Asian Journal of Information Technology 6 (2): 170-175, 2007

The Use of Conceptual Query Processing for Livestock Management System

O.T. Arogundade and A.T. Akinwale
Department of Computer Science, University of Agriculture, Abeokuta, Nigeria

Abstract: This study presents a concept based information system for livestock management system. It employs a semantic graph models that use nodes as entities set and edges as semantic roles. The data base end-users formulate queries by selecting a set of edges between selected sets of nodes and specifying for each node a set of data retrieval criteria. The system displays all schema, attributes and criteria which the end-user can select in form of natural question. The system arranges them and transforms them into Structural Query Language (SQL) statement for execution. At end, it enables the end users to pose desired query statements that generate reports for decision making about livestock farm at University of Agriculture, Abeokuta, Nigeria.

Key words: Schema database system, conceptual query language, structural query language, microsoft structural query language server DBMS, set cover partition, topological sort, mix fix function, extract minimum paths, query formulation

INTRODUCTION

Among the duties of livestock management system at University of Agriculture, Abeokuta, Nigeria, include monitoring of the animals in the farm in terms of inventory, routine treatment, sales, purchases and so on. In Nigeria, many data base end-users are not data base specialists. They find it very hard to write SQL statement that will generate simple and accurate reports.

To achieve this, sets of existing algorithms are employed in this study. These algorithms are modified to suit the aims of the work. These algorithms are set cover partition, topological sort, mix fix and extract minimum paths. Set cover partition is used to partition the scheme graph based on source node and target node of the query statement. This approach elevates the burden of having to perform extrusive search through the whole scheme graph. The topological sort allows linking the set partition from right edge (source) to left edge (target). Mix-fix function is employed to enhance the semantic meaning of the query statement results. Extract minimum paths choose the shortest semantically correct path sets for query statement (Thomas *et al*., 2001).

Research works on semantic query formulation and optimization have been a thing of interest because of the benefits and the ease of querying it offers to the end-users. A good discussion on research efforts in semantic query formulation and optimization can be found in (Piittges, 1995). A few recent works have been done in conceptual query formulation (Owei and Navathe, 2001). Most of these works were done in the free context of query formulation form for databases.

SEMANTIC ENTITY-RELATIONSHIP SCHEMA

The application of the research employs livestock farm of University of Agriculture, Abeokuta, Nigeria. The research identifies the set of entity with their relationship semantics. The entity sets and their semantic links are put together to design semantic relationship schema. Figure 1 illustrates the semantic relationship of livestock farm of the University of Agriculture, Abeokuta. As shown in Fig. 1, entity sets bear explicitly named relationships among themselves. Each relationship has semantic meaning. The schema uses many to many, many to one and one to one entities relationships. Among the semantic roles include carries_out, assigns, is_assigned_by, takes_care_of, affect_by, etc, by the entities in the specific relationships. The semantic of the links between entities lies in the forms of roles.

From the graph theory point of view, entities are conceptually conceived of as graph nodes and link semantic roles as graph edges. Therefore Fig. 1 can be given as directed graph G(V, E) where V is a non-empty finite set V(G) of elements called the entities (vertices) and a finite family E(G) of ordered pairs of elements of V(G) called semantic roles (arcs). We call V(G) the vertex and E(G) the arc Family of G.

Corresponding Author: O.T. Arogundade, Department of Computer Science, University of Agriculture, Abeokuta, Nigeria

