データベースシステム 119−20
（１９９９．７．２１）

従属性を用いた多次元データベースの設計

沙　飛　　古川哲也

九州大学経済学部

大量の数値データを利用するために，OLAP やデータウェアハウスでは多次元データベースの利用が研究されている。本稿では，次元間の関係に注目した多次元データベースの設計について議論している。次元間に関数従属性があると，それは適当な多次元空間とはならない。独立した次元で多次元空間を構成し，関係集合でその他の次元との対応を表す多次元データモデルを提案し，関数従属性集合からデータベースを設計する手法を与えている。また，基本的な操作である次元選択と区間集約の処理法についても検討している。

Designing Multidimensional Databases with Dependencies

Fei Sha　　Tetsuya Furukawa

Dept. Economic Eng., Kyushu Univ.

In order to use a large amount of data, multidimensional databases are studied in OLAP and data warehouse areas. This paper discusses how to design multidimensional databases using the relationships among dimensions. Dimensions with functional dependencies do not make a proper multidimensional data space. A new multidimensional data model is proposed, whose space is organized by the independent dimensions with the relations keeping the correspondence with the other dimensions. The paper gives a design method of multidimensional databases using functional dependencies. It is also shown how the basic operations, selection of dimension and aggregation of section, are processed in the proposed model.

1. はじめに

業務のコンピュータ化やネットワークの発展にともない，大量のデータが様々な分野の様々な状況で発生しており，組織の活動にそれらを利用するようになった。利用するデータは組織内のものだけでなく，外部のデータを収集したものも含まれる。収集したデータを活用するためには，コンピュータによる管理が必要である。データは組織の様々な意思決定に用いられるため，様々な側面から解析できるようにデータベースを構成することが重要となる。企業の意思決定において，これらのデータを利用するために OLAP (On-Line Analytical Proccessing) に基づく支援システムが研究されている[1]。このようなシステムにおいてデータを管理するものは，データウェアハウスと呼ばれる[3]。

データウェアハウスに蓄えられるデータには，事実データと数値データがある。業務の分析を行う際，数値データの解析が重要な役割を果たしている。大量の数値データのデータベース化が各分野で行われており，数値データの解析に適したデータ構造は重要な課題になっている。そのために，データウェアハウスに関する研究では，多次元データベースの概念が提案されている。本稿では，多次元データベースの体系的な設計法について議論する。

数値データには，そのデータが何に対するものであるか，またどのような状況で発生したかについての説明がある。そのような説明は数値データの属性であると考えることができる。多次元データベースは，属性を次元としすることで n 個の属性に対して n 次元空間を構成し，その空間に数値データを配置する考え方である。多次元データベー

スの構築の際に最も重要な問題は，多次元空間をどのように構成するか，すなわち何を次元とするかである。単純に数値データのすべての属性を次元にするのは適当ではない。多次元データベースの処理は次元に対して行われるため，次元間の関係を考慮しなければ最適なデータベースの構築ができない。

例えば，売上のデータに対して，商品コードを値とする属性とその分類コードを値とする属性を独立した次元とするのは適当ではない。分類コードは商品コードを分割する属性であるため，同一の次元とみなすべきである。その際，商品コードと分類コードの対応関係を記憶する必要が生じる。これらの属性間の関係は，関数従属性として形式的に記述できる。属性間の従属性を用いることによって，最適な多次元データベースの設計を行うことが可能となる。本稿の目的は，与えられた関数従属性集合に対して，多次元データベースと次元に関する情報の記憶方式の設計手法を与えることにある。

多次元データベースの設計に関する研究には，データウェアハウスの物理的な実現に関するもの[2]から，モデルの形式化と拡張[4, 5]などがある。次元の決定は，鎖状の関数従属性を用いるものはあるが，次元間の関係はデータベース全体として議論されていない。本稿では，数値データに付随する多次元データベースの構造に関わる情報を全体的に検討している。

2. 多次元データベース

多次元データベースで対象とするデータは数値データである。数値データは多次元空間に配置され，その位置は各次元の値で示される。次元の値は数値が何に対するものかやどのうな状況で発生したものかを表す。したがって，多次元データベースのデータは，数値データと多次元空間の座標の値の組 $(n,(d_1,d_2,\cdots,d_n))$ の集合として考えることができる。データベースの構造を $MDB(N,\boldsymbol{D})$ で表す。Nは数値データであり，$\boldsymbol{D}=\{D_1,D_2,\cdots,D_n\}$ は多次元空間である。

