

[4] A. F. Cardenas. Heterogeneous distributed database management: the HD-DBMS. *Proceedings of the IEEE*, **75**(5) (1987).

[5] A. F. Cardenas, I. T. Ieong, R. Barker, R. K. Taira, and C. M. Breant. The knowledge-based object-oriented PICQUERY+ language. *IEEE Transactions on Knowledge and Data Engineering* (1993).

[6] A. F. Cardenas, R. K. Taira, W. W. Chu, and C. M. Breant. Integration and interoperability of a multimedia medical distributed database system. *Bulletin of the Technical Committee on Data Engineering* (1993).

[7] N. S. Chang and K. S. Fu. Query-by-pictorial-example. *IEEE Transactions on Software Engineering*, **SE-6**(6):519–524 (1980).

[8] M. Chock, A. F. Cardenas, and A. Klinger. Database structure and manipulation capabilities of a picture database management system (PICDMS). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-6**(4) (1985).

[9] W. W. Chu and Q. Chen. Pattern-based data and knowledge integration for intelligent query answering. *Journal of Integrated Computer-Aided Engineering*, **1**(4):279–293 (1994).

[10] W. W. Chu and Q. Chen. A structured approach for cooperative query answering. *IEEE Transactions on Knowledge and Data Engineering* (1994).

[11] W. W. Chu, Q. Chen, and R. Lee. Cooperative query answering via type abstraction hierarchy. In S.M. Deen, editor, *Cooperating Knowledge Based Systems, 1990*. Elsevier Science Publishing Co., Inc. (1991).

[12] W. W. Chu, I. T. Ieong, and R. K. Taira. A semantic modeling approach for image retrieval by content. *Journal of VLDB* (1994).

[13] W. W. Chu, I. T. Ieong, R. K. Taira, and C. M. Breant. A temporal evolutionary object-oriented data model and its query language for medical image management. In *Proceedings of the 18th International Conference on Very Large Databases* (1992).

[14] W. W. Chu, R. C. Lee, and Q. Chen. Using type inference and induced rules to provide intensional answers. In *Proceedings of the 7th International Conference on Data Engineering* (1991).

[15] W. W. Chu, M. A. Merzbacher, and L. Berkovich. The design and implementation of CoBase. In *Proceedings of ACM SIGMOD* (1993).

[16] Wesley W. Chu and Kuorong Chiang. Abstraction of high level concepts from numerical values in databases. In *Proceedings of AAAI Workshop on Knowledge Discovery in Databses* (1994).

[17] F. Cuppens and R. Demolombe. Cooperative answering: A methodology to provide intelligent access to databases. In *Proceedings of the 2nd International Conference on Expert Database Systems* (1989).

[18] F. Cuppens and R. Demolombe. How to recognize interesting topics to provide cooperative query answering. *Information Systems*, **14**(2) (1989).

[19] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure and App. Math*, **XLI**:909 (1988).

[20] R. Elmasri, G. Wuu, and Y. Kim. The time index: An access structure for temporal data. In *Proceedings of the 16th International Conference on Very Large Databases* (1990).

[21] G. Fouque, W.W. Chu, and H. Yau. A case-based reasoning approach for associative query answering. In *Proceedings of the Eighth International Symposium on Methodologies for Intelligent Systems*, Charlotte, NC (1994).

[22] T. Gasterland, P. Godfrey, and J. Minker. An overview of cooperative answering. *Journal of Intelligent Information Systems*, **1**(2):123–157 (1992).

[23] A. Gupta, T. Weymouth, and R. Jain. Semantic queries with pictures: the VIMSYS model. In *Proceedings of the 17th International Conference on Very Large Databases* (1991).

[24] H. K. Huang and R. K. Taira. Infrastructure design of a picture archiving and communication system. *American Journal of Roentgenology*, **158** (1992).

[25] P. Impekoven. Expert system for determining skeletal age and adult height. *Radiology*, **181**(P):153, Abstract (1991).

[26] T. Joseph and A. F. Cardenas. PICQUERY: A high level query language for pictorial database management. *IEEE Transactions on Software Engineering*, Special issue on image data management (1988).

[27] T. S. Levit. Model-based predication of phalanx radiograph boundaries, Presented at the SPIE Conference on Medical Imaging, Newport Beach, CA (1993).

[28] T. S. Levit and M. W. Hedgcock. Model-based analysis of hand radiographs. In *Proceedings of the SPIE: Medical Imaging III*, volume 1093, pp. 563–568 (1989).

[29] T. S. Levit, M. W. Hedgcock, J. W. Dye, and S. E. Johnson. Automation of arthritis measures in hand radiographs. In *Proceedings of the SPIE: Medical Imaging IV*, volume 1233, pp. 473–480 (1990).

[30] G. Manos, A. Y. Cairns, I. W. Ricketts, and D. Sinclair. Automatic segmentation of hand-wrist radiographs. *Image and Vision Computing*, **11**(2):100–111 (1993).

[31] M. Mantyla. *An Introduction to Solid Modeling.* Computer Science Press (1988).

[32] D. Marr and E. Hildreth. Theory of edge detection. In *Proc. R. Soc.*, page 187 (1980).

[33] Matthew Merzbacher and Wesley W. Chu. Pattern-based clustering for database attribute values. In *Proceedings of AAAI Workshop on Knowledge Discovery in Databases* (1993).

[34] D. J. Michael and A. C. Nelson. HANDX: A model-based system for automatic segmentation of bones from digital hand radiographs. *IEEE Transactions on Medical Imaging*, **8**(1):64–69 (1989).

[35] E. Pietka, L. Kaabi, M. L. Kuo, and H. K. Huang. Feature extraction in carpal bone analysis. *IEEE Computer Graphics and Applications* (1991).

[36] E. Pietka, M. F. McNitt-Gray, M. L. Kuo, and H. K. Huang. Computer assisted phalangeal analysis in skeletal age assessment. *IEEE Transactions on Medical Imaging*, **10**(4):616–620 (1991).

[37] N. Roussopoulos, C. Faloutsos, and T. Sellis. An efficient pictorial database system for PSQL. *IEEE Transactions on Software Engineering*, **14**(5) (1988).

[38] B. K. Stewart, D. R. Aberle, M. I. Boechat, Z. Barbaric, R. K. Taira, J. W. Sayre, and S. J. Dywer III. Clinical utilization of grayscale workstations. *IEEE Engineering in Medicine and Biology*, **11**(1):86–102 (1993).

[39] J. M. Tanner, R. H. Whitehouse, W. A. Marshall, M. J. R. Healy, and H. Goldstein. *Assessment of Skeletal Maturity and Prediction of Adult Height (TW2 Method).* Academic Press, London (1975).

[40] D. J. Valentino, J. C. Mazziotta, and H. K. Huang. Volume rendering of multi-modal images: Application to MRI and PET images of the human brain. *IEEE Transactions on Medical Imaging* (1991).