EDH: A Powerful method for the modeling of Irregular Dimensions

Zehai Li[1, 2],Jigui Sun[1, 2],Haihong,Yu[1, 2]

[1]College of Computer Science and Technology, Jilin University, 130012 Changchun, China
[2]School Key Laboratory of Symbolic Computation and Knowledge Engineer of Ministry of Education, 130012 Changchun, China

Summary
Pre-aggregation is one of the most effective techniques in OLAP to ensure quick responses to user queries in data warehouses. Partial pre-aggregation is used widely, and the technique usually requires that the hierarchies of a dimension are onto, covering and self-onto. However, in real-world applications, many irregular dimensions do not meet the requirements. To solve the problem, in this paper, we propose a new dimension model to support the modeling of irregular dimensions based on extended directed Hasse graph. The model can express various relationships among dimension levels and different relationships among elements in the domain of a dimension. The model can also avoid aggregation loops in a dimension and ensure correct aggregation paths in the dimension by controlling the directions of paths and disallowing isolated nodes in EDH graphs. Thus, it can support both regular and irregular dimensions.

Key words:
EDH, Irregular dimension, OLAP

1. Introduction

OLAP (On-Line Analytical Processing) systems support decision-making processes by providing dynamic analytical operations on high volumes of data. These operations are performed on the data with the help of a group of dimensions. Pre-aggregation is one of the most effective techniques in OLAP to ensure quick response to user queries in data warehouses. The technique pre-aggregates and stores some (or all) results of the queries on original data.

The fastest response times may be achieved when materializing the results of all possible aggregate queries in advance, i.e., full pre-aggregation. However, the required storage space grows rapidly, and become prohibitive, as the complexity of the application increases. This phenomenon is called data explosion. Another problem with full pre-aggregation is that it takes too long to update the materialized aggregates when base data changes. With the goal of avoiding data explosion, research has focused on how to select the best subset of aggregation levels given space constraints [1] or maintenance time constraints [2]. The approach is commonly referred to as partial pre-aggregation [2]. The premise underlying the applicability of partial pre-aggregation is that lower-level aggregates can be reused to compute higher-level aggregates, known as summarizability [3]. Usually, summarizability requires the dimension hierarchies to be onto, covering, and self-onto [4,5]. However, in real-world applications, many irregular dimensions do not meet the requirements; and we call them as irregular dimensions, which can be non-onto, non-covering, or self-into [4,5]. There are two ways to deal with irregular dimensions. One way is to propose an effective multidimensional model to support these dimensions, and also the cubes with irregular dimensions and the OLAP operations on them. The models can be found in [5,6,7]. Another way is to devise some algorithms to transform irregular dimensions into well-formed ones that can be used in most multidimensional models. In this paper, we will mainly concern proposing a new powerful model to support the modeling of irregular dimensions.

The modeling of irregular dimensions is much more difficult than that of regular ones, because the complicated relationships among dimension levels are hardly described with simple mappings on the set of levels. Moreover, there might be aggregation loops in irregular dimensions, which will lead to error aggregation results; therefore, we should also control aggregation loops in dimensions.

In addition, most multidimensional models [6~14] describe the relationships among levels only with a single partial order. However, this modeling method would lose some semantics of a dimension because of the exclusive character of a set.

Recently, some research work [4,7,8,9,10,11,12,15] has been done to solve the problem. In [4], Pedersen proposes a model mainly to deal with many-to-many relationships between facts and dimensions; and he also concerns the modeling of irregular dimensions. However, the model will lose partial semantics when describing dimensions with Hasse diagram, and Pedersen did not mention the problems caused by aggregation loops.

In [11], Jensen presents a new approach to model partial containment relationships among the spatial objects

Manuscript revised January 2006.

in spatial dimensions; however, Jensen did not mention whether the model can support irregular dimensions and whether the model can control aggregation loops.

In [13], Tapio proposed a new formal form for OLAP cube design, and some decomposition algorithms to produce normalized OLAP cube schemata, and then it could control the structural sparsity resulting from inter-dimensional functional dependencies. However, the normalization algorithms are mainly used to control the sparsity that is mostly caused by non-normalized relationships between dimensions and facts; moreover, Tapio could not concern the possible non-normalization caused by the complex inner structure of a dimension; therefore, there are no further discussions in their work about the normalization of other irregular dimensions.

