

セルラーデータベースにおけるセル ID 設計

田代貢一

法政大学大学院工学研究科電気工学専攻

i00r3232@k.hosei.ac.jp

國井利泰

法政大学大学院工学研究科電気工学専攻

tosi@kunii.com : <http://www.kunii.com/>

要旨

現在サイバー世界では、様々な取引が行われていると同時に、時間とともに常に状況変化が起きている。セルモデルでは、サイバー世界で起きる急激な状況変化を柔軟にモデル化できる。これはその基本である不変量に基づく同値関係により、セルを動的に結合できるところにある。セル結合の際それぞれのセル空間において、セルの特定をする必要がある。そのため、セル ID を用いる。このセル ID の設計の際、重要となるのがスコープの概念である。セルのスコープ(有効範囲)を見だし、セル ID に付加することで、セルを特定することが可能になる。これによりセルをサイバー世界で一意に特定し、動的にセル結合を行うことが可能となる。

The Design of Cell Identifiers in the Cellular Databases

Koichi Tashiro

Electrical Engineering Course, Graduate School of Engineering, Hosei University

i00r3232@k.hosei.ac.jp

Tosiyasu L. Kunii

Electrical Engineering Course, Graduate School of Engineering, Hosei University

tosi@kunii.com; <http://www.kunii.com/>

Abstract

Today, in cyberworlds, various dealings are conducted on the web, while the dynamic changes of the situations are constantly taking place. The cellular model can flexibly model the rapid changes of the situations in cyberworlds. The foundation of the cellular model is in its capability to attach the cells dynamically via equivalence relations in quotient spaces (identification spaces) based on invariants. Since it is necessary to specify cells for cell attachment, Cell identifiers (Ids) are used. The scope is important for designing Cell IDs. It becomes possible to specify a cell by identifying the scope of the cell. This serves to specify the cell in cyberworlds and attach cells dynamically.

1. はじめに

データベースとはコンピュータ内に構築さ

れた実世界のモデルである。それは実世界の写像といってもよい。よって、データベース

を構築するためには、実世界をモデル化する必要がある。

現在、ほとんどの DBMS はリレーショナル DBMS により稼働している。リレーショナルデータモデルは、1970 年に E.F.Codd により提案[2]されたデータモデルである。数学の集合論の直積に基づいており、すべてのデータは数学的リレーションに対応して表で表現される。このリレーショナルデータモデルはデータベース管理者がデータの依存関係を認知していることを前提として、データをモデル化する。これにより、以下に示すような問題が発生してしまう。

- web のように急激に状況の変化が起こるような空間を表現することができない。
- 閉じた空間しか表現できないため、それぞれのデータベースでデータ管理の仕方に違いが出てきてしまい、異なるデータベース間でデータ管理、データ操作ができない。

以上のような問題があるため、リレーショナル DBMS では、状況変化が起こる度にデータベースの再正規化という再設計を行わなければならない。そのため、状況変化が起こるたびに、多大なコストがかかってしまう。

このような問題点を解決できる有効なデータモデルがセルラーデータモデルである。セルラーデータモデルは不変量に基づく様々な同値関係によりセル結合が動的に行える[1]。さらに、変化をホモトピー不変量[4]として保存することにより、web 上のような常に状況変化が起こるような空間をモデル化できる。

本論文では、セル結合を行う際にセルを特定するために用いるセル ID と、スコープについて論ずる。

2. セル空間理論

セル空間理論とは、抽象階層の概念を取り入れている事から、階層において継承される不変量を定義する事により、様々な状況をモデル化できる。このセル空間理論についてその概要を以下に示す。

2.1 セル空間構造

セル構造空間 cellular spaces についての概略を説明する。まず、セルは、トポロジー的に n 次元の開(OOPEN)ボール $IntB^n$ と同等なトポロジー空間にあり、 n -cell e^n と表記する。X から、セル接合により、有限あるいは、無限のセル列 X^p を inductive に構成することができる。 X^p は、X の、部分空間であるように構成し、整数 Z で索引付ける。この様にして得られる。 $\{X^p \mid p \in Z\}$ を、filtration と呼ぶ。

$X^p \subset C$ covers X (or X^p is a covering of X)