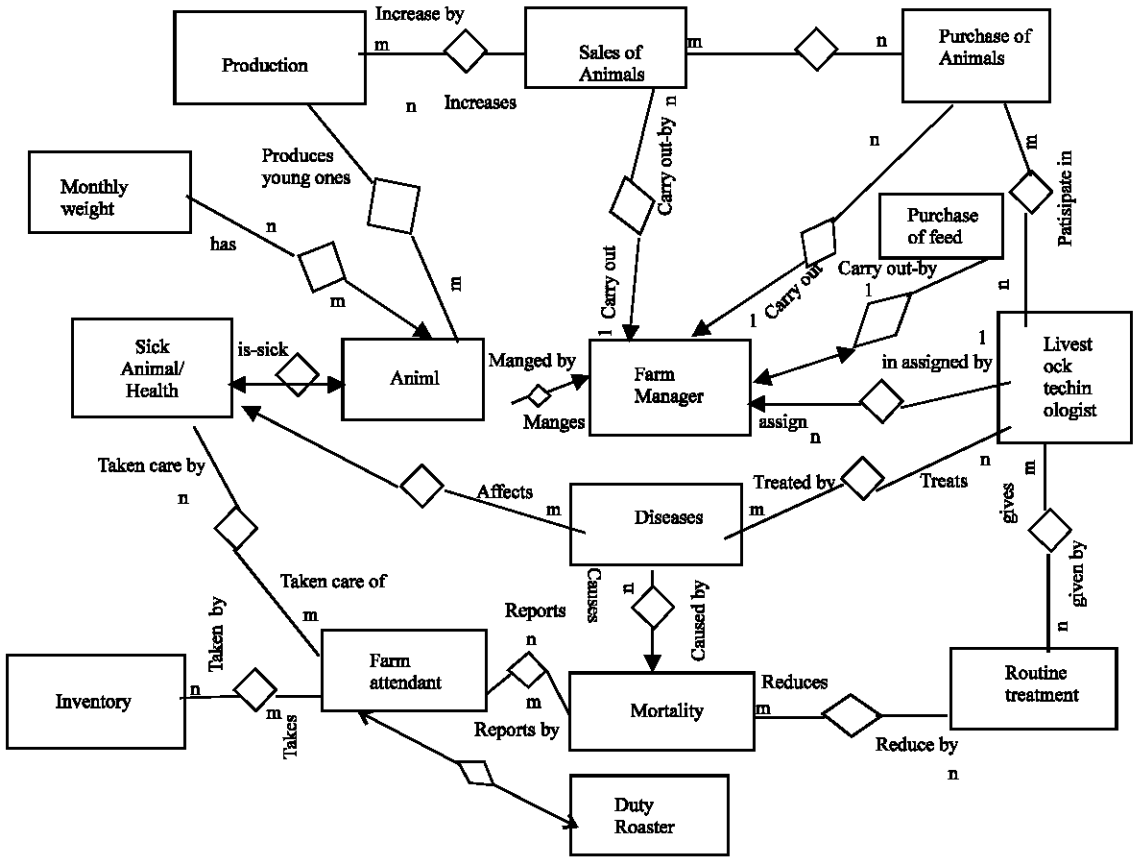

Fig. 1: Semantic relationship of livestock farm managment system

SCHEMA ENTITIES AND THEIR ATTRIBUTES

The schema entities and their attributes which serve as Livestock Farm data base are as follows:

Animal(tag_Number, animal_Name, sex, birth_Date, birth_Weight, dam_Number,
sire_Number, breed_Type)
Inventory (date, animal_Name, male, female, total_Number)
Disease (tag_Number, animal_Name, disease_Type, Symptons, weight)
Attendant (officer_Id, officer_Name, animal_Name, duty_Period, remarks)
Livestock_Technologist (officer_Id, officer_Name, tag_Number,
animal_Name, date_Inspection)
Sick_Animal(tag_Number, animal_Name, sex, disease_Type,
period_of_Treatment, cost_of_Treatment, remark)
Weight(previous_Weight, present_Weight, tag_Number, animal_Name, remark)
Sales_of_Animal (date, tag_Number, animal_Name, officer_Id,
present_Weight, amount_sold, buyer_Name)
Purchase _ of _ Animal (date _ Purchase, animal_Name,
present_Weight, number_Animal_Purchased, officer_Id, seller_Name)
Purchase_of_Feed (date_Purchase, name_Feed, grade_Quality, quantity,
amount, officer_Id, seller_Name)
Routine_Treatment (date, animal_Name, date-last-Inspected,
total_Number, officer_Id, remark)
Production (tag_number, animal_Name, delivery_Date, production_Period,
little_Size, mortality_Rate, survivor_Number)

Mortality(date, tag_Number, animal_Name, symptom, number_Died, officer_Id)

COMPONENTS OF QUERY PROCESSING ALGORITHMS

In generating the query paths, there is an initial derivation of all paths from source to target. This task is carried out by partitioning the semantic graphs into sets of nodes that include both the source node and the target node. Let V be the set of all the nodes. Let f be subsets of V, sets of all the paths from the source to the target node on the graph. The step by step of the query processing is illustrated in the algorithm 1.

```
Input = source node (beginning of the relations)
target node (end of the relations)
int main ( ) {
 G (V, E, f)
      V  =  Finite entity sets
      E  =  Finite semantic roles
      F  =  {h(s, t), s, t å V s = source entity nodet =
            target entity node}
      C  =  Set-cover_partition (V, E, s, t)
      P  =  Topological-sort (C)
      D  =  Mix-fix-function (P)
      R  =  Extract-minimum (D)
      return 0;
}
set-cover partition (V, E, s, t){
      C = 0, P =0;
      s,t ε V→t
      P = s U Vt
      C = C U {P}→ t
      return C
 }
      topological-sort (C) {
      call DFS (C)
      P= entity sets (s, t, E C)
      return P
}

mix-fix-function (P) {
      P (C, E)
      s, tε [C]
      E = semantic roles
      Mix-fix [C1 C2 C3 …… Cm] [E1 E2 E3 …… En)
```

