IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com
ITB12671

This paper appears in the book, *Emerging Trends and Challenges in Information Technology Management, Volume 1 and Volume 2* edited by Mehdi Khosrow-Pour © 2006, Idea Group Inc.

A Conceptual GeoSpatial Data Warehouse Model for Forest Information Management

Robert Magai, PhD, Selkirk College Geospacial Research Center, Castlegar, BC V1N 3J1, Phone: 250-365-1349, Fax: 250-365-6568, rmagai@selkirk.ca

Wookey Lee, Division of Computer Engineering, Sungkyul University, Anyang, Korea, Phone: 82-31-467-8067-8174, Fax: 82-31-467-8067-8067, wook@sungkyul.ac.kr

ABSTRACT

An architectural framework is developed for a GeoSpatial Data Warehouse (GSDW) for which a conceptual model is designed that accommodates the following dimensions: spatial (map object), temporal (time), agent (contractor), management (e.g. planting) and tree species (specific species) of the GSDW information. According to the GSDW model, spatial queries such as spatial selection, spatial projection, and spatial join are defined and incorporated along with other database operators having interfaces via the model. This GSDW model was implemented using data from the Malcolm Knapp Research Forest of the University of British Columbia (UBC). Based on cost function estimates, preliminary experimental results show that our view materialization of the GSDW model performs highly as compared to re-computation and intermediate methods.

1 INTRODUCTION

A data warehouse is an incorporated infrastructure that stores and analyzes data to aid decision support and in which operational data is reprocessed, aggregated and stored in base tables. One of the fundamental difficulties in designing a general spatio-temporal database is its data model. Incorporating both time and space in data models increases the complexity of the data structure and is a challenging task [7]. Entity-Relationship (ER) and its extensions are suggested to capture spatial semantics in geospatial modeling.

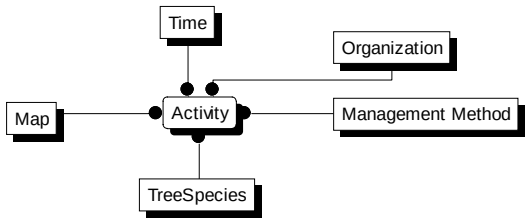
We focused on a special domain named the Malcolm Knapp Research Forest (MKRF) whose basic function is to produce timber efficiently at sustainable levels. Within this context, the production process follows certain business rules, which influence the type of data model that is applied. Operationally, MKRF has used data are stored in various electronic and hard copy formats. Hence a decision was made to create a more modern and centralized database that would facilitate storage and retrieval of forest resource data and create a resource for forestry strategic planning and decision-making.

The objective of this paper is to develop a framework for integrating geospatial and temporal data with relational information. In order to achieve full integration, data should flow bi-directionally from source to destination and vice versa between spatial and temporal/relational domains.

2 GEOSPATIAL DATA WAREHOUSE MODELING

A system managing geospatial information should provide database support for a diverse range of activities. In the forestry context, this would include carrying out simultaneous silvicultural management activities at various locations, such as planting, harvesting, surveying, personnel, and using specific agents. To facilitate sharing of geospatial information across the board, a common data model needs to be adopted for each type of activity. The common data model introduced here is known as a 'least common denominator,' and can be referred to as a 'basic model,' which can be applied as a consistent framework for our objects.

Figure 1. Basic model of GSDW



The conceptual GSDW model is given in Figure 1. The model has five dimensions associated with a fact table that contains fact attributes (e.g., number of trees planted). Each box represents an entity. The entities are linked by referential integrity; the black dot at the end of the line represents cardinality.

