A Temporal Evolutionary Object-Oriented Data Model and Its Query Language for Medical Image Management*

W. W. Chu and I. T. Ieong

Department of Computer Science
University of California
Los Angeles, CA 90024

R. K. Taira and C. M. Breant

Department of Radiological Sciences
University of California
Los Angeles, CA 90024

Abstract – A temporal evolutionary object-oriented data model (TEDM) for medical image data is presented in this paper. The images (e.g. X-rays, CT scans, MR scans, etc.) and their features are represented as objects in the data model. A high-level declarative temporal evolutionary query language (TEQL) integrated with the data model is proposed which provides the user with the capabilities of querying on the temporal evolutionary contents of the medical images. Our proposed model and language constructs can be applied to application domains that exemplify the evolutionary transformations of objects.

1 Introduction

Few fields in medicine have changed as rapidly in the past decade as radiology. Several new digital imaging techniques of the human body have emerged including computed tomography(CT), magnetic resonance (MR), ultrasonography (US), projectional computed radiography (CR), digital subtraction angiography (DSA), positron emission tomography (PET), and nuclear medicine (NM) imaging. These medical imaging systems have revolutionized the means by which images are acquired, providing views of anatomical cross-sections and physiological state, as well as reducing patient radiation dose and examination trauma.

This revolution in the acquisition of radiological information has not yet brought about a parallel revolution in the intelligent management, visualization, integration, or knowledge extraction from data produced by these digital imaging system. The generation of large volumes of digital image data has caused a crisis in managing radiological studies[7]. A typical 700-bed teaching hospital conducts about 200,000 radiological studies per year generating over a million images. Current efforts have been focused on the development of a radiological picture archiving and communication system (PACS) infrastructure to provide efficient communication, archival, retrieval, and display of radiological images[9][26]. However, study retrieval in these systems is based on traditional artificial keys, such as patient hospital identification number. This clearly limits the querying power for radiology research which is concerned with optimizing the visualization of disease processes and mass screening a large number of cases and correlating radiographic features, patient history, and intermodality temporal comparison images to disease processes. Few radiologists have the time or energy to search through patient images and their support information to collect sufficient image samples for their research. There is a need in radiology research and education to manage the image data stored in the PACS such that image retrieval can be conducted based on image *contents* and *semantics*.

In this research, we examine image data derived from diagnostic medical procedures which provide snapshots of human growth and pathological states. Structures in the human body are not static and often change their characteristics and/or existence over time. For example, a database used to build a model of human skeletal maturity involves characterizing the growth patterns of structures in the hand. At birth, only a limited number of bones are present. As we mature, microscopic growth centers evolve into new bones. In the wrist area, the eight carpal bones normally appear in roughly four stages. In the fingers, cartilage, the precursor to hard bones, begins to undergo chemical transformations. The epiphysis, a structure between the phalange bones of children, begins to fuse with the tubular phalange bones at a certain point in skeletal maturation. Exceptional genetic conditions can cause some bones to undergo a fission process, splitting into multiple bones. This illustrates that medical data are not only temporal, but also evolutionary in nature. Current database

*This research is supported by NSF contract IRI 9116849.

**Proceedings of the 18th VLDB Conference
Vancouver, British Columbia, Canada 1992**

models provide the modeling of schema evolution[2][3] but lack the power of handling temporal evolutionary objects. In our approach, we represent the images (e.g. X-rays, CT scans, MR scans, etc.) and the image features related to a structure in the patient as objects. These features are extracted either automatically[17][18][19] or manually. We shall propose a temporal evolutionary data model (TEDM) that provides a powerful abstraction mechanism for modeling the temporally evolving data seen in real life (medical images). The model uses the traditional object-oriented data model as its departure basis. We enhanced the modeling power of the object-oriented data model by introducing a new set of novel constructs to describe the evolutionary behavior of objects. The new model utilizes three types of object constructs: traditional object constructs, temporal relation object constructs, and evolutionary object constructs. We also propose a high-level declarative temporal evolutionary query language (TEQL) based on the data model that provides the user with the capabilities of querying on the *temporal evolutionary contents* of the medical images.

2 Modeling Temporal Information

Time is an integral piece of information in the description of an object and/or process in a constantly evolving real world. Our model represents time as a set of discrete equidistant points. Events in general occur over intervals of time. Intervals are an effective abstraction of temporal representation. Points or instants of time can be easily represented as closed intervals with identical lower and upper bounds.

Let T be a countably infinite set of totally ordered discrete equidistant time points, where T is denoted as $T = \{t_0, t_1, t_2, ..., t_{now}, ...\}$. We define a time interval, denoted by $[t_i, t_j]$, to be a set of consecutive equidistant time instants. The distance between two consecutive time instances, t_i and t_{i+1}, represents the granularity of the application, and can be any suitable time unit defined by the user. A time sequence is a series of time points $\{t'_1, t'_2, t'_3, ..., t'_{n-1}, t'_n\}$ where $t'_1 < t'_2 < ... < t'_n$ may or may not be uniformly consecutive.

In temporal databases, it is customary to include a number of different time dimensions. The most common kinds of time are: valid time, transaction time, record time, event time, future time, and user-defined time. We adopt the taxonomy of time terminology given in [4] for valid time and record time and in [21] for event time. Valid time consists of the start time and the end time used to delimit the records of an object history. Start time is the time when a new object becomes valid in the database. The end time is the time when an object becomes invalid. The record time is when the information of an object is recorded in the database. The event time is when the event about an object actually occurs.

In our approach, each object is stamped with either a valid time interval, <start time, end time>, or an event time. The reasoning behind making a difference between them can be explained by the following examples. A patient may stay in a hospital for a month from January 15 to February 15, 1992 which can be conveniently represented by a time interval. However, a medical image represents a snapshot of the human body in a particular growth stage. It is valid only at the time point when the image is taken and therefore, it is more appropriate to describe it by an event time. The distinction between these two time concepts also has implementation implications.

Objects in our model are defined based upon more primitive objects. They are either versionable or non-versionable. Versionable objects are the collections of their non-versionable counterparts tagged by a time sequence or a sequence of time intervals in ascending order. For example, non-temporal information such as the social security number of a person can be defined using a non-versionable character string. The person's height can be defined using a versionable real number tagged by a time sequence to record the growth history of that person. The employment history of a person can be defined by using a versionable composite object to record the employer, the location, and salary tagged by a sequence of time intervals. The motivation behind the differentiation is to provide a supporting mechanism for storing and manipulating each kind of object. In our model, each instance of an object type is an object uniquely identified by its object identity.