In this paper, we proposed a new technique called extended directed Hasse graph (EDH) to describe structure of special posets. The technique uses both original relationships and the extended partial order to describe the structure of a poset; and thus this avoids the problem when processing direct mappings and transitive relationships. Based on EDH, we propose a new dimension model to support irregular dimensions. Not only can the model express various relationships among the dimension levels of a dimension and the relationships among the elements in the domain of the dimension, but also it can avoid aggregation loops by controlling isolated nodes and the directions of paths. Therefore, the model can support both regular dimensions and irregular dimensions.

The rest of the paper is organized as follows. We define the EDH technique and propose a new dimension model in section 2. A cube model is defined in section 3. In section 4, we mainly define two major OLAP operations based on the EDH dimension model. Finally, section 5 concludes the paper.

2. Dimension model

In this section, we extend the way that Hasse diagram uses to describe the structure of a poset, which is called extended directed Hasse graph (EDH). EDH describes the structure of a poset based on both the partial order and the corresponding original relationships on the poset. On the basis of EDH, we propose a new multi-dimensional model, which can express the loop structure of an irregular dimension and disallow directed loops in a dimension by imposing restrictions on the direction of an arc. Therefore, the model can ensure correct aggregation paths in a dimension. Compared with those models in [6~10], the model can support the covering, onto dimensions and the non-covering, non-onto, self-into dimensions.

Definition 1. Given $D=\{l_1,l_2,\ldots,l_n\}$, where l_i is called the attribute of set A_i and $dom(l_i)=A_i$ $(1\leq i\leq n)$. For any l_i, $l_j\in D$ $(1\leq i,j\leq n)$, if there is a mapping from $dom(l_i)$ to $dom(l_j)$, then we define it as f: $dom(l_i)\rightarrow dom(l_j)$. If there is an attribute $l_k\in D$ such that there are two mappings g: $dom(l_i)\rightarrow dom(l_k)$ and h: $dom(l_k)\rightarrow dom(l_j)$; for any $a\in D_f$, $c\in R_f$, where D_f is the domain of f , R_f is the range of f, if there is an element $b\in dom(l_k)$ such that $b=g(a)$ and $c=h(b)$, then we call f as the transitive mapping from $dom(l_i)$ to $dom(l_j)$; otherwise, we call f as the direct mapping from $dom(l_i)$ to $dom(l_j)$.

Definition 2. Given $D=\{l_1,l_2,\ldots,l_n\}$ is a set of attributes, where $dom(l_i)=A_i$ $(1\leq i\leq n)$. Let $\prec_{pp}, \prec_{pf}, \prec_{fp}, \prec_{ff}$ be four bi-relationships on D; for any l_i, $l_j\in D$ $(1\leq i,j\leq n)$, if there is a direct mapping f: $dom(l_i)\rightarrow dom(l_j)$, then, (1) If f is not everywhere defined and onto, then $l_i\prec_{pp} l_j$, (2) If f is not everywhere defined and onto, then $l_i\prec_{pf} l_j$, (3) If f is everywhere defined and non-onto, then $l_i\prec_{fp} l_j$, (4) If f is everywhere defined and onto, then $l_i\prec_{ff} l_j$.

Definition 3. Given four relationships on D: $\prec_{pp}, \prec_{pf}, \prec_{fp}, \prec_{ff}$, let $\prec=\prec_{pp}\cup\prec_{pf}\cup\prec_{fp}\cup\prec_{ff}$, $\prec'_{pp}=\prec_{pp}$, $\prec'_{pf}=\prec_{pf}$, $\prec'_{fp}=\prec_{fp}$, $\prec'_{ff}=\prec_{ff}$ and $\prec''=\prec'_{pp}\cup\prec'_{pf}\cup\prec'_{fp}\cup\prec'_{ff}$. For any l_i, $l_j\in D$ $(1\leq i,j\leq n$, $i\neq j)$, if $(l_i,l_j)\in\prec_{pp}\cup\prec_{pf}\cup\prec_{fp}\cup\prec_{ff}$ and $(l_j, l_i)\in\prec_{pp}\cup\prec_{pf}\cup\prec_{fp}\cup\prec_{ff}$, then $l_j=l_i$. We extend $\prec'_{pp}$, $\prec'_{pf}$, $\prec'_{fp}$ and $\prec'_{ff}$ according to the following rules.