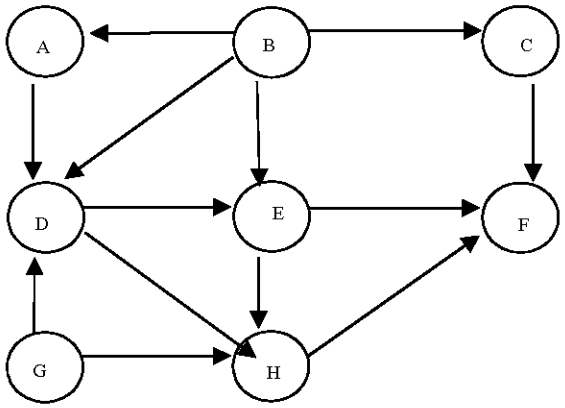

Fig. 2: Behavior of set cover partition

```
      D = C1 E1 C2 E2 C3 E3 C4….. En-1 Cm
      return D
}
Extract-minimum (D) {
      D = G (C, E)
      R = Minimum (count (C[ i , j ])) or Minimum (count
      (E[i , j ]))
      return R
}
```

The algorithm 1: Step by step of query processing

SET COVER PARTITION

The function set_cover_partition will divide the whole graph into different subsets that contain source entity-set and target entity set. The selected set paths from source to target node are stored in C. The selected paths contain source and intermediate nodes that lead to the target node. Figure 2 explains the role of set cover partition.

Assuming in the Fig. 2, the source node is B and the target node is H, then the set cover partition can be as follows: { B, E, H }, { B, A, D, E, H }, { B, A, D, H }, { B, D, H }.

The topological sort module arranges entity sets along each path set from source (B) to target (H) in order of edges connection, taking into consideration their semantic roles. Figure 3 illustrates the connection of entities' nodes, intermediate nodes, and intermediate semantic links from source and target.

Mix-fix function is used to validate the semantic correctness of the query paths from set partition and topological sort. It is employed to formally validate the

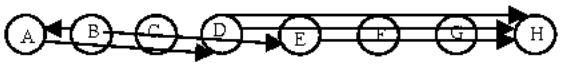

Fig. 3: Behavior of topological sort

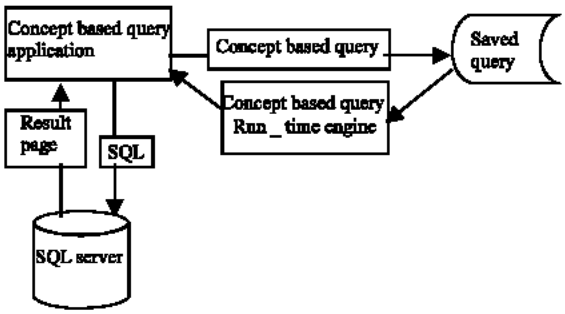

Fig. 4: Architecture of query processing

semantic correctness of the query paths resulting from set partition and topological sort.

If $w = \{w_1 w_2 \ldots w_m\}$ and $P = \{P_1, P_2, P_3 \ldots P_n\}$ are sets of names, then mix-fix expression on those sets can be used to describe different possible sentential verbalizations involving the sets. Let $w = [w_1, w_2, \ldots, w_m]$ and $P = [P_1, P_2, P_3, \ldots, P_n]$ represent ordered instances of w and P. The mix-fix verbalization on w and P is given by: Mix-Fix $([w_1, w_2, \ldots\ldots w_m], [P_1, P_2, P_3, \ldots, P_n]) = P_1 w_1 P_2 w_2 P_3 , \ldots, w_{m-1} P_n$.

This extract minimum path function is associated with entity set and semantic roles. In G(C,E), the length of an edge directed from entity-set(i) or semantic-role (i) to entity-set (j) or semantic role (j) is denoted by count (i, j). If there is no edge directed from entity-node(i) or semantic role(i) to entity node(j) or semantic role(j), then count[i, j] = 0. The minimum path is the minimum number of count of the entity nodes or semantic roles in the path. This is the minimum number of count directed paths from source to target.

Queries executions are formulated in a user friendly. SQL in Microsoft SQL Server DBMS can only generate results on SQL statements. The select clause in SQL is equivalent to target node in the Conceptual Query Language (CQL). The from clause is the same as the source nodes and other intermediate nodes on the set cover partition. The where clause corresponds to source nodes and their values.