すなわち

$$X = \bigcup_{p \in Z} X^p$$

X^{p-1} は、 X^p の部分集合

すなわち、

$$X^0 \subseteq X^1 \subseteq X^2 \subseteq \dots \subseteq X^{p-1} \subseteq X^p \subseteq \dots \subseteq X$$

この様にして X から得られるセル構造空間 $\{X; X^p \mid p \in Z\}$ を filtration space と呼ぶ。

2.2 セル結合(cell attachment)によるセル空間構造

開 n -cell e^n を、既に構築されたトポロジー空間 X に、surjective かつ continuous な写像 f により結合することにより、セル構造空間 Y を構築できる。

写像 $f: X \rightarrow Y$ が surjective であるということは、

$$(\forall y \in Y) (\exists x \in X) [f(x) = y]$$

を意味する。

写像 f が continuous であるということは、
“a subset $A \subset Y$ is open in Y if and only if $\{f^{-1}(y) \mid y \in A\}$ is open in X ”

を意味する。

$$X \sqcup_f Y = X \sqcup Y / \sim$$

は、attaching space と呼ばれる transitivity から、同値関係により、空間を equivalence classes の排他的和に分割できる。一つの equivalence class を x/\sim と表記する。すると

$$x/\sim = \{y \in X \mid x \sim y\}$$

である。

すべての equivalence class の集合を X/\sim と表記するとそれは X の quotient space あるいは、identification space と呼ばれ

$$X/\sim = \{x/\sim \in 2^X \mid x \in X\} \subset 2^X$$

である。

3. セルラーデータモデル

セルラーデータモデルとは、セル空間理論を適用したデータモデルである。セルラーモデルは、抽象概念の階層に基づいているため、既存の様々なデータモデルの性質を包含している。分かりやすくするため、リレーショナルデータモデルと比較しながら説明する。リレーショナルデータモデルにおける relation は集合であるので、抽象階層的には下位のセル理論に継承される。従って、セルは少なくともテーブルとして扱える。N 個の属性をもつテーブルは、n 個の自由度をもつことから、n-cell として表わす。リレーショナルデータモデルではリレーションスキーマの度数に応じて、n 項リレーション(n-ary relation)と表わしている。

セル空間理論には境界という概念がある。n 次元の close-cell とは境界を持つセルであ

り B^n と表わし、n 次元の open-cell とは境界を持たないセルであり e^n と表わす。境界とは 1 次元低いセルであるので、 $n \geq 1$ の場合 B^n の境界とは

$$\partial B^n = B^n - \overset{\circ}{B^n} = S^{n-1}$$

で表わされ、n-1 次元の空間である。

セルをオープンにすると、 $n \geq 2$ の場合には属性が 1 つのセル(テーブル)になる。($n = 1$ の場合には境界は B^0 となり、属性を持たない。)セルラーデータベースではオープンセルの属性をリレーショナルデータモデルにおける主キーの役割になるように設計する。さらに、セルを特定するために、後述するセル ID を付けてセルを管理する。

4. スコープの定義

人は知らず知らずに、物事を一意に特定するために、スコープという概念を取り入れている。一般に、スコープは、XML[5]に見られるように、ツリー構造あるいはグラフ構造をとる。例えば、「山田太郎」という人物を特定するために、「法政大学工学部電気電子工学科の山田太郎」というように、「法政大学工学部電気電子工学科」というスコープ(有効範囲)をつける。スコープとは、オブジェクトを必ず一意に決めるため範囲であるので、もし法政大学工学部電気電子工学科に 2 人「山田太郎」という人物がいたのなら、このスコープのとり方は不適切になり、「山田太郎」を特定することができなくなる。このような場合、「法政大学工学部電気電子工学科 3 年の山田太郎」というように、スコープをより狭くとり直さなければならない。この場合のスコープは図 1 のようなツリー構造をとっている。そして、このツリーの親からのスコープをす