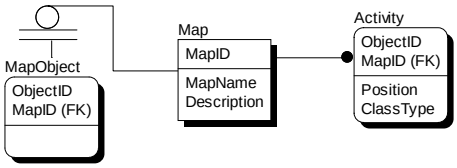
3 GSDW ISSUES

3.1 Spatial Dimension

As a dimension table for the data warehouse, we have considered a spatial object as an abstract data type that will be described later in detail. The highest abstract level of geometry in this paper is a map. The map is a collection of spatial objects, for instance a Forest Cover Polygon (FCP), and this spatial object is derived from the real world forest. A FCP is defined as a stand of trees with similar attributes at a particular time. In designing a dimension for data warehouse, there are two extreme choices: (1) The highest abstract level the object is engaged in, or (2) the detailed object instances the object is engaged in. The latter was introduced by Tryfona et al., [12], as representing the interface between the spatial object and other objects (e.g., relational objects); and that individual objects or instances had to be engaged as dimensions. Their model was too complex and vague for a detailed fact table and associated dimension tables. In this paper, however, the former approach is applied, because, the map is considered as a whole (not as individual objects) engaged in the GSDW. The dimension should be simple, and the details can be maintained in the MapObject instance table (Figure 2).

A spatial object may consist of other spatial objects. The objects may be included in a number of homogeneous domain maps, e.g., a hydrographic domain that has streams, lakes, or reservoirs. The spatial object consists of a set of maps as layers.

Figure 2. Map interface of the GSDW model



3.2 Temporal Dimension

A temporal aspect in the data warehouse data model is recorded by design, in terms of years. For example, days, weeks, and months will not be factored into the design, simply because the effects of most forestry operations are observed after a number of years have elapsed. Therefore, warehouse updates will be occurring annually. The temporal notations by Snodgrass [9] and Slivinskas et al., [8] were considered. Two kinds of temporal dimensions considered are time points and periods. The schema of temporal dimension is composed of start-time and end-time, so the time point can be represented by either one of the two, and both temporal dimensions represent a period of time.

3.3 Relational Dimension

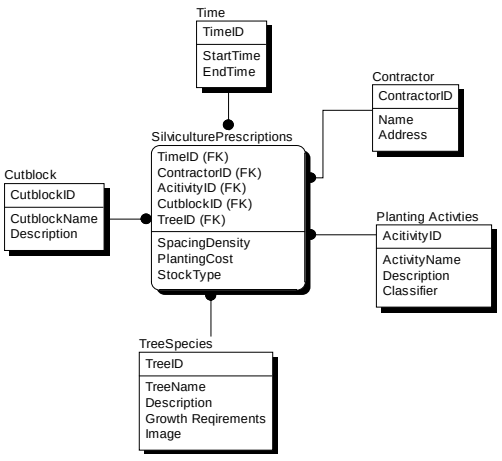
Three relational dimensions are considered in our model: Organization dimension, Tree Species dimension, and Management Activity dimension, representing who, what, and how of the model, respectively. These three relational dimensions are mandatory factors in the model. The most important factor is the Management Activity dimension. We apply these dimensions consistently all though the management activity options (e.g., preparing sites, planting, harvesting etc.).

3.4 Management Point of View

With the same data model, we can generate forestry management points of view based on activity options such as harvesting, planting, rehabilitating a cutblock, etc.

- **Silviculture Management Activities** – allow for the planning and creation of silviculture prescriptions and tracking silviculture management activities at the cutblock or strata level. This module is linked to cutblock and strata map layers. An illustration of a model on silviculture prescription activity is given in Figure 3. In the model, all the dimension factors are engaged in the instances.

Figure 3. Modeling for Silviculture Management activity



Similar management points of views can be made for timber production, harvesting, pre- and post-maintenance, timber inventory surveys and road management activities.

4 GEOSPATIALTEMPORAL OPERATORS

4.1 Terms and Definitions

In this section, we present conceptual geospatial data warehouse queries independent of any database representation. Data warehouse data are defined as a set of database states consisting of current as well as historic states. So we assume that a data warehouse schema also includes a database schema, in which case the database is focused on the current state of a base relation whereas a data warehouse is focused on collecting the state trajectory (history) of the base relation.

We define a data warehouse schema Ω as a triplet component (i.e., spatial, relational, and temporal). It is assumed that data warehouse objects include data, operator, and queries in the relational terminology. For clarity and simplicity, only spatial objects and spatial operations are redefined in the paragraphs that follow.