多次元データベースにおける操作には，次の2つが必要である。

- 次元選択：必要な次元を抽出する。
- 区間集約：値を次元の区間ごとに集約する。

多次元データベースに対して，次元選択は重要かつ不可欠な操作である。数値データは様々な分析に用いられるため，数多くの性質で分類した詳細なデータとしておく必要がある。しかし，利用者は分析の対象とする次元のみが必要であるため，そのような次元の抽出を行わなければならない。次元選択は抽出する次元の座標が同じである値を集約することで行われる。

次元選択を不必要な次元の削除として考える。多次元データベース $MDB(N,\boldsymbol{D})$ から次元 Dを削除する。Dを削除することによって，$MDB(N,\boldsymbol{D})$ は $MDB'(N,\boldsymbol{D}-\{D\})$ となる。数値の集約は，

group-by($\boldsymbol{D}-\{D\}$)
for each group sum(N)

によって行われる。

[例1] 図1 (a) の多次元データベース $MDB(N,\{A,B,C\})$ から次元 Cを削除する。Cの削除のために，ABの値が同じデータの集約を行う。ABの値が (a_1,b_1) のデータは，Nの値が1と3の2つなので，集約の結果は ABの値 (a_1,b_1) に対して4である。ABの値が (a_1,b_2) のデータは，Nの値は5の1つのみであり，集約の結果は ABの値 (a_1,b_2) に対して5である。したがって，$MDB(N,\{A,B,C\})$ から Cを削除した結果は，図1 (b) となる。 □

N	A	B	C
1	a_1	b_1	c_1
3	a_1	b_1	c_2
5	a_1	b_2	c_2

(a)

N	A	B
4	a_1	b_1
5	a_1	b_2

(b)

図1 次元の削除

区間集約における次元の区間は，次元の値集合の分割である。次元 Dの値集合の分割を $S=\{s_1,s_2,\cdots,s_t\}$ とする。s_iは Dの値の集合であり，$1\leq i,j\leq t$ に対し $s_i\cap s_j=\emptyset$である。区間集約の結果は，$\boldsymbol{D}-\{D\}$ の値は等しく，Dの値は s_iに含まれるデータを集約したものである。例えば，毎日の売上のデータを週ごとに集約しさらに月ごとに集約するなどの処理は，日付に対する次元を区間で集約するものである。

3. 次元間の関数従属性

多次元データベース $MDB(N, \boldsymbol{D})$ の次元が独立していないとき，多次元データベースの構造は最適であるとはいえない。本節では，次元間に関数従属性があるときの問題点について検討する。

次元 D_i と D_j に対し，関数従属性 $D_i \rightarrow D_j$ が存在したとすると，D_i の各値に対して対応する D_j の値はただ1つしかない。このとき，多次元空間において数値データは，D_iD_j 平面では D_i の1つの値に対してただ1箇所に存在する。図2はそのような D_iD_j 平面であり，n_k で数値データが存在する位置を示している。このときの次元 D_j と D_i に対する操作は以下のようになる。

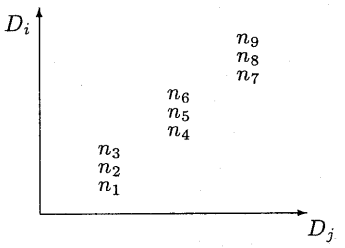

図2 $D_i \rightarrow D_j$ に対する D_iD_j 平面

次元選択で，D_j が選択される次元ではないために削除される場合を考える。次元 D_j の削除では，D_j を消去し，$\boldsymbol{D} - \{D_j\}$ で同じ値を持つデータを集約する。関数従属性 $D_i \rightarrow D_j$ が存在するので，D_i の超集合である $\boldsymbol{D} - \{D_j\}$ の各値に対し D_j の値は1つしか存在しないため，group-by($\boldsymbol{D} - \{D_j\}$) の各グループには要素は1つしかない。図3は D_j の削除を D_iD_j 平面について示したものである。D_i 軸上の n_k は D_j を削除したときの集約結果であり，集約前のデータと1対1に対応する。したがって，D_j の削除のための集約は必要がなく，D_j は多次元空間において空間の構成に関する役割を持っていない。