べて付加していき「山田太郎」を特定できる。
また、これはこのツリーの親からのパスという見方もできる。

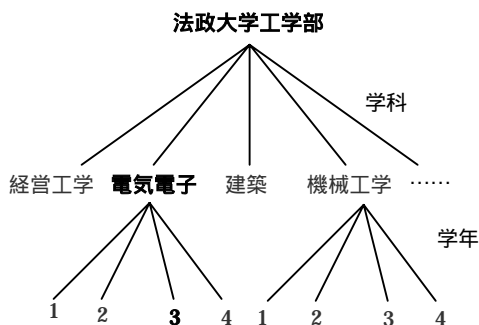


図 1. スコープのツリー構造

4.1 プログラミング言語におけるスコープ

このスコープという概念はコンピュータでは広く取り入れられている。プログラム言語におけるスコープの例を示す。

プログラミング言語には変数を特定するために、変数のスコープルールがある。変数を定義するときには、その変数がどの範囲まで効果が及ぶのかを知る必要があり、変数を参照するときには、どこで定義されている変数なのかを知る必要がある。これを解決するのが、変数のスコープルールである。

JAVA 言語を例にとると、JAVA 言語では変数を参照するときのスコープルールを以下のように定めている[3]。

1. 参照している場所と同じレベルのブロックを探す。
2. メソッド内でその外側のブロックを探す。
3. メソッドの属するクラス内の変数を探す。
4. スーパークラスの変数を探す。
5. ここまで見つからなければ、コンパ

イル時にエラーになる。

プログラム中の変数はこの順番で探され、最初に見つかった変数が参照されます。以降の条件に見合う変数が合っても参照されない。スコープルールを図 2 に示す。

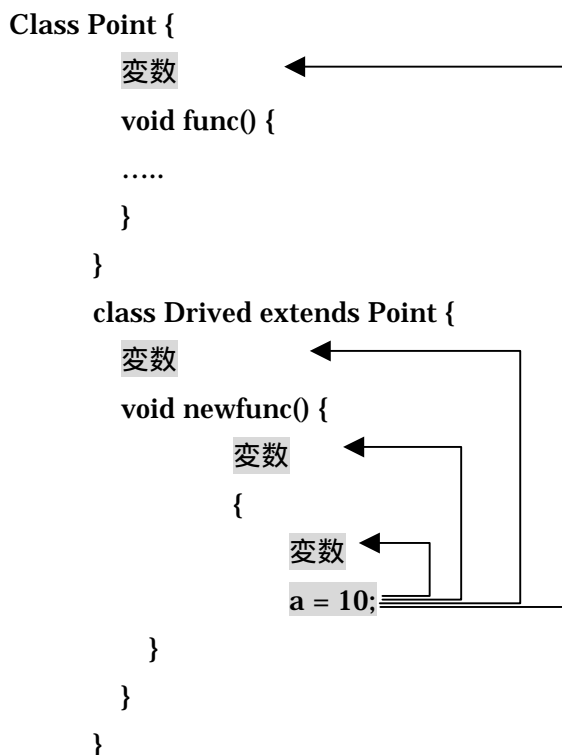


図 2. JAVA 言語におけるスコープルール

4.2 e-mail アドレスにおけるスコープ

次に、web 上でのスコープについて考察する。以下の e-mail アドレスを例にとる。

tashiro@k.hosei.ac.jp

e-mail アドレスは web 上で、一意に特定できるものである。その構成を見てみると、次のようになる。@前の名前(tashiro)は@後のドメイン名(k.hosei.ac.jp)の中でユニークでなければならない。よって、@前の名前のスコープは@後のドメイン名となっている。また、ドメイン名は web 上でユニークでなければならないので、@後のドメイン名のスコープは、

web 上である。図 3 に email アドレスのスキームのツリー構造を示す。

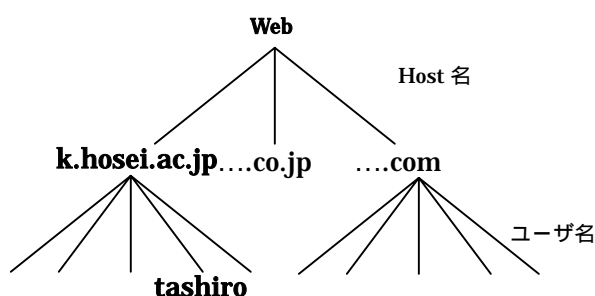


図 3. e-mail アドレスのスキーム

e-mail アドレスは以下のような構成になっており、@前の名前にスキームであるドメイン名を付加することにより、web 上でユニークなものとなっていることが分かる。