A geospatial data warehouse schema $\Omega = <\Sigma, \Re, \Psi>$ follows a triplet where $\Sigma = \{s_1, s_2, \ldots, s_n\}$ is a spatial schema consisting of FCP s_i, DB schema $\Re = \{r_1, r_2, \ldots, r_m\}$ is a set of base relation instances, and $\Psi = \{t_1, t_2, \ldots t_l\}$ be a set of temporal instances t_i for all i representing a finite number of an index set. The following notations and algebraic expressions are introduced:

$\Omega = <\Sigma, \Re, \Psi>$ where Σ is a spatial schema, $\Re$ a relational schema, and Ψ a temporal schema, respectively.

$|\ \omega = \{<s_i, r_i, t_i>|$ condition C$\} \in \Omega$ is a triplet instance set that satisfies the condition C.

$|\ \sigma^{\Psi}((x\ \theta\ P)|$ where $x \in \omega$) is a select operator with a predicate condition p for an argument Ψ

$|\ \Pi_A^{\Psi}(<s_i, r_i, t_i>)|$ is a project operator over a set of attributes A for an argument Ψ

$|\ \square^{\Psi}((x\ \theta\ y \mid x =$ condition, y=condition), where $x, y \in \omega$) join operator on x and y for an argument Ψ

$|\ \Psi \in \{s, r, t\}$, where s is a spatial, t temporal, and r a relational argument respectively

$|\ \theta \in$ a comparison operator set $\{=, <, \leq, >, \geq, \neq\}$

Definition: Spatial entity: We define a thematic map as a collection of spatial objects derived from a real world forest. The map is the highest level of a spatial set domain and the map consists of several objects. The spatial object consists of a set of maps as layers. Below is a definition of a map schema.

- Spatial objects (s_i) = {<MapID, Map: Spatial type>}
- Map = {<OID, FCP object, Position, Description>}

Where the MapID and OID are identifiers and the Map consists of subcomponents such as a map object, called the Forest Cover Polygon (FCP), Position is represented by (longitude, latitude), and Description details can be included such as height, orientation, etc.

Spatial Selection: The spatial selection (σ^s) is defined as selecting a portion of a map that matches a condition of a predicate of alphanumeric attribute. The signature of spatial selection can be represented as Map $\times$ condition $\rightarrow$ Map, where condition is a predicate of one or many alphanumeric attributes: condition = predicate (Ai). The spatial selection is defined as follows:

Spatial selection = {<s_i, r_i, t_i>| x = σ^s (s_i.attribute θ Constant); x∈Σ, r_i∈ℜ, t_i∈Ψ}

Example 1: A user query is issued as follows: '*Show Forest Cover Polygons within 50m from a lake'*. It can be represented formally: {<s_i, r_i, t_i>| x= σ^s (|s_i - a lake| < 50m); x∈Σ, r_i∈ℜ, t_i∈Ψ}.

Note that the notation on spatial objects such as spatial selection, spatial projection, and spatial join (including temporal notations) follows the standard database representation such as Open Geospatial Consortium. The others are abbreviated due to space limitation. The topological subsumptions on the operators and multiple query optimizations on more than two spatial objects can be applied. We focused on the basic operators, and other operators such as spatial merge, windowing, clipping, map overlay, and spatial aggregation operators, are not considered here [14].

4.2 Integrated Query Operator

Now we can consider the integration of spatial objects with temporal and/or relational objects. We assume that those elements s_i, r_i, and t_i are orthogonal and are connected with the three subspaces S, Â, and Y. Note that this paper uses the conventional relational algebra, temporal notations by [8, 9], spatial and spatio-temporal representations by [1, 12, 13].In this paper, we assume that the multidimensional joins can be done through the fact table that has a spatial, a temporal and relational dimension(s) of the data warehouse.

Spatial Relational Join: The spatial relational theta (θ) join is formally defined as:{<s_i, r_i, t_i> □ (x, *y* | x = s_i.attribute θ Fact.attribute, y = r_i.attribute θ Fact.attribute); s_i∈Σ, r_i∈ℜ, t_i∈Ψ}

Spatial Temporal Join: The spatial temporal join is formally represented as: {<s_i, r_i, t_i>| □ (s_i.attribute θ Fact.attribute, t_i.attribute θ Fact.attribute); s_i∈Σ, r_i∈ℜ, t_i∈Ψ}

Spatial Temporal Relational Join: According to the above definitions, an integrated form of Spatial Temporal Relational Join can be suggested as follows.