(1) For any $l_i\in D$, if $(l_i, l_i)\notin\prec_{pp}\cup\prec_{pf}\cup\prec_{fp}\cup\prec_{ff}$, then let $\prec'_{ff}=\prec'_{ff}\cup\{(l_i, l_i)\}$.

(2) We extend $\prec'_{pp}$, $\prec'_{pf}$, $\prec'_{fp}$, $\prec'_{ff}$ according to the following transitive rules:

Rule 1: if $l_i\prec_{pp} l_j$, $l_j\prec_{pp} l_k$, $l_i\prec l_k$ and there is no other level $l_p\in D(p\notin\{i,j,k\})$ such that $l_i\prec l_p$, then $l_i\prec'_{ff} l_k$.

Rule 1': if $l_i\prec_{pp} l_j$, $l_j\prec_{pp} l_k$ and $(l_i,l_k)\notin\prec_{pp}$, then $l_i\prec'_{pp} l_k$.

Rule 2: if $l_i\prec_{pp} l_j$, $l_j\prec_{pf} l_k$, $l_i\prec l_k$ and there is no other level $l_p\in D(p\notin\{i,j,k\})$ such that $l_i\prec l_p$, then $l_i\prec'_{ff} l_k$.

Rule 2': if there are only $l_i \prec_{pp} l_j$ and $l_j \prec_{pf} l_k$, then $l_i \prec'_{pf} l_k$.

Rule 3: if $l_i \prec_{pp} l_j$, $l_j \prec_{fp} l_k$, then $l_i \prec'_{pp} l_k$.

Rule 3': Given $l_i \prec_{pp} l_j$, $l_j \prec_{fp} l_k$, $l_i \prec l_k$ and there is no other level $l_p \in D(p \notin \{i,j,k\})$ such that $l_i \prec l_p$, if there is no other level $l_n \in D(n \notin \{i,j,k\})$ such that $l_n \prec l_k$, then $l_i \prec'_{ff} l_k$, otherwise $l_i \prec'_{fp} l_k$.

Rule 4: if $l_i \prec_{pp} l_j$, $l_j \prec_{ff} l_k$, then $l_i \prec'_{pf} l_k$.

Rule 5: if $l_i \prec_{pf} l_j$, $l_j \prec_{pp} l_k$, then $l_i \prec'_{pp} l_k$.

Rule 6: if $l_i \prec_{pf} l_j$, $l_j \prec_{pf} l_k$, then $l_i \prec'_{pf} l_k$.

Rule 7: if $l_i \prec_{pf} l_j$, $l_j \prec_{fp} l_k$, then $l_i \prec'_{pp} l_k$.

Rule 8: Given $l_i \prec_{pf} l_j$, $l_j \prec_{ff} l_k$, $l_i \prec l_k$, if there is no other level $l_p \in D(p \notin \{i,j,k\})$ such that $l_i \prec l_p$, then $l_i \prec'_{ff} l_k$, otherwise $l_i \prec'_{pf} l_k$.

Rule 9: Given $l_i \prec_{fp} l_j$, $l_j \prec_{pp} l_k$, if $l_i \prec l_k$ and there is no other level $l_p \in D(p \notin \{i,j,k\})$ such that $l_i \prec l_p$, then $l_i \prec'_{fp} l_k$, otherwise $l_i \prec'_{pp} l_k$.

Rule 10: Given $l_i \prec_{fp} l_j$, $l_j \prec_{pf} l_k$, if $l_i \prec_{pp} l_k$, then $l_i \prec'_{pf} l_k$, otherwise $l_i \prec'_{pp} l_k$.

Rule 11: If $l_i \prec_{fp} l_j$, $l_j \prec_{fp} l_k$, then $l_i \prec'_{fp} l_k$.

Rule 12: If $l_i \prec_{fp} l_j$, $l_j \prec_{ff} l_k$, then $l_i \prec'_{ff} l_k$.

Rule 13: If $l_i \prec_{ff} l_j$, $l_j \prec_{pf} l_k$, then $l_i \prec'_{pf} l_k$.

Rule 14: If $l_i \prec_{ff} l_j$, $l_j \prec_{pp} l_k$, then $l_i \prec'_{pp} l_k$.