3 The Temporal Evolutionary Data Model(TEDM)

The temporal evolutionary object-oriented data model extends the traditional object constructs by introducing a new set of novel constructs to describe the evolutionary behavior of objects that are essential for modeling medical images. In addition to the traditional object constructs, TEDM also uses evolutionary and temporal constructs to represent the relationships among different objects as described in the following:

1. Traditional object constructs[10][29](see Figure 1a):
   (a) Aggregation: An object is composed of several constituent objects that form an "Is-part-of" hierarchy. Figure 2a shows a model of the growth of a hand in various developmental stages. The model is based on the TW2 method[27] using the object constructs in our model for bone age assessment. The hand is composed of Carpal bones, Phalanges, and Metacarpals. The Metacarpals are composed of either an epiphysis and a unfused tabular bone, or a fused tabular bone as shown in Figure 2a.
   (b) Generalization/Specialization: Relates an object type to more generic ones and forms an "Is-a" hierarchy. The more generic ones are called supertypes while the more specialized ones are

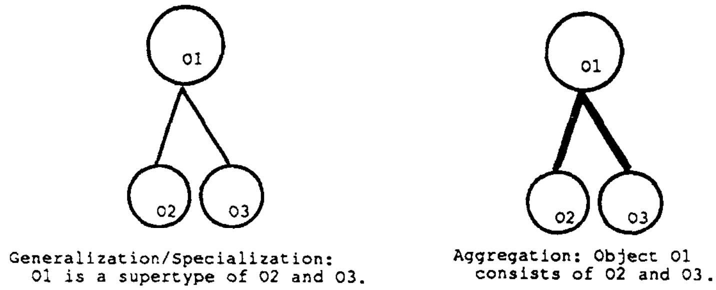

Figure 1a Object type hierarchy

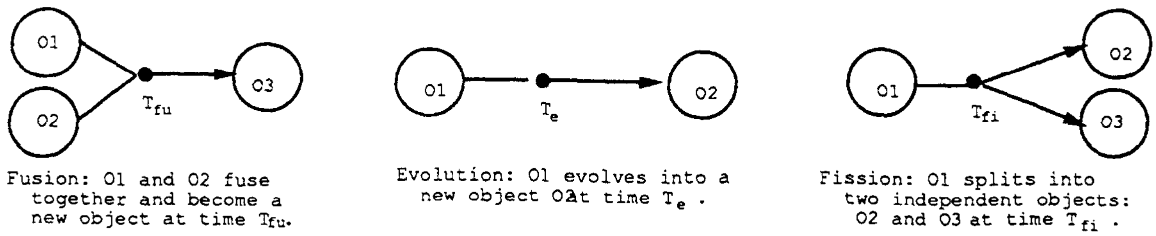

Figure 1b Evolutionary object constructs

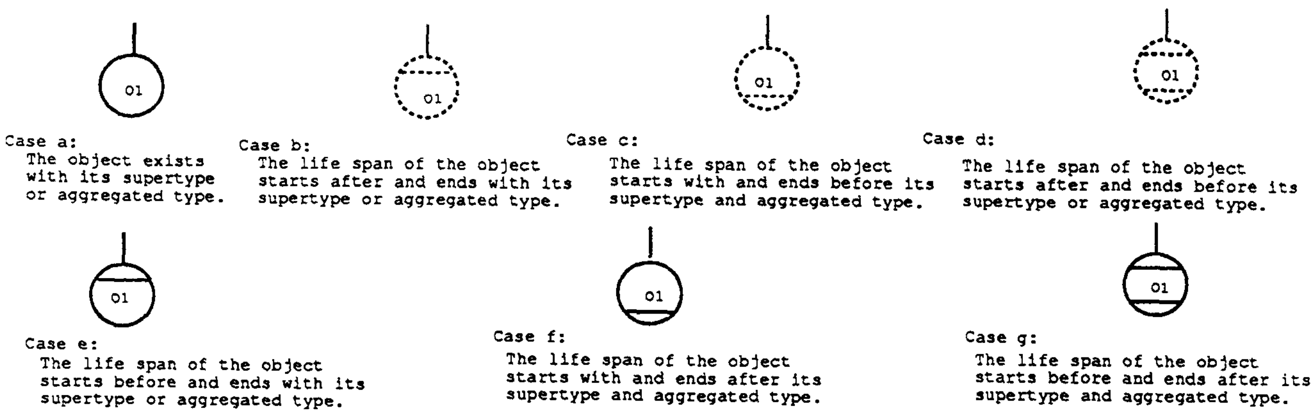

Figure 1c The temporal relation constructs between an object and its supertype or aggregated type.

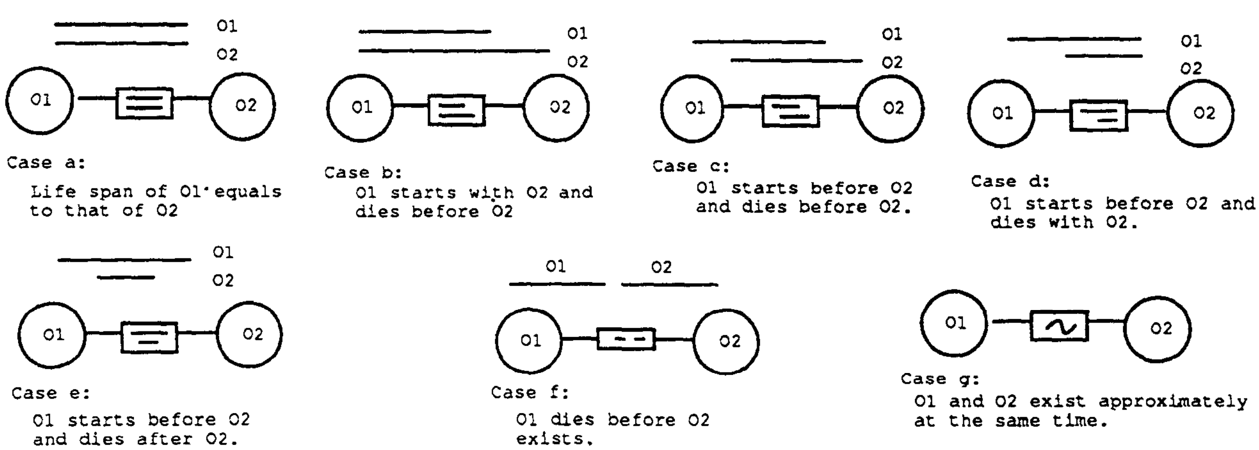

Figure 1d The temporal relation constructs between peer object types

called subtypes. The subtypes inherit the characteristics from their supertypes. One contribution 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. For example, if the thumb metacarpal bone is in its early stage of development (termed TW2 stage C), then it inherits all the characteristics of both its constituent objects, the epiphysis and the unfused tabular bone. When the thumb metacarpal bone fully matures, it inherits the characteristics of a different object type, the fused tabular bone(Figure 2a). The general rule is that an object may only inherit characteristics from other objects which exist in its own space-time domain. The aggregation and generalization/specialization form a type lattice which is a directed acyclic graph and the conflicts of multiple inheritance are resolved in the conventional way[30].