QUERY PROCESSING ARCHITECTURE

The concept based query application is implemented as a front end to a Microsoft Structural Query Language Server Data Base Management System. Figure 4 shows the query processing architecture for the system. The

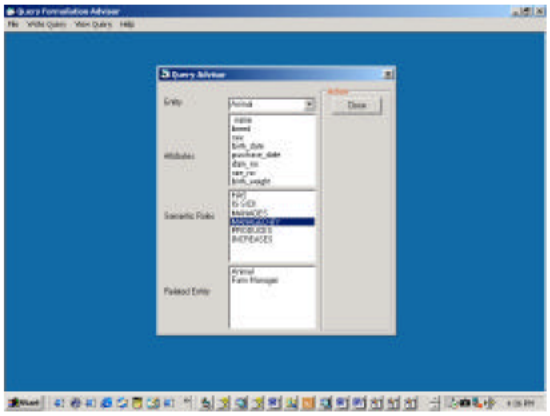

Fig. 5a: Query Formulation Advisor (QFA) screen

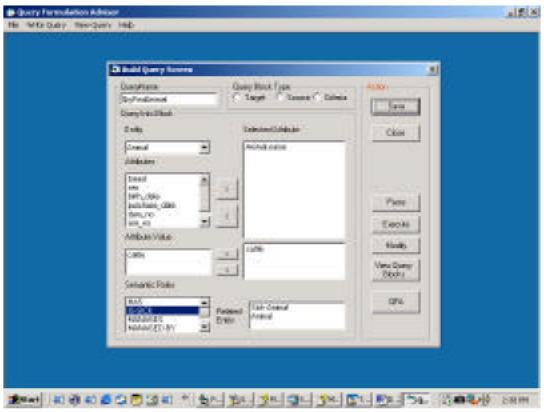

Fig. 5b: Build query screen

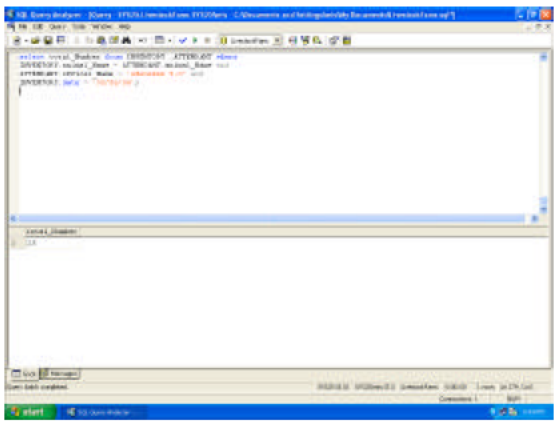

Fig. 5c: Query answer screen

main components of the query processing architecture are the following: the concept based query application, the concept based query run-time engine and SQL Server.

CONCEPT BASED QUERY APPLICATION

The concept based Query Application provides end-users with powerful user interface facilities which help them to tailor their queries to the target database. The first of these interfaces is the Query Formulation Advisor (QFA). Query formulation advisor utilizes the conceptual schema concepts on its operation. It provides users with helpful information on the schema concepts that they must be familiar with in order to be able to formulate queries. By utilizing a conceptual schema, a system could guide the user through a stepwise construction of his desired query. With the assumption that users know what they want, they can compose the query with the help of QFA.

Figure 5a shows the query formulation advisor screen. In this screen, the user can pick any entity of interest by checking the down arrow in the window labeled entity. This action will automatically display the attributes of the selected entity, its semantic roles and the entities related to it through the semantic roles. In Fig. 5a, the entity "Animal was chosen, its attributes, semantic roles and related entities were displayed.

Figure 5b, shows the build query screen which is another powerful interface provided by the concept based query application. This is where the user builds his query based on the meta-knowledge acquired from the query formulation advisor. The end-users give a name to their queries in the query name window as seen in Fig. 5b. Then they move to the query block type, where the BFN specification is stated as target source and criteria. The end users knowing what they want will first enter the component of the target query block by selecting the desired entity (ies) and the attributes to the selected attribute window using the arrows shown accordingly. The target block is then saved by the end-user. The same procedure is repeated for source block and the criteria block. The entity and the selected attribute windows are refreshed after saving the content of each query block type. This saved query is then passed to the query run-time Engine.

QUERY RUN-TIME ENGINE

Query run-time engine consists all the functions in the algorithm 1 namely set cover partition, topological sort, mix fix, extract minimum paths and query execution. Here the specified query is interpreted and all necessary concepts such as entities (source, intermediate and target), semantic roles/relationships are extracted for latter use in the aforementioned modules.