Rule 15: If $l_i \prec_{ff} l_j$, $l_j \prec_{fp} l_k$, then $l_i \prec'_{fp} l_k$.

Rule 16: If $l_i \prec_{ff} l_j$, $l_j \prec_{ff} l_k$, then $l_i \prec'_{ff} l_k$.

Let $\prec'$ be the transitive and reflexive closure of $\prec''$ gotten by the rules in (1) and (2), then $(D, \prec')$ is a poset. The four relationships $\prec_{fp}$, $\prec_{ff}$, $\prec_{pp}$, $\prec_{pf}$ are called the original aggregation relationships on D. The relationships $\prec'_{pp}$, $\prec'_{pf}$, $\prec'_{fp}$ and $\prec'_{ff}$ processed after closure operations are called the extended aggregation relationships on D, and $\prec'$ is called call the aggregation partial order on D. Currently, there are two notation methods for a poset: Hasse diagram and digraph.

For a poset $(D, \prec')$, the corresponding digraph uses arcs to connect any two elements which satisfy the partial order; therefore, the digraph can explicitly express each relationship between any two elements. However, for the sakes of using reflective arcs and transitive arcs explicitly in the digraph, the structure of the resulted digraph is very complicated and hard to read and process for us.

The way of using a Hasse diagram to describe the structure of a poset is much simpler than that of using a digraph, because reflective edges and transitive edges are ignored in the Hasse diagram. However, a Hasse diagram has some obvious disadvantages. (1) the Hasse diagram is an undirected graph and vertices in an undirected graph is position-irrelative; moreover, an edge could not denote whether an vertex is higher (or lower) than another vertex in the graph. (2) An edge in the Hasse diagram might make us to misunderstand that the connected elements satisfy a symmetrical relationship; obviously, this is wrong and it might form a loop in the graph. (3) If it is used to describe irregular dimensions, a Hasse diagram would lost some semantics of the dimension. If we use Hasse diagram to describe the structure of the depot dimension in fig.1, which is shown in fig.3, then the figure will ignore the edge (GrainDepot, ParentCompany) based on the transitivity of the partial order. As we can see, it cannot fully express the non-covering mapping between "SubCompany" and "ParentCompany" w.r.t "GrainDepot".

To solve the problems, we propose a new notation method for the special posets defined in definition 1, which combines the advantages of both digraphs and Hasse diagrams, and avoids their disadvantages. Given a poset $(D, \prec')$ and the original aggregation relationships $\prec_{pp}, \prec_{pf}, \prec_{fp}$ and $\prec_{ff}$; let $\prec'_{pp}$, $\prec'_{pf}$, $\prec'_{fp}$, $\prec'_{ff}$ be the extended relationships of $\prec_{pp}$, $\prec_{pf}$, $\prec_{fp}$, $\prec_{ff}$ respectively, and $\prec' = \prec'_{pp} \cup \prec'_{pf} \cup \prec'_{fp} \cup \prec'_{ff}$, then we describe the structure of poset $(D, \prec')$ according to the following steps. For any $l_i, l_j \in D$ $(1 \le i,j \le n, i \ne j)$,

(1) If $l_i \prec' l_j$ and there is no middle level $l_k \in D(1 \le k \le n, k \ne i, k \ne j)$ such that $l_i \prec' l_k$ and $l_k \prec' l_j$, then we put l_i under l_j, and draw an arc from l_i to l_j according to the following rules (the arcs are called the 1st arc, 2nd arc, 3rd arc and 4th arc, as shown in fig.2).

(2) If $l_i \prec' l_j$ and there is an level $l_k \in D$ $(1 \leqslant k \leqslant n, k \ne i, k \ne j)$ such that $l_i \prec' l_k$ and $l_k \prec' l_j$, then if $(l_i, l_j) \in \prec_{pp} \cup \prec_{pf} \cup \prec_{fp} \cup \prec_{ff}$, then we draw an arc from to according to (1), otherwise, draw no arc.

(3) If $(l_i, l_i) \in \prec'_{pp}$, draw a 4th arc from l_i to itself.