次に，次元 D_j に対する分割 S_j による区間集約を考える。同じ D_j の値に対応する D_i の値の集合を要素とする D_i の分割を S_i とする。また，S_i の要素を併合し，S_j の要素と1対1に対応するようにしたものを S_i' とする（図4 (a)）。D_j の S_j による区間集約で集約されるデータ集合と D_i の S_i' による区間集約で集約されるデータ集合は等しい。図4 (b) は，D_i の S_i' と D_j の S_j による区間集約を D_iD_j 平

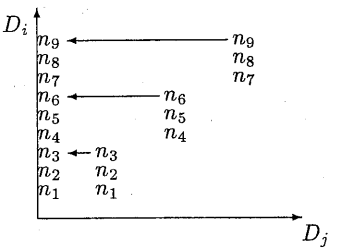

図3 次元 D_j の削除

面で示したものである。D_i 軸上の N_k は，S_i' による D_i に対する区間集約の結果，D_j 軸上の N_k は，S_j による D_j に対する区間集約の結果であり，対応する値は等しい。したがって，D_j に対する区間集約は，D_j を削除した多次元空間においても D_i に対する区間集約として求めることができる。

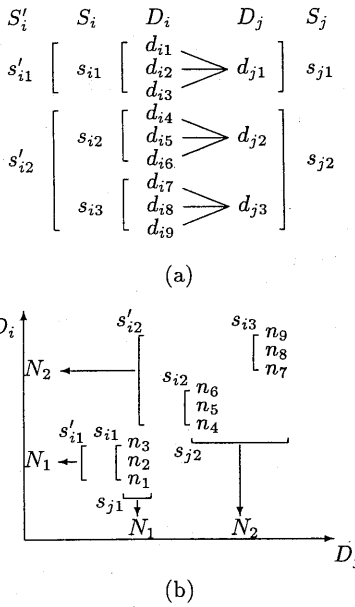

図4 次元 D_i の区間集約

一方，D_i の削除では，$\boldsymbol{D} - \{D_i\}$ が同じ値であるデータを集約する。D_j の1つの値に対し対応する D_i の値の集合は共通集合を持たないので，集約されるデータの D_i の値はすべて異なる。その区間は，D_j の値に対応する D_i の値集合である。すなわち，D_i の削除は D_i の区間選択で表すことができ

る。図5は，D_iの削除をD_iD_j平面で示したものである。D_j軸上のN_kはD_iを削除したときの集約結果であり，集約されるデータ集合はD_jの値で区間を定めたD_iの値集合である。D_i軸上のN_kはそのような区間によるD_iの区間選択の結果であり，対応するD_j軸上の値と等しい。したがって，D_jはD_iの区間を定める属性であると考えることができる。

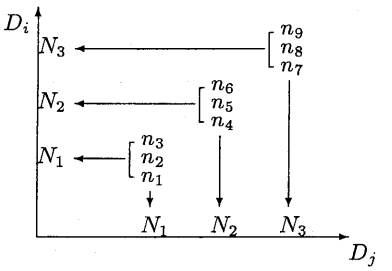

図5 次元D_iの削除

関数従属性の左辺が複数の次元からなる場合も同様に考えることができる。多次元データベース$MDB(N, \boldsymbol{D})$の次元間に，関数従属性$D_iD_j \rightarrow D_k$が存在したとする。次元選択でD_kが必要ではないために削除される場合，D_kを削除し$\boldsymbol{D}-\{D_k\}$で同じ値を持つデータを集約する。$\boldsymbol{D}-\{D_k\}$の各値に対しD_iD_jの値は1つしか存在しないため，group-by($\boldsymbol{D}-\{D_k\}$)の各グループには要素は1つしかない。したがって，D_kの削除のための集約は必要ない。

D_kの値はD_iD_jの値集合を決定し，その値集合間の共通集合は空集合なので，D_kはD_iD_jの区間（D_iD_j平面における領域）を定める属性であると考えることができる。

多次元データベース$MDB(N, \boldsymbol{D})$で，関数従属性$X \rightarrow Y$が成り立つとき，Yに含まれる次元は多次元空間の構成に関する役割を持っていないので$\boldsymbol{D}$から削除できる。$MDB(N, \boldsymbol{D})$を$MDB(N, \boldsymbol{D}-Y)$とすることを関数従属性$X \rightarrow Y$による多次元空間の正規化という。