5 EXPERIMENT

The cost model decomposes a spatiotemporal query into several atomic formulas computing cost of each atomic formula such as the three query transformation rules that detect certain materialized views. We follow the cost minimization algorithm of view maintenance from Corral et al. [15], Leung and Lee [16] and Theodoridis et al. [17].

The map objects, base relations, and the relevant files are assumed to be located in distributed sites and the corresponding materialized views are located at another site. The cost functions and parameters with their explanations in terms of the view materialization are represented as follows:

Table 1. Parameters used in cost functions

Term	Descriptions	Size
X	Variable that includes object or base relation (ex: *R, S*), materialized view (ex: *V*), etc.	Varied
¡ Ã,¡ Ã	Insertion and deletion of X respectively	Varied
B	Page size	*4000 bytes*
$C_{I/O}$	I/O cost	ms/block
C_{comm}	Transmission cost	bits/s
α_s	Selectivity factor	0.5
$\Phi[n, m, k]$	Cost that accesses *k* records in a file of *n* records stored in *m* pages	Varied [*16*]
N[*X*]	Cardinality of tuples of *X* per page	B/W_S
W[*X*]	Width of *X*	*200 bytes*
$H_B[X]$	Height of the index of *X*	$Log_{B/W}[N[X]]$-1

VIO1 = Cost of reading the materialized file of *S* (i.e., △*S* and ▽*S*) and sorting it
$= C_{I/O}*(N[\nabla S]*W[S.key] + N[\triangle S]*W[S]+ N[\triangle S]*W[S]*Log_B(N[\triangle S]*W[S]))/B$

VIO2 = Cost of reading the *differential data* of materialized file *R* (i.e., △*R* and ▽*R*) and sorting it
$= C_{I/O}*(N[\nabla R]*W[R.key] + N[\triangle R]*W[R] + N[\triangle R]*W[R] *Log_B(N[\triangle R]*W[R]))/B$

VIO2 = Cost of reading materialized file and sorting it
$= C_{I/O}*(N[\alpha_s *N(R)]*W[R] + N[\tilde{}_s *N(R)]*W[R] *Log_B(N[\alpha_s *N(R)]*W[R]))/B$

VIO3 = Cost of creating materialized file and maintaining it
$= C_{I/O}*\Phi[N(V), N(V)* W[R]/B, N[R]]/B$

VIO4 = Cost of accessing the index tree of the view and reading the view table
$= C_{I/O}*\{(H_B[V]-1)+\Phi[N(V), N[V* W[R]/B], N[\triangle S+\nabla S+\nabla R+ N(R)]]\}/B$

VCOM1 = Cost of transmitting the materialized file of *S* to the view
$= 8*N[\triangle S] * W[S]/C_{comm}+ 8*N[\nabla S.key]*W[S.key])/C_{comm}$

VCOM2 = Cost of transmitting the materialized file *R* to the view
$= 8*N[\nabla R.key]*W[R.key] + N[\triangle R]* W[R]/C_{comm}$

VCOM3 = Cost of transmitting the joined materialized file to the view
$= 8*N[\alpha_s *N(R)]* W[R]/C_{comm}$

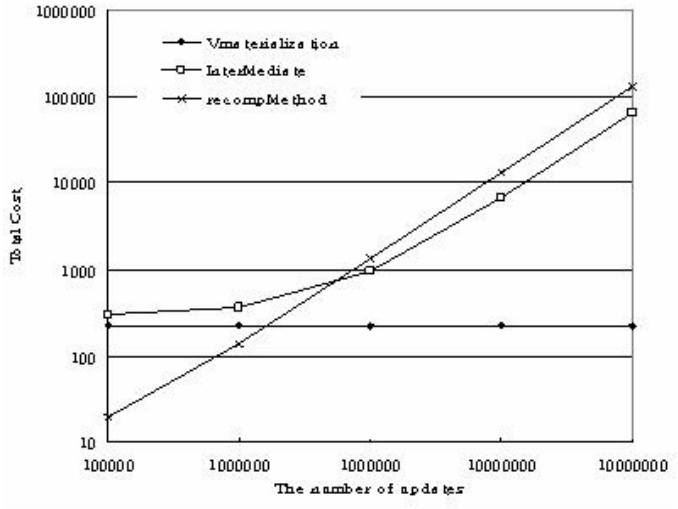
Then, the total cost of the view materialization method is the summation of the above.