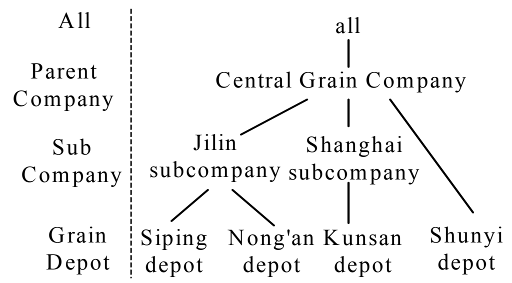
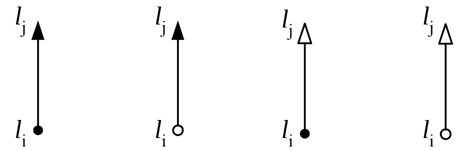


Fig.1. Depot dimension and it's partial member structure



(a) $l_i \prec'_{ff} l_j$ (b) $l_i \prec'_{pf} l_j$ (c) $l_i \prec'_{fp} l_j$ (d) $l_i \prec'_{pp} l_j$

Fig.2. Four types of arcs

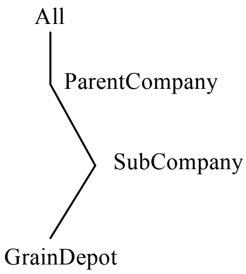


Fig.3. The Hasse diagram

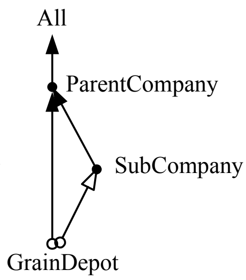


Fig.4. The EDH

The graph obtained according to above method is called Extended Directed Hasse Graph (EDH). For example, we use EDH to describe the structure of the depot dimension, which is shown in fig.4. In the figure, although there are already two arcs (GrainDepot, SubComany) and (SubComany, ParentComany), we need still draw a 2nd arc from “GrainDepot” to “ParentCompany” because (GrainDepot, ParentComany) $\in \prec_{pf}$.

Given an EDH graph G, let *v* be a vertex in G. For any vertex *v*’(*v*’≠*v*) in G, if there is always a directed path from *v*’ to *v*, then *v* is a root node in G. If there is no any directed path ending at *v* in G, then *v* is a leaf node in G. Otherwise, *v* is a middle node.

Definition 4. Given a poset (D, $\prec'$), and $\prec_{pp}$, $\prec_{pf}$, $\prec_{fp}$, $\prec_{ff}$ are the original aggregation relationships; let $\prec'_{pp}$, $\prec'_{pf}$, $\prec'_{fp}$, $\prec'_{ff}$ be the extended relationships of $\prec_{pp}$, $\prec_{pf}$, $\prec_{fp}$, $\prec_{ff}$ respectively, and $\prec' = \prec'_{pp} \cup \prec'_{pf} \cup \prec'_{fp} \cup \prec'_{ff}$ be the aggregation relationship on D, where D={$l_1,l_2,\ldots,l_n$}. Let G be the EDH graph of (D, $\prec'$), if G satisfies the following three conditions, then (D, $\prec'$) is called a dimension schema, written d=(D, $\prec'$), and G is the schema structure graph (SSG) of dimension d; moreover, let dom(d)= $\bigcup_{1\le i\le n}$ dom(l_i).

(1) G is connected.

(2) There are at least two nodes in G.

(3) There is only one root node l_i(1≤i≤n) in G, written *All*[8], and one single leaf node l_j(1≤j≤n, i≠j), written Atomic, where dom(*All*)={all}.

Given a dimension d and its EDH graph G, a path from *Atomic* to *All* is called a hierarchy of d. Dimension schema defines the basic structure of a dimension, namely the organization form of the elements in the dimension domain. On this basis, we define the partial order on the dimension domain, and also use an EDH graph to describe the structure of the dimension domain.

Definition 5. Given a poset (D, $\prec'$), and the aggregation relationship $\prec'$ on D, where D={$l_1,l_2,\ldots,l_n$}. Let E= $\bigcup_{1\le i\le k}$ dom(l_i), and $\le$ is a bi-relationship on E, if $\le$ satisfies the following conditions, then we call $\le$ as the element aggregation relationship (EAR) on E.