2. Evolutionary object constructs(see Figure 1b):

   (a) Evolution: The characteristics of an object may evolve with time. As shown in Figure 2a, the growth of the thumb metacarpal matures through eight stages. Figure 2b shows the images of how thumb metacarpal evolves. We make a distinction between versions of an object and the evolution of an object. For example, at age 3, the thumb metacarpal is in growth stage B when the epiphysis is a small growth center that is barely visible. If the child takes an X-ray, it will be considered an object in object type stage B (Here we treat all the objects in an uniform way, including the image objects). At age 12, in growth stage F, the epiphysis goes through several phases of transformation, evolving into a hard bone with roughly the same diameter as the tabular bone. If the child takes an X-ray at age 12, it will be considered an object of object type stage F. The X-ray in stage F is considered to have evolved from that in stage B. However, if the child takes multiple X-rays at age 3, they are considered versions of the objects in stage B, represented by a versionable object.

   (b) Fusion: An object may fuse with different objects to form a new object with different characteristics than either of the constituent objects. As modeled in Figure 2a, the epiphysis and the tabular bone of the metacarpals will fuse into the fused tabular bone at T_{fu}. Figure 2b shows the fusion of the thumb metacarpal epiphysis and the tabular bone into the fused tabular bone from stage G to H. Figure 3 illustrates explicitly how the fusion between epiphysis and the unfused tabular bone into the fused tabular bone is temporally inherited by the various stages. For example, stages E to G inherit the characteristics of the epiphysis and the unfused tabular bone while stages H and I inherit those of the fused tabular bone. Since the aggregated object types are evolving from stage to stage, the corresponding constituent object types are also evolving. The epiphysis in stage F evolves into the epiphysis in stage G. The epiphysis and tabular bone in stage G are fused into a fused tabular bone in stage H. The relationships between the constituent types do not have to be explicitly specified as in Figure 3. They can be inherited. As in Figure 2a, they are abstracted and placed into a higher level. The methods defined on the objects will recursively search the supertypes to verify the associations. Notice the attributes of each constituent object type are also temporally inherited. The black squares inside an object type represent the object instances of that object type.

   (c) Fission: At some time, an object may split into two or more independent objects. For example, a cell may split into two cells. The Department of EECS may split into two independent departments: Department of EE and Department of CS. An adult woman may become a pregnant woman. She may deliver an infant and becomes an adult (unpregnant) woman. The fission among the constituent types of an aggregated object type can be inherited in a similar manner as the fusion.

   Let us define an *evolutionary net* to be a directed graph where the nodes are the object types connected by the evolutionary constructs. Therefore, with the above evolutionary object constructs, evolutionary nets can be constructed by database designers in the data model. An *evolutionary sequence* in an evolutionary net is a directed graph where the nodes are object instances of the object types connected by the evolutionary constructs. An evolutionary net (solid line) and the evolutionary sequences (dotted line) representing the evolution of the thumb metacarpal are shown in Figure 3. The directions of evolutionary transformations in the evolutionary sequences are dictated by the object types. The evolutionary nets in our data model can be represented in abstracted forms as in Figure 2a and are inherited by the subtypes.

3. Temporal relation object constructs:

   The weakness of current object oriented data models is the difficulty in specifying the integrity constraints of objects. Integrity constraints are not considered specific concepts of the model. They are defined in a uniform way as any procedure describing the behavior of an object. Other data models represent integrity constraints as declarative assertions on the data structure. The constraints are specified either as: a complementary information of binary arcs, new predicates in Morse semantic networks[4], or constraint equations[15] that provide a concise declarative language for expressing semantic constraints and require con-