PROCESSING AND EXECUTION OF END-USER SAMPLE QUERIES

The data base end user can pose the following question:

"What Disease affect sick animal with tag No.C65 treated by livestock technologist whose name is Oyerinde A. A.?"

For the specified source (Sick Animal and Livestock Technologist) targets (Disease and Livestock Technologist) and conditions of our motivational query, the CQL formulation in terms of canonical form is Query {Disease: (Sick Animal, Livestock Technologist): ({selection: semantic meaning)}}.

When the attributes and values of interest are indicated, the expression becomes as follows:

Query {Disease (Disease Name): (Sick Animal (tag.No. = C653))
Livestock Technologist (Livestock Technologist Name
= "Oyerinde A. A.):

Sick Animal affected_by Disease and
Livestock Technologist treats Disease)}

In this expanded form "Sick Animal affected_by Disease" and "Livestock Technologist treats Disease" are the pertinent semantic relationships for the correct query paths.

In the query run-time engine, the source and the target entities are extracted and passed to the set cover partition function as input. From Figure 1, set cover partition will divide the whole graph into different subsets that contain source and target entity set as follows:

(path 1)
Sick Animal, Disease, Livestock Technologist
(path 2)
Sick Animal, Farm Attendant, Mortality, Disease, Livestock Technologist
(path 3)
Sick Animal, Farm Attendant, Mortality, Routine Treatment, Livestock Technologist
(path 4)

Sick Animal, Animal, Farm Manager, Livestock Technologist

Topological sort module arranges the entity sets within each subset from source to target in order of edges connection, taking into consideration their semantic roles for the sample queries. Topological sort generates the following candidate paths set using Fig. 1.

(path 1)

Sick Animal->affected_by->Diseases->treated_by->Livestock Technologist

(path 2)

Sick Animal->Take_care_of->Farm Attendant->reports->Mortality->causes->Diseases->treated_by->Livestock Technologist

(path 3)

Sick Animal->Take_care_of->Farm Attendant->reports->Mortality->Reduces->Routine Treatment->given_by->Livestock Technologist

(path 4)

Sick Animal->In_sick->Animal->Manage_by->Farm Manager->In_assigned_by->Livestock Technologist

The validation of each of these query paths involves checking the semantic correctness with the user intended query statement. Mix fix function corrects the statement by grouping the nodes with their respective semantic roles. For example, path 1 will be as follows:

Let w be the sets of semantic roles played by schema entities and P be sets of schema entities.

w_1 = affected_by, w_2 = treated_by

P_1 = Sick Animal, P_2 = Disease, P_3 = Livestock Technologist.

$= P_1w_1\ P_2w_2\ P_3$

= Sick Animal affected_by Disease treated_by Livestock Technologist

For path 2 is

w_1 = taken_care_by, w_2 = report, w_3 = reduced_by, w_4 = given_by

P_1 = Sick Animal, P_2 = Farm Attendant, P_3 = Mortality, P_4 = Routine Treatment, P_5 = Livestock Technologist $= P_1w_1\ P_2w_2\ P_3w_3\ P_4w_4\ P_5$

= Sick Animal take_care_by Farm Attendant report Mortality

reduced_by Routine Treatment given_by Livestock Technologist.

This method can be applied to path 3 and 4.

Since more than one path is found to be consistent with the intended query, then extract minimum path chooses the path with the minimum number of entities or semantic roles which in this case is path 1.

The SOURCE and the TARGET nodes are then extracted with the selection criteria/condition. These are passed to query execution module where the extracted ingredients are converted to Structural Query Language statement and then executed. For example the query of the following [What is the inventory of cattle in the farm as recorded by farm attendant whose name is Adenekan S.O. on 30/03/02?] has been converted from CQL into SQL and gives the result as shown in Fig. 5c.

CONCLUSION

In this study, existing algorithms were applied using Conceptual Query Language and Microsoft Structural Query Language Server software package. These algorithms are coded in Visual Basic Programming Language that group entity sets and their relationships in form of natural sentences. It allows the data base end user to formulate query statements that generate simple reports on livestock farm management system. Effort is on to visualize the query statements.

REFERENCES

Owei, V.S.B., 2001. Navathe, Enriching the conceptual basis for query formularization through relationship semantics in databases. Inform. Sys., pp: 26.

Pittges, J., 1995. Metadata view graphs: A framework for query optimization and metadata management. Ph.D thesis, Georgia Institute of Technology.

Thomas, H., 2001. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to Algorithms, (2nd Edn.), MIT Press and McGraw-Hill.