We illustrate the effectiveness of our view materialization method by comparing the results of the following three implemented methods:

- The basic re-computation method (*RecompMethod*), which reads and sorts the base relation and objects when every update happens.
- The intermediate method (*Intermediate*), which is similar to the *RecompMethod* except that the method uses old views, differential files, and source data.
- Our view materialization method (*Vmaterialization*), which uses only old views, and differential files which means that this method avoids accessing source relations.

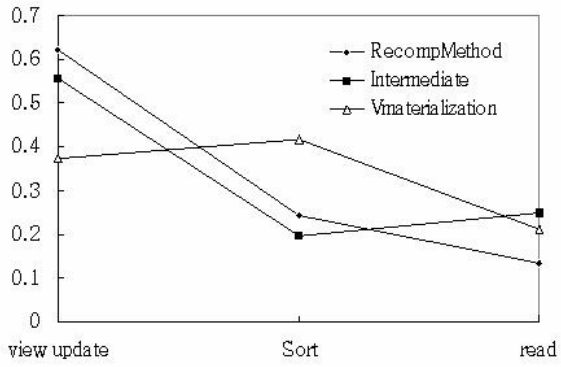
The experimental results are represented in Figures 4(a) and 4(b). The parameters of the experiments are extracted from the conditions

Figure 4(a). Total cost analysis with respect to the number of data updates, and (b) showing detailed cost ratio



(a)

Figure 4(a). Total cost analysis with respect to the number of data updates, and (b) showing detailed cost ratio



(b)

described in Table 1 such that if the selectivity equals 0.02, the communication speed is 100,000 bps, and the view size is fixed at 1 mega byte. The Fig. 4(a) indicates that the *Vmaterialization* is superior to the other methods in a large update environment of up to 10 gigabytes. The x-axis of Fig. 4(a) shows the ratio of updates; the y-axis, in logarithmic scale, shows the speedup of the improved method and our Vmaterialization method against the other methods. The trajectories illustrate that the total cost of the *Vmaterialization* remains the same even though the update size of objects increase tremendously. If the volume of the objects is large, the costs of the other methods diverge too much while the cost of the *Intermediate* is in the middle. In other words, we can expect that the *Vmaterialization* is appropriate to a data warehouse environment with its huge data volume. As for the quantity of objects, it can be concluded that the higher the number of updates in the data objects, the more advantageous the *Vmaterialization* method.

In more detail, we can see that the cost advantage of the *Vmaterialization* method mainly comes from the view update cost factor by which the data in the view can be stored as indicated in Figure 4(b). Even though we intentionally constrain the *Vmaterialization* method to additionally sort the updated data, the total cost gain still favors the view materialization effect.

6 RELATED WORK

It is common in GIS to make a distinction between space-based and feature-based models [11]. The former represents regions within the space. The latter represents geographic entities that are the primary objects with which spatial attributes are associated. Recently these two approaches were integrated [1]. The integrated approach has been adopted in this paper. Prior to integrating information, we derived a fundamental atomic data unit, called a triplet, by which the abstract information of an object and its inheritance are separated.

In order to generate a common framework for geospatial information, a number of researchers proposed approaches that appeared to represent a basic data model. Alternatives included an Extended Entity Relationship (EER) model [2, 6, 10], an annotation approach [3, 4], and a data warehouse approach [12]. Many conventional conceptual models, such as the ER model, Extended Entity Relationship model (EER) and starER [12] could potentially serve as a basis for representing a geospatial data warehouse model.

The starER was considered as a viable conceptual model for a data warehouse, but has several limitations as follows. The interface of spatial dimension and its associated fact table is represented by just an object ID (e.g., real estate 'number') [17]. To this effect, a hierarchical region and a city object are linked without a schema. This example fitted the description of a small data mart and did not embrace the essence of a spatial data warehouse. Tryfona et al., [12] in this model did not show a spatial schema but maintained the word 'star' while sacrificing the true meaning of data representation by including cyclic relationships or alternative relationships between the corresponding objects. There is neither a schema of spatial objects nor spatial operators accessing a spatial point corresponding to relational objects and vice versa. Therefore, in summary, their representation is that of a relational star schema, with just a dimension that references spatial names (e.g., city, region, county).