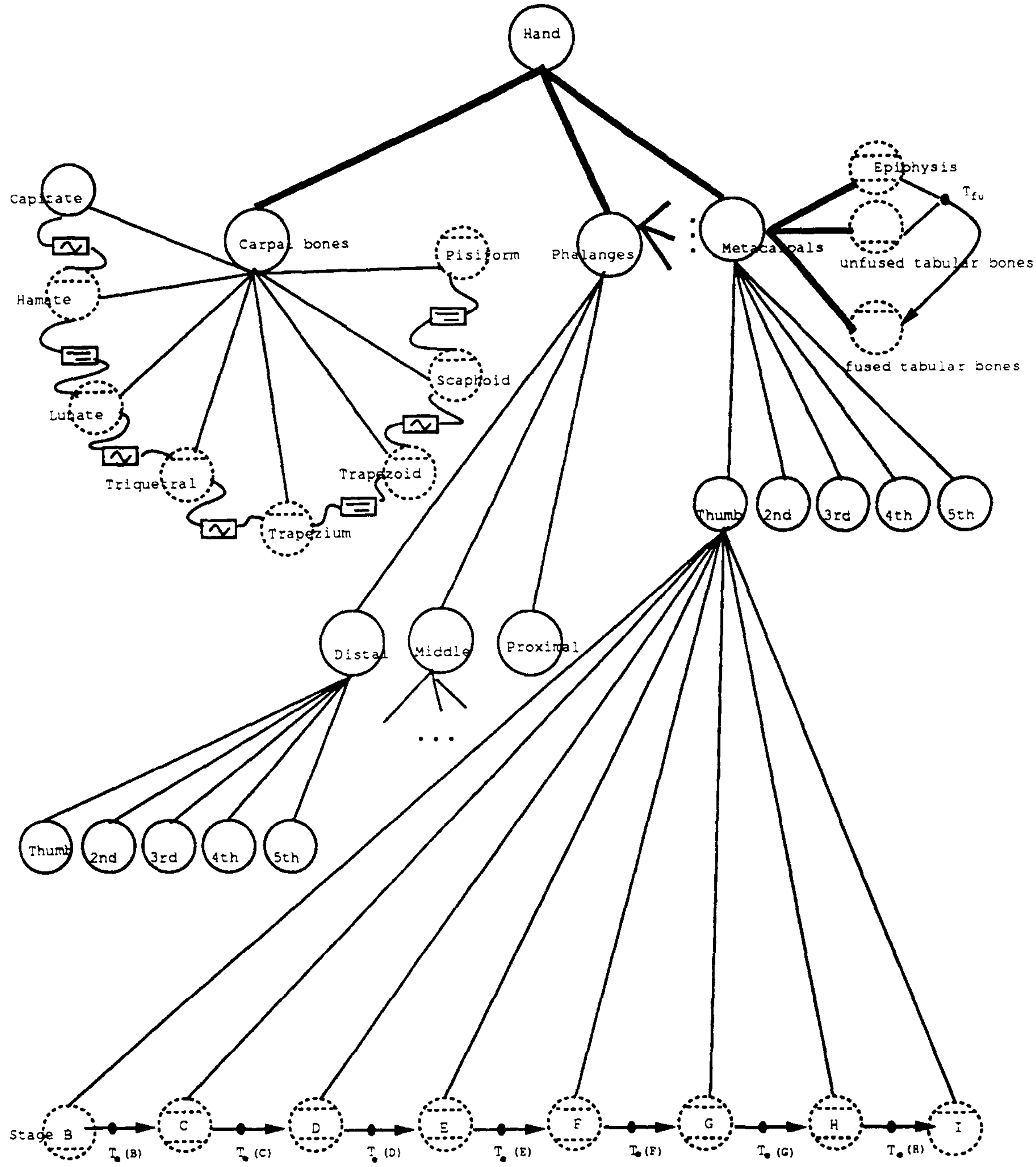

Figure 2a **Modeling the growth of a hand for the TW2 method (Partial representation).**

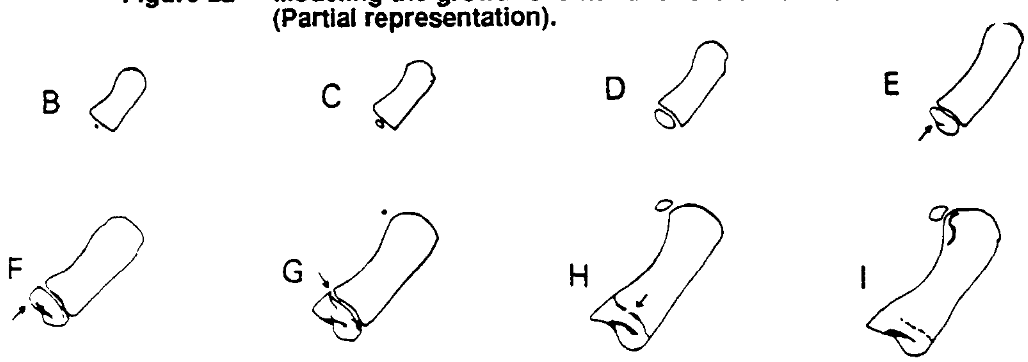

Figure 2b **The growth evolution and fusion of the thumb metacarpal for the TW2 method. (Courtesy of Academic Press)**

sistency among several relations. Some models[5] use first order logic formulas whose variables refer to the content of the semantic database to express general integrity constraints.

Constraints in general can be classified into: static constraints, dynamic constraints, and temporal constraints[21]. In our model, constraints are used to maintain data consistency and represent inter-object relationships. We focus our attention on the modeling of inter-object temporal constraints with the temporal relation object constructs. They are modeled in the following way:

(a) The constructs that represent the temporal relationships between an object and its supertype or its aggregated type are shown in Figure 1c. From Cases a to d, the life span of the supertype or aggregated type contains the life span of its subtypes or constituent types. From Cases f to g, the life span of an object type is allowed to be larger than that of its supertype or aggregated type. The life span of an object that exists with its supertype or aggregated type does not need to be specified (Case a). It is inherited or defined by its supertype or aggregated type. Therefore, only the differing components of the time domain in the remaining six cases need to be specified. This reduces the number of redundant constraints. Furthermore, these temporal relationships define how the characteristics of the supertypes are temporally inherited.

(b) The object constructs show the temporal relationships between the life spans of peer objects at the same level in the hierarchy as shown in Figure 1d.

To illustrate the use of the temporal relation object constructs, we apply them to model the eight carpal bones that appear in four stages as we grow from infant to adult (Figure 2a). There are eight carpal bones in the wrist area. The Capitate appears first. It exists as long as the supertype Carpal bones exists and is modeled by the Case a temporal relation construct in Figure 1c. The remaining seven bones appear later with respect to their supertype Carpal bones and are modeled by the Case c temporal relation construct in Figure 1c. The Hamate appears right after the Capitate. They both belong to the first stage. Since the difference in timing in general does not have significant impact in bone age assessment, and is considered to appear at about the same time with the Capitate. The Lunate and the Triquetral also appear at about the same time with the Trapezium in the second stage. The Trapezoid and the Scaphoid appear at about the same time in the third stage. They are modeled by the temporal relation construct Case g in Figure 1d. The Lunate appears significantly later than the Hamate (in the second stage). To distinguish the timing difference of the first stage from the second stage, it is modeled by the construct Case d in Figure 1d. In the same manner, the same construct is used to model the timing difference of the life spans for the Trapezium with the Trapezoid and the Scaphoid with the Pisiform.

4 The Temporal Evolutionary Query Language (TEQL)

Conventional pictorial query languages lack the capabilities to query on the temporal evolutionary nature of medical images[8][11][13][22]. To remedy this, we propose a new query language TEQL integrated with our data model TEDM that operates on the temporal evolutionary domains of the medical images. TEQL contains constructs to specify the temporal and evolutionary conditions in addition to the traditional arithmetic predicates constructs. The temporal operators specify the data at a particular point in time and the evolutionary operators specify the evolutionary object sequences of interest. A query of TEQL is composed of the following seven optional clauses:

```
[CONTEXT a_subdatabase]
[CONSTRUCT a_view]
[WHERE clauses]
[WHICH clauses]
[WHEN clauses]
[SELECT clauses]
[Operations]
```

CONTEXT references the view or subdatabase created by the user. CONSTRUCT creates a view or a subdatabase customized to the interests of the individual users. WHERE clauses describe the selection criteria using traditional arithmetic predicates. WHICH clauses describe the evolutionary processes among object types. WHEN clauses select the appropriate snapshot of the database. SELECT selects the desired data items. Operations specify the required system or user-defined operations on the chosen data such as display, movie_loop, contour, rotate, superimpose, and panning. The full text of the query language on retrieval in BNF is given in the appendix.

4.1 CONSTRUCT

CONSTRUCT is used to create a view customized to the interest of the individual users. The user specifies a desired subdatabase by specifying its intentional pattern and extensional pattern types. CONSTRUCT clauses reflect the hierarchical structure of the underlying data model which is lacking in most current object-oriented query languages[12]. It starts by selecting the root object types. One can then select the desired subtypes. The intra-type conditions enclosed in the brackets following an object type name are optional and expressed in the form of predicates that involve the attributes of that type[25]. The user can also specify what attributes should be retained in each type. The default is to include all the attributes for the corresponding types.