4. 多次元データモデル

関数従属性の右辺の次元は，多次元データベースでは次元とする必要がなく，他の次元あるいは次元集合の区間や領域を定める属性と考えることができる。本節では，与えられた関数従属性集合から多次元データベースの次元を決定する方法について議論し，関数従属性の右辺の次元を多次元空間から分離したモデルについて検討する。

最適な多次元データベースの基準には次のものが考えられる。

- 次元数が少ない。

利用者にとってすべての次元が必要というわけではなく，分析の対象となる次元だけが利用できればよい。次元が必要以上に多くなると，次元の関係や数値データの持つ意味が複雑になり，理解しづらい。次元数が少ないほど利用者にとって理解しやすい多次元空間となる。

- 必要なデータを得ることができる。

利用者は，常に定まった次元に対する分析を行うわけではなく，さらに詳細な分析を行うときには，任意の次元が利用できなければならない。また，次元選択や区間集約によりデータを集約していても，データの詳細化が行えることが必要である。すなわち，データベース化するデータの情報は失われてはならない。

- 次元間の関係が理解しやすい。

次元間の関係を理解しやすくするために，独立でない次元が他の次元と同様に多次元空間を構成するよりは，次元間の従属性を把握できる構成の方がよい。ある次元が他の次元の区間を定めるものであれば，それがわかる構造になっている方が望ましい。

関数従属性による多次元空間の正規化は，多次元空間の構成に関する役割を持っていない次元を削除することで多次元データベースの次元数を減らすものである。正規化により次元数の少ない多次元データベースを作ることができる。

正規化によって削除された次元の持つ情報を失わないようにする必要がある。そのために，関数従属性の左辺と右辺の対応を記憶する。多次元データベース$MDB(N, \boldsymbol{D})$で，関数従属性$X \rightarrow Y$によって正規化したデータベース$MDB(N, \boldsymbol{D}-\{Y\})$では，$X$と$Y$の対応が必要である。$XY$を関係モデルにおける関係として考える。$MDB(N, \boldsymbol{D})$と$\{MDB(N, \boldsymbol{D}-\{Y\}), XY\}$を比較すると，正規化されたデータベースは元のデータベースの情報無損失分解であり，情報は失われていない。これは多次元空間$\boldsymbol{D}$の関数従属性による分解である。一般に関数従属性は複数存在するので，記憶する関係

も複数となる。数値データを配置する多次元空間を構成しない次元の空間での役割を関係集合で表した多次元データベースを $MRDB(N, \boldsymbol{D}, \boldsymbol{R})$（$\boldsymbol{R}$は関係集合）で表す。

次に，多次元データベースの設計について議論する。データベース化する数値データには数値データの性質を説明する値がある。性質を数値データの属性とする。数値データは属性集合$\boldsymbol{A} = \{A_1, A_2, \cdots, A_m\}$の値でその意味が示される。

属性集合が$\boldsymbol{A}$である数値データに対し，$\boldsymbol{A}$に関数従属性集合$\boldsymbol{F}$が存在するときに多次元データベースを設計する。設計は，多次元空間$\boldsymbol{D}$と関係集合$\boldsymbol{R}$を決定することである。

多次元空間は，$\boldsymbol{A}$の独立した属性で構成する。Kを$K \rightarrow \boldsymbol{A}$となる最小（極小）の属性集合とする。$K$の各属性を次元とすることで多次元空間を決定する。自明でない関数従属性$X \rightarrow Y$ $(X, Y \subseteq K)$が$\boldsymbol{F}$から導くことができるならば，$K - Y \rightarrow \boldsymbol{A}$が成り立つので，$K$は最小ではない。したがって多次元空間の次元間で成り立つ関数従属性は存在しない。多次元空間は$\boldsymbol{F}$から導かれる関数従属性によって正規化できないので，最小である。

Kに含まれない属性と多次元空間を構成する属性の関係が明らかになるように，関係集合$\boldsymbol{R}$を決定する。$\boldsymbol{A}$中の各属性と$\boldsymbol{D}$との関係が分かればよいのであれば，$\boldsymbol{R}$は次元とはならない属性と次元との対応を表す1つの関係でよいが，冗長性を削減し関数従属性による属性間の関係を陽に表すために関係を分解する。関数従属性による関係データベースの設計法には分解設計法や合成設計法があるが，それらの手法を用いて，無損失分解となるように属性集合$\boldsymbol{A}$を分解する。無損失分解では，結果の関係にKを含むものが存在する。そのような関係の属性集合がKに一致するときには，その関係を削除し，結果を$\boldsymbol{R}$とする。