(1) Given $l_i,l_j \in$ D(1≤i,j≤n) such that $l_i \prec' l_j$, let σ_{ij} be the mapping from dom(l_i) to dom(l_j) corresponding to the pair (l_i,l_j), then for any a $\in$ dom(l_i) and b $\in$ dom(l_j), if (a,b) $\in \sigma_{ij}$, then a $\le$ b.

(2) Given $l_i,l_j \in$ D (1≤i,j≤n), for any a $\in$ dom(li) and b $\in$ dom(lj), if a $\le$ b, then $l_i \prec' l_j$.

(3) Given $l_i,l_j,l_k \in$ D(1≤i,j,k≤n), for any a $\in$ dom(l_i), b $\in$ dom(l_j) and c $\in$ dom(l_k), if a $\le$ b and b $\le$ c, then a $\le$ c.

Given two elements a and b, they could only have following three relationships: (1) a $\le$ b and b $\not\le$ a, (2) b $\le$ a and a $\not\le$ b, (3) a $\not\le$ b and b $\not\le$ a. Let $\le_{ff} = \le$, $\le'_{ff} = \le_{ff}$, $\le' = \le$, then we can use the EDH graph that only contain 1st arcs to describe the structure of poset (E, $\le'$). Moreover, we need not to check the sixteen rules in definition 2, because there is only one relationship $\le'_{ff}$ on E. We call the EDH of poset (E, $\le'$) as the element structure graph (ESG) of dimension d.

Based on the definitions of dimension schema and dimension structure graph, we will investigate the relationship between an irregular dimension and its SSG. Given a dimension d =(D, $\prec'$) and G=(V,A) is the SSG of d, then, (1) d is non-covering iff there are two nodes a, b $\in$ V (a≠b) such that there are two paths in G from a to b. (2) d is non-onto iff there is at least one 3rd arc or one 4th arc in G. (3) d is self-into iff there is one node a in G such that there is one 4th arc from a to itself.

3. Cube model

In this section, based on the above dimension model, we will present a cube model by combining the dimension model with various types of measures to support the modeling of the cubes with irregular dimensions.

We know that cubes are basic units in a data warehouse, and cubes are made of dimensions and measures, so cubes should be modeled along with the dimension model.

A measure structure is defined first, and then a cube model is defined.

Definition 6. Let m=(M,*agr*) be a measure structure, where M is a numeric set, and *agr* is a function: $2^M \rightarrow$ M, which is called the aggregation function on M. And let dom(m)=M, m is called the measure attribute.

Definition 7. A data cube is defined as a tri-tuple: C=(D,M,f), where,(1) D={d_1,…,d_n} is the set of all dimensions, and $d_i=(D_i, \prec'_i)$ ($1 \leqslant i \leqslant n$), (2) M={$m_1$,…,$m_k$} is the set of all measures, and $m_j=(M_j,agr_j)$ ($1 \leqslant j \leqslant k$), (3) let dom($D$)=dom($d_1$) × … × dom($d_n$), dom(M)=dom($m_1$) ×…×dom($m_k$), (4) f is a mapping: P→dom(M), where $P \subseteq$ dom(D) and $P \neq \phi$, (5) let dom(C)=P×dom(M).

P is a set of n-tuples, and the set of the element in the *i*th position of each tuple in P is a subset of dom(d_i), which is called the current domain of dimension d_i, written cdom(d_i). In addition, we should impose a restriction on the elements in the current domain of dimension d to ensure the structure of a cube is organized reasonably, which is described as follows: the current domain of dimension d_i must be the set of all minimal elements in dom(d_i) or a subset of one certain level's domain of d_i.

Given a cube C=(D,M,f), in fact, D and M define the schema of the cube C, written: schema(C)=(D,M), while f defines the domain of the cube C. For any non-null set P that satisfies the restriction, we could get a new cube which is different from C only in the domains, and they have the same schema. We call these cubes as the *base-common* cubes. In these cubes, there is a special cube, of which the set P is the Cartesian production of all minimal elements of every dimension's domain; and then, we call the cube as the *base cube* of the schema (D,M). Moreover, the base cube is unique for the schema (D,M). The *base cube* is formally defined as follows.

Definition 8. Let C_b=(D,M,f_b) be a cube of the schema (D,M), where D={d_1,…,d_n}, f_b:P→dom(M) and P⊆dom(D). For each tuple $x=(x_1,x_2,\ldots,x_n)\in P$, where $x_i\in$dom(d_i) ($1\le i\le n$), if x_i is a minimal element in dom(d_i), then we call C_b as the base cube of the schema (D,M).