For example, to construct a subdatabase for Caucasian patients of age 14 or older with a positive metacarpal sign

(See Section 5.1 for more detailed discussion) and with a fourth metacarpal bone less than or equal to 15 mm, we have:

```
CONSTRUCT view_x =
  (hand [ethnic_group='Caucasian' AND age>=14]
  (metacarpal [metacarpal_sign='positive' ]
  (fourth_finger[length <= 15 mm ])))
```

We can refer to the above subdatabase by using the following clause:

```
CONTEXT view_x
```

Notice the user does not need to specify the evolutionary relationships between object types in the view. The evolutionary relationships will be verified with the underlying data model during query execution.

4.2 WHERE

The WHERE clauses remove the extensional patterns that do not satisfy certain conditions. Non-temporal inter-type comparison conditions and conventional arithmetic predicates can be specified in the WHERE clauses. Comparison conditions involving aggregation functions such as COUNT and AVG are also allowed. The arithmetic predicates used in our model are similar to those in the relational model.

4.3 WHICH

The WHICH clause describes various evolutionary processes on a set of objects in an evolutionary net. There are three kinds of evolutionary processes: evolution, fusion, and fission. A discussion of each evolutionary process follows:

4.3.1 Evolution

The evolution predicate "$object\_type_1$ EVOLVED_FROM $object\_type_2$" selects all single step evolutionary sequences in which an object instance of $object\_type_2$ evolves into another object instance in $object\_type_1$. For example, applying the single step evolution operator EVOLVED_FROM in the following WHICH clause

```
WHICH epiphysis_stage_F_thumb_metacarpal
      EVOLVED_FROM
      epiphysis_stage_E_thumb_metacarpal
```

on the objects in Figure 3 selects the evolutionary sequences for John and Mary on the evolutionary net from stage E to stage F as shown in Figure 4a.
Let us now discuss the multiple step evolution operators. The evolution condition "$object\_type_1$ EVOLVED+_FROM $object\_type_2$" selects all multiple step evolutionary sequences in which an object instance of starting $object\_type_2$ evolves into another object instance in ending $object\_type_1$ in the evolutionary net. For example, at age 8, Patient Joy took an X-ray, which is an object instance in stage E. She did not take another one until 12 years old, which is an object instance in stage G. Using the EVOLVED+_FROM in the following WHICH clause

```
WHICH epiphysis_stage_G_thumb_metacarpal
      EVOLVED+_FROM
      epiphysis_stage_E_thumb_metacarpal
```

selects the evolutionary sequences for John and Joy on the evolutionary net from stage E to stage G as shown in Figure 4b.
"$object\_type_1$ EVOLVED*_FROM $object\_type_2$" selects all single and multiple step evolutionary sequences in which an object instance evolves into another object instance in the evolutionary net where $object\_type_2$ evolves into $object\_type_1$ as follows:

```
WHICH epiphysis_stage_G_thumb_metacarpal
      EVOLVED*_FROM
      epiphysis_stage_E_thumb_metacarpal
```

selects the sequences for John, Mary, and Joy from stage E to stage G as shown in Figure 4c.

4.3.2 Fusion

The operator FUSED_FROM selects the evolutionary sequences in an evolutionary net where a fusion process of several objects occurs. For example,

```
WHICH
  fused_tabular_bone_stage_H_thumb_metacarpal
  FUSED_FROM
  epiphysis_stage_G_thumb_metacarpal,
  unfused_tabular_bone_stage_G_thumb_metacarpal
```

selects the single step evolutionary sequence for Joy (Figure 5a).
We can define the multiple step fusion operators, FUSED+_FROM and FUSED*_FROM in the same manner as the multiple step evolution operators. For example,

```
WHICH fused_tabular_bone_stage_I_thumb_metacarpal
      FUSED+_FROM
      epiphysis_stage_G_thumb_metacarpal,
      unfused_tabular_bone_stage_G_thumb_metacarpal
```

selects the multiple step evolutionary sequence for John (Figure 5b).
The fusion condition "$object\_type$ FUSED*_FROM $object\_type_1$, $object\_type_2$, ..., $object\_type_k$" selects all the evolutionary sequences with a fusion process in the evolutionary net from $object\_type_1$, $object\_type_2$, ..., and $object\_type_k$ to $object\_type$. For example,

```
WHICH fused_tabular_bone_stage_I_thumb_metacarpal
      FUSED*_FROM
      epiphysis_stage_F_thumb_metacarpal,
      unfused_tabular_bone_stage_F_thumb_metacarpal
```

selects the evolutionary sequences with a fusion for John, Mary, and Joy as shown in Figure 5c.

4.3.3 Fission

Similarly, there are three operators to describe the fission processes: SPLITTED_FROM, SPLITTED+_FROM, and SPLITTED*_FROM that are used to select the evolutionary sequences where an object splits into several objects. The fission condition "$object\_type_1$, $object\_type_2$,

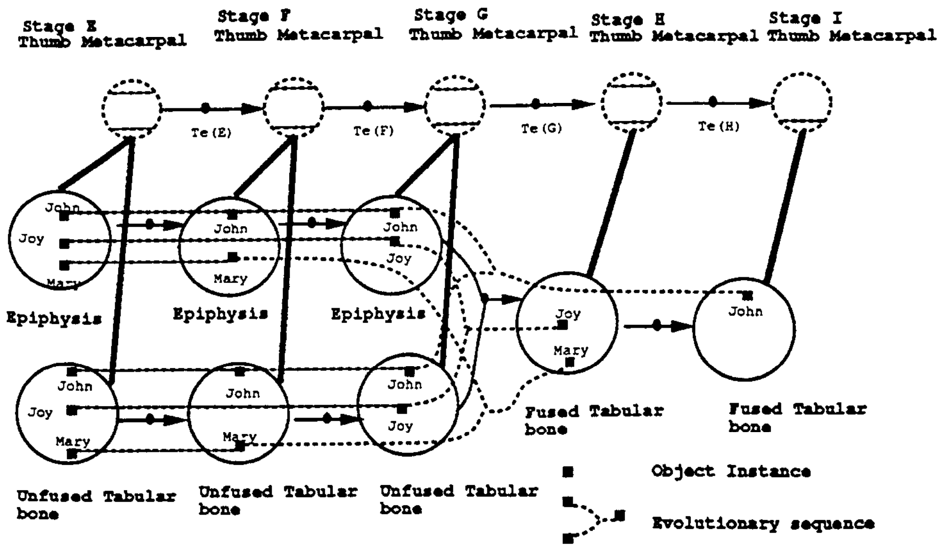

Figure 3 Fusing the epiphysis with the unfused tabular bone into the fused tabular bone from stage E to I for the thumb metacarpal.