7 CONCLUSIONS

The objectives of this paper were to develop an architectural framework for integrating geospatial and temporal data with relational information from which a geospatial data warehouse (GSDW) was successfully built and implemented.

The data warehouse is considered an architectural framework for the following reasons: (1) it maintains both current and historical data; (2) it is scalable to handle a sheer volume of data that may be collected from various data sources and users; and (3) it has high performance and is cost effective. Maintaining both current and historic data in the same database structure requires a non-conventional spatially enabled object-relational data warehouse that integrates spatial, temporal, and relational objects.

In this paper, we have developed a model for forestry management that can be applied to different management operations. The model has a fact table with five dimensions. Using the basic model, an interface has been developed that creates links to spatial, temporal and relational operators. This interface is simple and yet powerful enough to organize and maintain the integrity of the data in the GSDW. This model has been successfully implemented using data from the Malcolm Knapp Research Forest of the University of British Columbia.

REFERENCES

1. Hadzilacos, T., Tryfona, T., An Extended Entity-Relationship Model for Geographic Applications, SIGMOD Record, (1997), 26(3): 24-29.
2. Hezzah, A., Tjoa, A.: Design and Representation of the Time Dimension in Enterprise Data Warehouses - A Business Related Practical Approach. ICEIS (1) (2004), 416-424
3. Khatri, V., Ram, S., Snodgrass, R.T.: Augmenting a Conceptual Model with Geospatiotemporal Annotations, IEEE TKDE. 16(11) (2004) 1324-1338.
4. Marshall, C., Toward an Ecology of Hypertext Annotation. Hypertext, (1998), 40-49.
5. Papadias, D., and Arkoumanis, D., Approximate Processing of Multiway Spatial Joins in Very Large Databases, EDBT, (2002), 179-196.
6. Shekhar, S., Coyle, M., Goyal, B., Liu, D., Sarkar, S., Data Models in Geographic Information Systems, CACM, (1997), 40(4): 103-111.
7. Raza, A., Kainsz, W., Cell Tuple Based Spatio-Temporal Data Model: An Object Oriented Approach, In Proc, ACM GIS, (1999), 20-25.
8. Slivinskas, G., Jensen, C., Snodgrass, R., A Foundation for Conventional and Temporal Query Optimization Addressing Duplicates and Ordering, TKDE, (2001), 13(1): 21-49.
9. Snodgrass, R., An Overview of TQuel. Temporal Databases, (1993), 141-182.
10. Sudarsky, S., Hjelsvold, R., Visualizing Electronic Mail, In Proc, IV (2002), 3-9.
11. Trujillo, J., Luján-Mora, S., Song, I.: Applying UML and XML for Designing and Interchanging Information for Data Warehouses and OLAP Applications. Journal of Database Management. 15(1): (2004) 41-72.
12. Tryfona, N., Busborg, F., Christiansen, J., starER: A Conceptual Model for Data Warehouse Design, In Proc, DOLAP, (1999), 3-8.

13. Tryfona, N., Hadzilacos, T., Logical Data Modeling of Spatio Temporal Applications: Definitions and a Model, In Proc, IDEAS, (1998), 14-23.
14. Voisard, A and David, B., A Database Perspective on Geospatial Data Modeling, TKDE, (2002), 14(2): 226-243.
15. Corral, A., Manolopoulos, Y., Theodoridis, Y., and Vassilakopoulos, M.: Multi-Way Distance Join Queries in Spatial Databases. GeoInformatica 8(4): (2004) 373-402
16. Leung, C., and Lee, W.: Exploitation of Referential Integrity Constraints for Efficient Update of Data Warehouse Views. BNCOD (2005), 98-110
17. Theodoridis, Y., Stefanakis, E., and Sellis, T.: Efficient Cost Models for Spatial Queries Using R-Trees. IEEE TKDE. 12(1): (2000) 19-32