Compared with the cube models in [6~10], definition 7 and definition 8 have broken the restriction that the elements in the current domain of each dimension of a cube must belong to the same dimension level's domain. Therefore, it provides a more agile way for cube modeling.

4. OLAP operations

For the cubes with irregular dimensions, the procedures of the operations, such as roll-up and drill-down, are much more complex than those on cubes with regular dimensions. Moreover, they cannot be expressed explicitly only with one formula. In this section, we use functions to describe these operations.

Fig.5 shows the procedure of the operation drill-down. In the figure, C, G1, G2, d_i and H_i are the input parameters of the function, where C is the current cube, d_i and H_i are the dimension and the hierarchy on which we perform drill-down operation, G1 is the ESG of the dimension d_i, G2 is the ESG of the dimension d_i. The output of the function is the final cube. For any element x in cdom(d_i), if there are no arcs in G1end at x, then the function does nothing and return C. Otherwise, if there are some elements in cdom(d_i) such that there are arcs in G1end at them and cdom(d_i) is exactly the domain of some level l_p, then the level l_p is the current level of dimension d_i. In addition, there is another node l_k in G2 such that the arc (l_k,l_p) belongs to H_i. If l_k is *Atomic*$_k$, then let s' be the set of all leaves of G1;otherwise, let s' =dom(l_k). We construct a cube C' =(D,M,f') according to s' and C, where f' defines P'=cdom(d_1)×…×cdom(d_{i-1})×s'×…×cdom(d_n), then C' is the final cube of operation drill-down.

Cube Function Drilldown(C, G1, G2, d_i, H_i)
(1) For $\forall x \in$ cdom(d_i), If there is no arc in ESG that end at x, then return C
(2) Otherwise, if there are some elements in cdom(d_i) satisfying that there are arcs in ESG which end at them, then there must be a level l_p in d_i such that cdom(d_i)=dom(l_p).
(3) And there is a vertex l_k in SSG such that $(l_k,l_p)\in$ Hi, if l_k is Atomic, then s'={all leaves in G1}
(4) If l_k is not Atomic, then s'=dom(l_k)
(5) Let the cube C' =(D,M,f'), where f' is a function: P'→dom(M), and P'=cdom(d_1)×…×cdom(d_{i-1})×s'×…×cdom(d_n).
(6) Return C'

Fig.5. Function Drilldown

Fig.6 shows the procedure of the operation roll-up. In the figure, the function has the same input parameters with Drilldown, and the output is the final cube. If there is an element x in cdom(d_i) that is the root of the ESG, then the function returns C directly; else if all the elements of cdom(d_i) are the leaves of the ESG, then *Atomic*$_i$ is the current level of dimension d_i; and there is one node l_{iq} in the SSG such that the arc (*Atomic*$_i$,l_{iq}) belongs to the hierarchy H_i; otherwise, cdom(d_i) must be the domain of some level l_{ip}, and there is an arc (l_{ip},l_{iq}) that belongs to the hierarchy H_i. Therefore, l_{iq} is the current level of dimension d_i after Rollup is performed. Following previous analyses, we construct a cube C' =(D,M,f'), where f' defines the set P'= cdom(d_1)×…×cdom(d_{i-1})×dom(l_{iq})×…×cdom(d_n), then C' is the final cube after Rollup is performed.

Cube Function Rollup(C, ESG, SSG, d_i, H_i)
(1) If $\exists\ \forall x \in$ cdom(d_i) such that x is the root of ESG, then return C.
(2) Else if elements in cdom(d_i) are the leaves of ESG, then there is an arc (*Atomic*$_i$,l_{iq}) in H_i.
(3) Else there is a level l_{ip} such that cdom(d_i)=dom(l_{ip}), and there is an arc (l_{ip},l_{iq}) in H_i.
(4) Then level l_{iq} is the resulting level and cdom(d_i)=dom(l_{iq}).