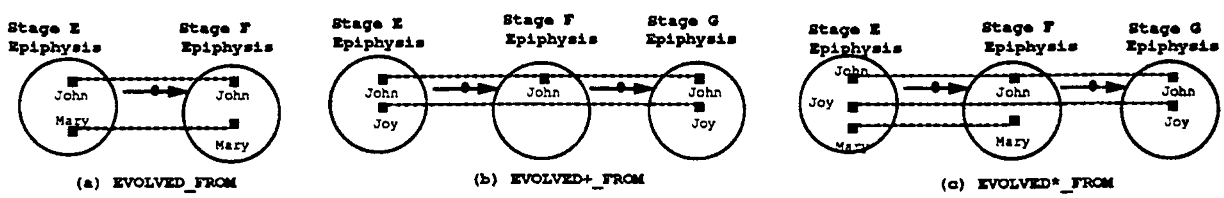

Figure 4 The evolution operators for the thumb metacarpal.

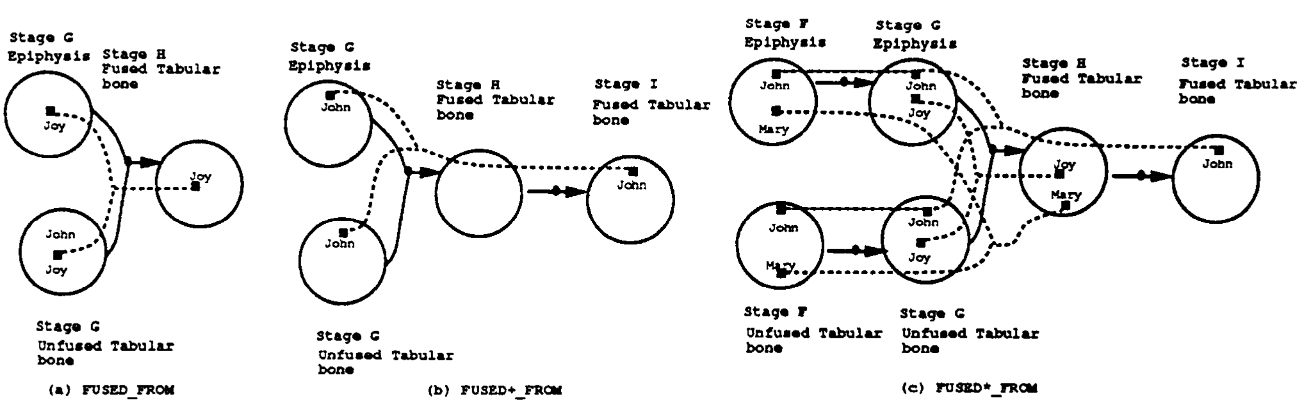

Figure 5 The fusion operators for fusing the epiphysis with the tabular bone into a fused tabular bone for thumb metacarpal.

..., *object_type*$_k$ SPLITTED_FROM *object_type*" selects all single step evolutionary sequences with a fission in the evolutionary net from *object_type* to *object_type*$_1$, *object_type*$_2$, ..., and *object_type*$_k$. SPLITTED+_FROM and SPLITTED*_FROM are defined similarly.

4.4 WHEN

The WHEN clauses select the appropriate snapshot of the data of interests at a particular point in time. We next discuss the temporal operators.

4.4.1 Temporal Functions

Several useful temporal functions are described below: NOW[14] is a special function used to return the current time point. The "+" and "-" symbols are used together with NOW to indicate the time relative to the current time. For example, "NOW - 3 years" stands for three years before the current time. START_TIME, END_TIME, EVENT_TIME, and RECORD_TIME are the methods used in the query language to retrieve the start time, end time, event time, and record time of objects. Notice that START_TIME and END_TIME return a non-versionable object while EVENT_TIME and RECORD_TIME return a versionable object which contains the event time or the record time for the object versions in ascending order. The temporal functions are used to specify the selection criteria in WHEN clauses.

4.4.2 Temporal ordering functions

Temporal ordering of an object history sorts the object versions in ascending order based on their time stamps so that retrieval of object versions in a specific order can be specified. In this paper, we introduce FIRST, LAST, and N_th[16][21] as the forward temporal ordering functions. The parameter for functions FIRST, LAST, is a temporal sequence or a sequence of temporal intervals. The output is the desired object. The N_th functions requires an additional input specifying the desired object with a number. The FIRST and LAST can also be used with an additional number to select, for example, the last 4 hand images. PRIOR is a method with an input of a temporal event and returns the temporal object which exists prior to the input temporal event. NEXT is a method used to retrieve the temporal object that follows a reference object.

4.4.3 Temporal interval comparison operators

To specify a more complex temporal condition, the interval specified in the WHEN clause may be subjected to temporal interval comparison operators[23], such as PRECEDES, FOLLOWS and DURING. These operators specify how the intervals following the WHEN clauses are related to some other time intervals. The time point comparison operators such as BEFORE and AFTER are also included.
The above set of temporal interval comparison operators can be used in a Boolean expression.

4.5 SELECT

The SELECT clauses identify the attributes and object types that are to be operated on by the specified operations. It eliminates attributes and classes that are not relevant to the operations.
The operations in the Operations clauses can be either a system-defined or user-defined data, image manipulation, or visualization operation[11], such as movie_loop, display, contour, rotate, and superimpose.

5 Sample Queries

In this section, we shall present several sample queries to illustrate how to use the temporal and evolutionary language constructs to express certain clinical queries associated with radiographic findings in diagnostic images.

5.1 Temporal Queries

Patients with Turner's Syndrome often have a positive metacarpal sign: a line drawn tangential to the fourth and fifth metacarpal heads intersects the third metacarpal head. Shown in Figure 6 is the hand image of a XO Turner's Syndrome patient with a positive metacarpal sign[20].
Query 1: Retrieve the hand images taken in the last 3 years for any patient with a positive metacarpal sign.
To process this query, let us first build a high-level operator *collinearity* which involves the sizes and the spatial locations of the bones in the hand images to test the collinearity of the third, fourth, and fifth metacarpal heads. Based on the features extracted from the hand images[17][18][19] (Figure 2a) and in conjunction with the knowledge about the metacarpal sign, the above high-level query can be translated into the following TEQL query:

```
WHERE collinearity(stage_I_third_metacarpal,
      stage_I_fourth_metacarpal,
      stage_I_fifth_metacarpal ) = 'positive'
WHEN  hand EVENT_TIME IN [NOW - 3 years, NOW]
SELECT hand image
DISPLAY
```

To execute this query, the system searches the entire hand image database and selects the ones that satisfy the query specifications.