名前@スキーム(ドメイン名)

このように、スキームはオブジェクトを一意に特定するとき、重要な役割を果たしている。そして、オブジェクトにスキームを付加することによって、オブジェクトを一意に特定できるようになる。

5. セル ID

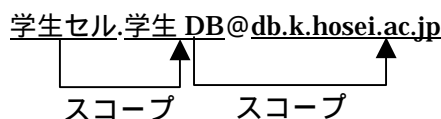
セルラーデータベースでは不変量に基づく同値関係によって、web 上でセル(テーブル)の結合が可能である。このとき、どのセルと結合するか特定する必要がある。そこで、セルラーデータベースではセルを特定するためにセル ID を用いる。リレーショナルデータベースにおけるテーブル ID の同等の役割である。

セルラーデータベースではセル ID をセルの 0-cell に書き込むように設計する。0-cell とは、独立変数一つも持たないセルである。すなわち、属性ではない。

$$\partial B^1 = B^0 = k = \text{cellID} \quad (k \text{ は定数})$$

0-cell はどのセルも必ず含んでいるので、ここにセル ID を書き込むことができる。さらに、セル ID をセル ID テーブルに書き込むことで、セルを管理する。

セル ID をつける上で、重要な役割を果たすのが前述したスキームである。Web 上のデータベースにおけるスキームの定義は次のようになる。データベース名はそのコンピュータ内でユニークであるため、データベースのスキームはホスト(ドメイン)である。セル名はデータベース内でユニークであるため、セルのスキームはデータベースである。属性名はセルの中でユニークであるため、属性のスキームはテーブルである。これより、データベース、セル、属性はスキームを付加することにより、それぞれ web 上で一意に決めることができる。例えば、コンピュータ「db.k.hosei.ac.jp」にある「学生 DB」というデータベースの「学生セル」というセルを特定する場合、それぞれのスキームを学生セルに付加して以下のように表現する。



このように表現することにより、web 上でこの「学生セル」を一意に特定することができる。これをセル ID とし、セル ID テーブルに書き込み管理することで、web 上でセルを特定することが可能になる。

6. 事例研究

web 上で商品販売を行っている会社が、取引先を追加する場合をリレーショナルデータモデルと比較する。Web 上での商品販売とは顧客データと商品データのマッチングである

とする。

6.1 リレーショナルデータモデルの場合

例として以下のようなデータベースで商品販売を行っていたとする。

自社データベース

Goods(GoodsID,CategoryID,
GoodsName,Price,MakerID)

GoodsCategory(CategoryID,
CategoryName)

Maker(MakerID,MakerName)

Customer(CustomerID,Name,
Address,Tel,Email>Password)

GoodsOrder(OrderID,CustomerID,Order
Date,Pay,MailFlag)

GoodsOrderDetail(OrderID,GoodsID,
Amount)

リレーショナルデータベースでは、異なるデータベース間でテーブルの結合等の操作はできない。そのため、この場合新しい取引先のデータベースの商品リストテーブルから、自社のデータベースの商品リストテーブルに商品データを追加する必要がある。

新規取引先のデータベースでは商品データを以下のように管理していたとする。

新規取引先商品テーブル

Goods(GoodsID,GoodsName,
Price,MakerName)

このとき、自社のデータベースでは商品マスタを3つのテーブルに分けて管理しているのに対し、新規取引先のデータベースでは商品マスタを1つのテーブルで管理しているので、自社のデータベースの商品マスタに新規取引先のデータベースの商品データをそのまま挿入することができない。これは、リレーショナルデータベースは1つの組織ごとにデ

ータベース管理者が、データの依存関係を認知した上でデータベースの設計を行っているためである。言い換えると、リレーショナルデータベースはその組織だけの閉じた空間を射影しており、その閉じた空間内ではデータ管理が行えないためである。またこのことは、リレーショナルデータベースの最大のスコープはデータベースであるため、データベースより大きくスコープをとることができず、異なるデータベース間でテーブル、属性の特定はできず、データ操作ができないという点からもいえる。