多次元空間$\boldsymbol{D}$を構成しない属性に対応する次元Dを$\boldsymbol{D}$に加えても，次元間に関数従属性$\boldsymbol{D} \rightarrow D$が成り立つので，3節の議論から$D$を多次元空間から削除できる。すなわち，すべての数値データは，$\boldsymbol{D}$で構成される多次元空間に配置することができる。また，$\boldsymbol{R}$は次元とはならない属性と次元になる属性の対応を表す関係の無損失分解なので，任意の属性に対し，その属性の値と多次元空間に配置された数値データの対応の情報は失われていない。多次元空間の座標$(d_1, d_2, \cdots, d_n)$に対し，1つの組$(d_1, d_2, \cdots, d_n)$からなる関係と$\boldsymbol{R}$の関係集合を結合する。$K \rightarrow \boldsymbol{A}$より，結果の関係は1つの組のみからなる。座標$(d_1, d_2, \cdots, d_n)$にある数値データの$\boldsymbol{A}$の値は，この結果の組の値として求めることができる。MDBでは不明確だった次元間の関係は，$MRDB$では$\boldsymbol{R}$の構造として明確になる。属性集合$\boldsymbol{A}$で本質的に独立している属性のみが次元となり，利用者の理解が容易な多次元データモデルとなる。

5. 次元選択の処理

データを解析するために，多次元空間に対する次元選択と区間集約を行う。区間集約は，区間を表す属性をあらかじめ与えることによって次元選択として扱うことができる。本節では，4節で導入した多次元データモデルにおける次元選択について検討する。

属性集合$\boldsymbol{A}$を持つ数値データに対して多次元データベース$MRDB(N, \boldsymbol{D}, \boldsymbol{R})$を構築したとする。$MRDB$では多次元空間を構成しない属性が存在するので，次元選択を$\boldsymbol{A}$に対する属性選択として考える。選択する属性集合をW，$\boldsymbol{A}$中で多次元空間を構成する属性集合をK，選択される属性でKに含まれる属性の集合を$W_K = W \cap K$，含まれない属性の集合を$W_{\bar{K}} = W - K$とする。

属性集合Wの属性はすべてKに含まれるとき$(W = W_K)$，属性選択は多次元空間$\boldsymbol{D}$における次元選択で行うことができる。このとき関係集合$\boldsymbol{R}$は必要なく，$\boldsymbol{D}$のみが操作の対象となる。

属性集合WにKに含まれない属性が存在するとき$(W_{\bar{K}} \neq \emptyset)$，簡単には$\boldsymbol{R}$中の必要な関係を結合して多次元空間を拡張することで処理できる。しかしその方法では中間結果が大きくなるため，関係の結合時に段階的に集約を行うことで処理を効率化する。

［例2］ 属性集合$\boldsymbol{A}$を$\{A, B, C, D\}$，関数従属性集合$\boldsymbol{F}$を$\{A \rightarrow B, B \rightarrow C\}$とする。構築された多次元データベース$MRDB(N, \boldsymbol{D}, \boldsymbol{R})$は，$\boldsymbol{D} = \{A, D\}, \boldsymbol{R} = \{AB, BC\}$である。

1．属性集合$W_1 = \{A\}$の選択

AはKに含まれるので，多次元空間におけるAに対応する次元の選択となる。

2．属性集合$W_2 = \{C\}$の選択

CはKには含まれないので，$\boldsymbol{R}$との結合が必要になる。まず，多次元空間でAに対応する次元に対し次元選択を行う。結果を(N_0, A)とする。次に(N_0, A)を関係と見なし，ABと結合してBの値で集約する。結果を(N_1, B)とする。同様にBCと結合してCの値で集約した結果(N_2, C)のN_2が求める集約値である。