5.2 Evolutionary Queries

To show the fusion process of thumb metacarpals, we have *Query 2: Show a movie loop of the fusion process for the thumb metacarpals.*
Similar to Query 1, by using the data model shown in Figure 2a, Query 2 can be expressed as the following TEQL query:

```
WHICH fused_tabular_bone_stage_I_thumb_metacarpal
      FUSED+_FROM
      epiphysis_stage_G_thumb_metacarpal,
      unfused_tabular_bone_stage_G_thumb_metacarpal
SELECT thumb_metacarpal image
MOVIE_LOOP 3 times frame_delay 2 seconds
```

The thumb metacarpal images as presented in Figure 2b will be shown in sequential order from stage G to I for three times with a frame delay of two seconds for each sequence.

5.3 Temporal Evolutionary Queries

The evolutionary conditions and the temporal functions can be combined together to express more complex clinical conditions. For example, it takes about 4 years for the epiphysis and the tabular bone to completely fuse. If it takes longer, the patient may have a certain underlying disease. Therefore, the following query may be asked:
Query 3: Show a movie loop for each sequence of images that demonstrate the fusion process of the epiphysis and the unfused tabular bone for the thumb metacarpal from developmental stage G to stage I if the whole process lasts longer than 4 years for Caucasian patients.
This query can be translated into the following TEQL query by using the temporal operators FIRST, LAST, EVENT_TIME, and BEFORE and the evolutionary operator FUSED+_FROM,

```
WHERE hand ethnic_group = 'Caucasian'
WHEN FIRST hand EVENT_TIME
  IN [
     fused_tabular_bone_stage_I_thumb_metacarpal
     FUSED+_FROM
     epiphysis_stage_G_thumb_metacarpal,
     unfused_tabular_bone_stage_G_thumb_metacarpal
     ] + 4 years
  BEFORE LAST hand EVENT_TIME
  IN [
     fused_tabular_bone_stage_I_thumb_metacarpal
     FUSED+_FROM
     epiphysis_stage_G_thumb_metacarpal,
     unfused_tabular_bone_stage_G_thumb_metacarpal
     ]
SELECT hand image
MOVIE_LOOP 3 times frame_delay 2 seconds
```

We now consider how to use radiographic findings to classify the patient population with Turner's Syndrome. Patients with Turner's Syndrome typically demonstrate a fourth metacarpal bone which has matured earlier than that of the second metacarpal. The hand image of a 14 year old girl with XX/XO karyotype Turner's Syndrome is shown in Figure 7. Note the early fusion of the epiphysis of the fourth metacarpal. Further, the thumb, second, third, and fifth growth plates are still open (shown by the arrows). This early fusion in Turner's Syndrome may account for the shortening of this bone[20].
Query 4: Show the hand images of patients in which the epiphysis and the unfused tabular bone of the fourth metacarpal fused earlier than those of the second metacarpal.
This query involves the comparison of the fusion processes of the second and the fourth metacarpal bones. It is a rather complex query. In the same manner as Query 3, using the PRECEDES and FUSED+_FROM operators, it can be translated in the TEQL query shown below:

```
WHEN [ FIRST epiphysis_stage_G_fourth_metacarpal
  EVENT_TIME
  IN [
     fused_tabular_bone_stage_I_fourth_metacarpal
     FUSED+_FROM
     epiphysis_stage_G_fourth_metacarpal,
     unfused_tabular_bone_stage_G_fourth_metacarpal
     ],
  LAST fused_tabular_bone_stage_I_fourth_metacarpal
  EVENT_TIME
  IN [
     fused_tabular_bone_stage_I_fourth_metacarpal
     FUSED+_FROM
     epiphysis_stage_G_fourth_metacarpal,
     unfused_tabular_bone_stage_G_fourth_metacarpal
     ], ] ]
  PRECEDES
     [ FIRST epiphysis_stage_G_second_metacarpal
  EVENT_TIME
  IN [
     fused_tabular_bone_stage_I_second_metacarpal
     FUSED+_FROM
     epiphysis_stage_G_second_metacarpal,
     unfused_tabular_bone_stage_G_second_metacarpal
     ],
     LAST fused_tabular_bone_stage_I_second_metacarpal
  EVENT_TIME
  IN [
     fused_tabular_bone_stage_I_second_metacarpal
     FUSED+_FROM
     epiphysis_stage_G_second_metacarpal,
     unfused_tabular_bone_stage_G_second_metacarpal
     ], ] ]
SELECT hand image
DISPLAY
```

The first half of the query (before the PRECEDES operator) determines when the fusion process of the fourth metacarpal occurs; while the second half determines when the fusion of second metacarpal takes place. Using the temporal interval comparison operator PRECEDES, we are able to compare whether the fusion process of the fourth metacarpal occurs earlier than the second. The query searches the entire database for all evolutionary sequences of hand images and identifies the ones that satisfy the query conditions.

6 Implementation

The prototype of the system has been implemented on top of Gemstone, an object-oriented database. Using an existing OODBMS provides support for such database capabilities as persistence, concurrency control, and recovery. Thus, it shortens the development time considerably. We have developed an algebra to realize the evolutionary operators. The query is parsed by the query processor and translated into an algebra form. The algebra form of the query is then optimized. Each operator of the algebra is implemented by a method in the Gemstone environment. Image analysis routines that segment, identify, and characterize various bones and objects in the hand images[17][18][19] are integrated with database operations.

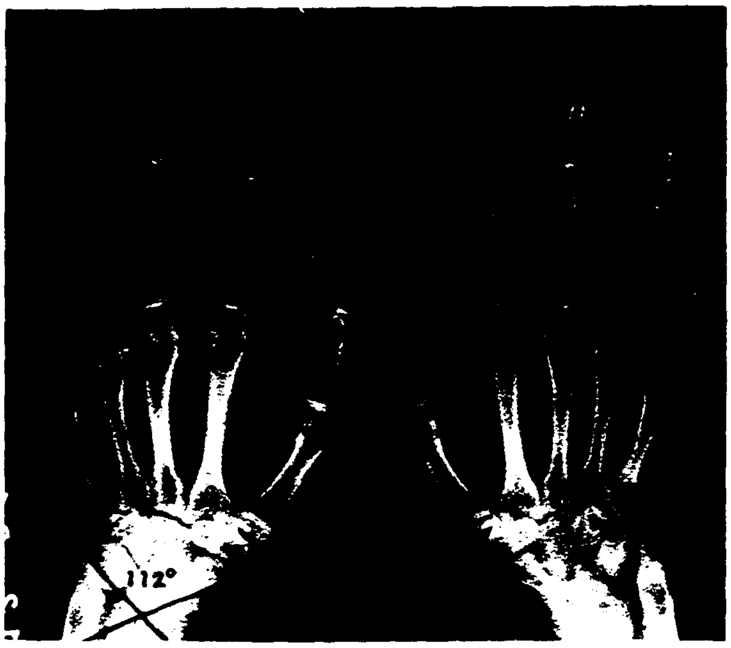

Figure 6 The hand image of a patient with Turner's syndrome. Notice the metacarpal sign is positive. The carpal angle is also diminished (normal=131°). (Courtesy of W. B. Saunders Company)