(5) Let the cube C' =(*D*,M,f'), where f' is a function: $P' \rightarrow dom(M)$, and $P' = cdom(d_1) \times \ldots \times cdom(d_{i-1}) \times dom(l_{iq}) \times \ldots \times cdom(d_n)$.
(6) Return C'

Fig.6. Function Rollup

5. Conclusion

In this paper, we extended Hasse diagram technique to describe the structure of a poset, and the extended technique is called extended directed Hasse(EDH) graph. EDH technique can describe the structure of a set according to the original four relationships we defined and the extended partial order on the set, and thus it can avoid losing partial semantics when describing transitive relationships among elements.

Based on EDH technique, we also proposed a new dimension model to support the modeling of both regular dimensions and irregular dimensions. The model can not only describe various relationships among levels and different relationships among elements, but also it can avoid aggregation loops in a dimension and ensure correct aggregation paths in the dimension by controlling the directions of paths and disallowing isolated nodes in EDH graphs.

Acknowledgment

This work is supported by National Natural Science Foundation of China (No. 60273080, 60473003).

References

[1] H. Gupta and I.S.Mumick. Selection of Views to Materialize Under a Maintenance-Time Constraint. In Proc. of ICDT, pp. 453–470, 1999.
[2] V. Harinarayan et al. Implementing Data Cubes Efficiently. In Proc. of SIGMOD, pp. 205–216, 1996.
[3] H. Lenz and A. Shoshani. Summarizability in OLAP and Statistical Data Bases. In Proc. of SSDBM, pp. 39–48, 1997.
[4] T.B.Pedersen, C.S.Jensen, C.E.Dyreson. Extending Practical Pre-Aggregation in On-Line Analytical Processing. In Proc of the 25th VLDB conference, pages 663-674, Morgan Kaufmann Publishers, 1999.
[5] Z.H. Li, J.G. Sun, et, al. Transforming Non-Covering dimensions in OLAP. In Proc. of 7th Asia-Pacific Web Int. Conf., Shanghai, China, 381~393. 2005
[6] A.Datta, H.Thomas. The cube data model: a conceptual model and algebra for on-line analytical processing in data warehouses. Decision Support Systems, 27(3): 289-301, 1999.
[7] T.B.Nguyen, A.M.Tjoa, R.Wagner. An Object Oriented Multidimensional Data Model for OLAP, In Proc. of the 1st Int. Conf. on Web-Age Information Management, pages 69-82, Springer-Verlag press, 2000.
[8]W.Lehner, J.Albrecht, H.Wedekind. Normal Forms for Multidimensional Databases. In Proc. of 10th Int. Conf. on Statistical and Scientific Database Management, pages 63-72, IEEE Computer Society press, 1998.
[9] H.V.Jagadish, L.V.S.Lakshmanan, D.Srivastava. What can Hierarchies do for Data Warehouses?. In Proc of the 25th VLDB conference, pages 530-541, Morgan Kaufmann Publishers, 1999.
[10] T.B.Pedersen, C.S.Jensen, C.E.Dyreson. A foundation for capturing and querying complex multidimensional data. Information System, 26(5): 383-423, 2001.
[11] Christian S. Jensen, Augustas Kligys, Torben Bach Pedersen, Igor Timko. Multidimensional data modeling for location-based services. The VLDB Journal (2004) 13, page:1–21.
[12] T B Pedersen, C S Jensen. Multidimensional Data Modeling for Complex Data. Proceedings of the 15th International Conference on Data Engineering.IEEE ComputerSociety, Washington DC USA (1999) 336-345.
[13]. N.Tapio, N.Jyrki, T.Peter. Normalising OLAP cubes for controlling sparsity. Data & Knowledge Engineering, 46: 317–343, 2003.

Zehai Li received the M.S. degree in Electrical Engineering from Jilin University in 2001, and received his Ph.D. degree in Computer Science from Jilin University in 2005.He is now a lecturer in the college of computer science and technology of Jilin University. His research interests include data warehouse, OLAP and decision support system.

Jigui Sun received the M.S. degree in Mathematics from Jilin University in 1988, and received Ph.D. degree in Computer Science from Jilin University in 1993. Since 1997, he has been a professor in the college of computer science & technology of Jilin University, and now he is also a Ph.D. supervisor. His research interests include artificial intelligence, constraint programming, and decision support system.

Haihong Yu received the M.S. degree in Computer Science from Jilin University in 2001, and now is a Ph.D. student in the college of computer science & technology of Jilin University. His research interest is decision support system.