3．属性集合$W_2 = \{C, D\}$の選択

DはKに含まれるが，Cは含まれないので，2と同様に$\boldsymbol{R}$との結合が必要になる。まず，多次元空間でA, Dに対応する次元に対し次元選択を行うが，結果(N_0, A, D)は多次元空間と同じである。次に(N_0, A, D)を関係と見なし，ABと結合してB, Dの値で集約する。結果を(N_1, B, D)とする。同様にBCと結合してC, Dの値で集約した結果(N_2, C, D)のN_2が求める集約値である。 □

属性選択の処理のために，関数従属性による$\boldsymbol{R}$の順序を有向グラフで表す。関係Rの属性集合を$at(R)$で表し，関係の集合Pに対し$at(P) = \bigcup_{R \in P} at(R)$とする。また，多次元空間$\boldsymbol{D}$を属性集合が$K$である関係$R_0$と考える。

グラフ$G(V, E)$において，VはR_0と$\boldsymbol{R}$の各関係に対応する節点の集合である。節点を対応する関係で表す。有向枝の集合Eの枝(R_1, R_2)は，R_0からR_1への経路上の関係集合Pに対し，関数従属性$at(P) \rightarrow at(R_2)$が$\boldsymbol{F}$から導かれるときに存在する。

選択属性Wに対し，$at(M) \supseteq W_K$となる関係集合Mを$M = \{R_1, R_2, \cdots, R_u\}$とする。グラフ$G$での$R_0$から$R_i$までの経路を$P_i = \{R_{i0}(= R_0), R_{i1}, \cdots, R_{im_i}(= R_i)\}$，経路で$R_{ij+i}$に含まれる$R_{ij}$の属性集合を$J_{ij} = at(R_{ij}) \cap at(R_{ij+1})$ $(0 \leq j < m_i)$，$J_{im_i} = W_K \cap at(R_{im_i})$とする。

$\bigcup_{i=1}^{u} P_i$に対し，グラフGで定められた半順序に違反しないように全順序を定めたものを$Q = \{R'_0(= R_0), R'_1, \cdots, R'_t\}$とする。$Q$は集約値を求める順序である。

R_0では，W_Kに含まれる属性とJ_{i0} $(1 \leq i \leq u)$の属性で集約を行わなければならない。R'_kまで処理が終了しているときの集約は属性集合Z_kに対して行われているものとする。$Z_0 = at(R_0) \cap (W \cup \bigcup_{i=1}^{u} J_{i0})$である。

R'_kの処理では，結合後の属性集合$Z_{k-1} \cup at(R'_k)$に対して，Wに含まれる属性とR'_{k+1}以降の結合で使われる属性で集約する。R'_kがR_{ij}であるときJ'_kをJ_{ij-1}とすると，$Z_k = (Z_{k-1} \cup at(R'_k)) \cap (W \cup \bigcup_{i=k+1}^{t} J'_i)$である。

Qの順序でZ_kに従って集約を行うことで属性選択の処理ができる。

6. むすび

関数従属性を用いた多次元データベースの設計について検討した。独立した属性のみを次元とし，その他の属性を関係集合とすることによって次元間の関係を構造化できる。また，多次元データベースに対する基本的な操作である次元選択と区間集約の処理について検討した。

多次元データベースは大量のデータを扱うため，その処理効率が問題となる。そのために，ある程度集約したデータを蓄えておくなどの方法が考えられる。関係集合の属性は，次元の区間を定めるものであり，そのような集約値の記憶にどのように利用できるかを検討する予定である。

参考文献

[1] Chaudhuri, S. and Dayal, U., "An Overview of Data Warehousing and OLAP Technology," *ACM SIGMOD Records*, Vol. 26, No. 11, pp. 65-74, Nov. 1997.

[2] Harinarayan, V., Rajaraman, A., and Ullman, J. D., "Implementing Data Cubes Efficiently," *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, pp. 205-216, June 1996.

[3] Inmon, W. H., *Building the Data Warehouse*, 2^{nd} edition, John Wiley & Sons, 1996.

[4] Lehner, W., Ruf, T., and Teschke, M., "CROSS-DB: A Feature-Extended Multidimensional Data Model for Statistical and Scientific Databases," *Proc. Int'l Conf. on Information and Knowledge Management*, pp. 253-260, Nov. 1996.

[5] Sapia, C., Blaschka, M., Höfling, G., and Dinter, B., "Extending the E/R Model for the Multidimensional Paradigm," *Advances in Database Technologies, Lecture Notes in Computer Sci.*, Vol. 1552, pp. 105-116, Springer, March 1999.