Figure 7 XX/XO karyotype Turner's syndrome. Notice the early fusion of the fourth metacarpal. (Courtesy of W. B. Saunders Company)

7 Future Work

The proposed TEQL allows us to formulate precise queries. However, in practice, many clinical conditions often cannot be expressed precisely. For example, conceptual queries such as "Retrieve all images for *pre-adolescent* oriental males with radiographic findings consistent with *Turner's Syndrome*" contain imprecise descriptors (e.g. pre-adolescent and Turner's Syndrome). Cooperative query processing techniques[6] can be used to handle such queries. Therefore, we are in the process of building a knowledge-based hierarchy to model the correlation among different diseases with patient subpopulation and image features. Also, a user-friendly graphical interface is being developed to input queries and specify the desired visual output and result representation.

8 Conclusion

This paper presents a temporal evolutionary object-oriented data model (TEDM) for modeling medical images. The proposed query language TEQL provides temporal evolutionary content-addressable capabilities for medical image retrieval. Our intelligent medical image management system (IMIS) lies on top of a picture archive and communication system (PACS) infrastructure. The IMIS can retrieve medical images (e.g. x-rays, computed tomography scans, magnetic resonance scans, etc.) by image features and contents rather than by traditional artificial keys such as patient hospital identification number. As a result, solutions to queries which associate the radiographic findings of an image, the disease pathology, and the categorical patient subpopulation can be obtained. This represents a significant advance in medical information processing. Our proposed model and language constructs can also be applied to the other domains that exemplify the evolutionary transformations of objects, such as modeling the growth of brain tumors.

References:

1. A. M. Alashqur, S. Y. W. Su, and H. Lam, "OQL: A query language for manipulating object-oriented databases", Proc. of the 15th VLDB, 1989.
2. J. Andany, M. Leonard, and C. Palisser, "Management of schema evolution in databases", Proc. of the 17th VLDB, 1991.
3. J. Banerjee, W. Kim, H. J. Kim, and H. F. Korth, "Semantics and implementation of schema evolution in object-oriented databases", Proc. of the ACM SIGMOD, 1987.
4. M. Bouzeghoub "Morse: a functional query language

and its semantic data model", Proc. of int. Conf. on Trends and applications of databases, IEEE-NBS, Gaithersburg, USA, 1984.

5. M. Bouzeghoub and E. Metais, "Semantic modeling of object oriented databases", Proc. of the 17th VLDB, 1991.
6. W. W. Chu, Q. Chen, and R. Lee, "Cooperative Query Answering via Type Abstraction Hierarchy", Proc. of the CKBS, October 1990.
7. S. J. Dwyer III, et. al., "The cost of managing digital diagnostic images", Radiology, Vol. 144, July 1982, pp. 313-318.
8. A. Gupta, T. Weymouth, and R. Jain, "Semantic queries with pictures: The VIMSYS model", Proc. of the 17th VLDB, 1991.
9. H. K. Huang, N. J. Mankovich, and R. K. Taira, " Picture archiving and communication systems (PACS) for radiological images: State of the art", CRC Critical Reviews in Diagnostic Imaging, 28(4), 1988.
10. R. Hull and R. King, "Semantic Database Modeling: Survey, Applications, and Research Issues", ACM Computing Survey, Vol. 19, No. 3, September 1987.
11. T. Joseph and A. F. Cardenas, "PICQUERY: A high level query language for pictorial database management", IEEE Transactions on Software Engineering, Vol. 14, No. 5, May 1988, pp. 630-638.
12. W. Kim, "A model of queries for object-oriented databases", Proc. of the 15th VLDB, 1989.
13. P. Kofakis, S. C. Orphanoudakis, "Graphical tools and retrievl strategies for medical image databases", Proc. of the International symposium on computer Assisted Radiology, pp. 519-524, Springer-Verlag, 1991.
14. C. H. Kung, "A temporal Framework for database specification and verification", Proc. of the 10th VLDB, 1984.
15. M. Morgenstern, "Constraint equations: Declarative expression of constraints with automatic enforcement", Proc. of the 10th VLDB, 1984.
16. S. B. Navathe and R. Ahmed, "A temporal relational model and a query language", International journal of information science, Vol. 48, No. 2, 1989, pp. 57-73.
17. E. Pietka, Lotfi Kaabi, H. K. Huang, and M. L. Kuo, "Feature extraction for bone age determination", Radiology, 177, 1990.
18. E. Pietka, M. f. McNitt-Gray, M. L. Kuo, H. K. Huang, "Computer assisted phalangeal analysis in skeletal age assessment", IEEE Trans. on Medical Imaging, Vol. 10, No. 4, pp 616-620, 1991.
19. E. Pietka, L. Kaabi, M. L. Kuo, and M. K. Hunag, "Feature extraction in carpal bone analysis", IEEE Computer Graphics and Applications, 1991.
20. A. Poznanski, "The hand in radiologic diagnosis: with gamuts and pattern profiles", 2nd edition, W. B. Saunders Company, 1984.
21. E.Rose and A. Segev, "TOODM - A temporal object-oriented data model with temporal constraints", University of California, Technical report LBL-30678, April 1991.
22. N. Rossopoulos, C. Falotsos, and T. Sellis, "An efficient pictorial database system for PSQL", IEEE Trans. on Soft. Engin., Vol. 14, No. 5, pp. 639-650, 1988.
23. R. Snodgrass, "The temporal query language for TQUEL", ACM TODS, June 1987, pp. 247-298..
24. R.Snodgrass and I. Ahn, "Temporal Databases", IEEE Computer, September 1986.
25. S. Su and H. Chen, "A Temporal knowledge representation model OSAM*/T and its query language OQL/T", Proc. of the 17th VLDB, 1991.
26. R. Taira and H. Huang, "A Picture Archiving and Communication System Module for Radiology", Computer Methods and Programs in Biomedicine, 30, 1989.
27. J. M. Tanner, R. H. Whitehouse, W. A. Marshall, M. J. R. Healy, H. Goldstein, "Assessment of Skeletal Maturity and Prediction of Adult Height (TW2 Method)", 2nd Edition, Academic Press, London, 1983.
28. E. Tsang, "Time structures for AI", Proc. of the 10th IJCAI, 1987.
29. S. B. Zdonik and D. Maier, "Readings in object-oriented database system", Morgan Kaufmann Publishers, Inc. 1990.
30. L. Cardelli, "Semantics of Multiple Inheritance", in [29].