この例では、新規取引先の商品データをエクスポートし、自社のデータベースに合わせて商品データを変換し自社のデータベースにインポートすることにより解決はできるが、会社の合併、提携では、データベースを再設計する必要があり、多大な費用、開発時間がかかることは避けられない。

6.2 セルラーデータベースの場合

セルラーデータベースでは抽象階層を取り入れることにより、不変量に基づく同値関係により動的にセル結合を行うことができる。したがって、この場合新規取引先の商品マスタをそのまま使用し、顧客セルと結合することができる。

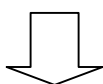
具体的には、セルを図4(a)のような、セルIDテーブルで管理していたとする。自社データベースにあるセルは、自社データベース内でユニークであるので、セルIDはスコープを省略して記述し管理しているものとする。また、実際にはセルIDテーブルにはパーミッション等の情報も記述するが、ここでは簡単のためそこまで議論しないこととする。

ここに、新規取引先の商品セルを追加して、

管理できるようにする。前述したように、新規取引先の商品セル ID にスコープを付加して、自社データベースからも新規取引先の商品セルを特定できるようにする。ここでは、web のルールである@を使用し、セル ID を記述することにする。

セル ID	セル
商品セル	B_m^q
顧客セル	B_c^p

(a)新規取引先追加前のセル ID テーブル



セル ID	セル
商品セル	B_{m1}^q
顧客セル	B_c^p
商品セル@新規取引先 DB	B_{m2}^t

(b)新規取引先追加後のセル ID テーブル

図 4 . セル ID テーブル

既存の商品を販売する場合は、既存の商品セルと顧客セルを結合する。

$$B_c^p \sqcup_f B_m^q$$

$$f : B_c^p \rightarrow B_{m2}^q / \sim$$

新規取引先の商品を販売する場合は、新規取引先の商品セルと顧客セルの結合する。

$$B_c^p \sqcup_f B_{m2}^t$$

$$f : B_c^p \rightarrow B_{m2}^t / \sim$$

このように、セルラーデータモデルではセルを動的に結合できるため、このような状況変化に対し柔軟に対応でき、現在使われているリレーショナルデータベースに比べて多大なコストを削減できることが分かる。

7. 結論

本論文では、セルラーデータモデルにおいて、セル結合を行う際セルを特定するためのセル ID とスコープについて議論した。セル ID はスコープを付けることにより、web 上でセルを特定することができるため、異なるデータベース間でもセル結合が可能となることを示した。

結論として、閉じた空間をデータベース管理者がデータの依存関係を認知していることを前提としているリレーショナルデータモデルでは、状況変化が起きる度にデータベースの再設計等をしなければならないのに対し、セルラーデータモデルはセル理論上に構築されているため、動的にセル空間を結合することができ、様々な状況の変化に柔軟に対応できる。

最後に、今回は主にリレーショナルデータモデルとの比較を行ったが、その他のデータモデルとの比較は今後の課題とする。

8. 引用文献

- [1] T.L.Kunii and H. S. Kunii, "A Cellular Model for Information Systems on the Web - Integrating Local and Global Information", Proceedings of 1999, International Symposium on Database Applications in Non-Traditional Environments (DANTE'99), November 28-30, 1999, Heian Shrine, Kyoto,

Japan, Organized by Research Project on Advanced Databases, in cooperation with Information Processing Society of Japan, ACM Japan, ACM SIGMOD Japan, pp. 19-24, IEEE Computer Society Press, Los Alamitos, California, U.S.A.

[2] E.F.Codd A Relational Model of Data for Large Shared Data Banks CACM , (1970)

[3]JAVA プログラミング講座 日本サン・マイクロシステムズ株式会社 サン・サービスエデュケーションサービス部 アスキー出版 (1996)

[4] T. L. Kunii, "Homotopy Modeling as World Modeling", *Proceedings of Computer Graphics International '99 (CGI99)*, (June 7-11, 1999, Canmore, Alberta, Canada) pp. 130-141, IEEE Computer Society Press, Los Alamitos, California, U. S. A.

[5]S.Abiteboul ,P.Buneman D.Suciu ,Data on the Web -From Relations to Semistructured Data XML- ,Morgan Kaufmann,2000.