

マルチメディアデータベースのための データ定義ダイアグラム

森寄 慎之 家富 誠敏 富井 尚志 有澤 博

横浜国立大学 工学部 電子情報工学科
〒240-8501 横浜市保土ヶ谷区常盤台 79-5

E-mail: {nor,eto,tommy,arisawa}@arislab.dnj.ynu.ac.jp

本稿では、マルチメディアデータベースのデータ定義(スキーマおよびインスタンス定義)を、視覚的なユーザインタフェースを用いて直観的な直接操作によって行なうことができるようなシステムについて述べる。本システムでは、スキーマ定義をスキーマダイアグラムを描くことで行ない、さらにスキーマダイアグラム上にインスタンス図を描くことによって意味制約に従ってインスタンス入力を行なう。

これによって利用者はスキーマおよびインスタンス情報を把握しながら、常に正しいデータ定義を行なうことができるようになる。

A Data Definition Diagram for Multimedia Database

Noriyuki MORISAKI, Masatoshi IETOMI, Takashi TOMII, Hiroshi ARISAWA

Division of Electrical and Computer Engineering
Faculty of Engineering
Yokohama National University
79-5, Tokiwadai, Hodogaya-ku, Yokohama-shi, 240-8501 Japan

E-mail: {nor,eto,tommy,arisawa}@arislab.dnj.ynu.ac.jp

This paper describes a visual interface of multimedia database to define a schema and instance data. In this interface, a schema data can be defined by drawing a schema diagram, and instance data can be defined by drawing an instance on the schema diagram. A user can define schema and instance data intuitively and flexibly.

1 はじめに

近年の計算機技術の発達により、画像、映像、3次元情報、音声などからなるマルチメディアデータを計算機上で取り扱うことが出来るようになってきた。そのため、様々なメディアを統括的に蓄積、管理、運用するためのマルチメディアデータベースの実現への要求が高まってきている。マルチメディアデータは文字、数値と比較すると大容量で、それぞれのメディアやデータ間の相互参照が入り組んでい

るなどして、表現するために非常に複雑な構造が必要となることが多い。例えば画像の場合、高度なマルチメディアデータベースにおいては、画像データそのもの(1次データ)だけでなく、インデックスを用いて、そこに何が写っているかといった意味情報(2次データ)も、それぞれ相互参照されている。

マルチメディアデータは上記のような特徴を持つため、全体の構造を把握することが非常に困難になっている。このような問題に対する解決法として、データベースを視覚的、直観的に捉えやすい形

で表現することが考えられる。

データベースの構造を視覚化するという試みは、バックマンダイアグラムやERダイアグラム[1]などに代表されるダイアグラム表記法があり、スキーマ設計などに古くから用いられてきた。これらを含め、視覚化方法は大きく分けて、

- タイプレベルでの関係や制約をダイアグラムによって記述する
- 大量のインスタンスの一部を、範囲を限って閲覧する。
- スキーマと、そのスキーマに対応するインスタンスを視覚化する

に分類される。

例えば、オブジェクト指向方法論 (OMT)[2] では、オブジェクトの設計をダイアグラムによって記述することで、オブジェクトの内部構造やオブジェクト間の対応関係の制約などを視覚的に表すことができる。そのためのツールとしてはOMTROLL[3]が研究されている。また、ERダイアグラムでは、定義されるエンティティとリレーションシップの対応を図示することで、データベーススキーマを視覚的に設計することが可能となっている。

ところで、これらはスキーマによって表現されたタイプ同士の制約関係を視覚化しているに過ぎず、そこに属するインスタンスの個別の対応までは視覚化されるにはいたっていない。この原因には、データベースで取り扱われるデータは大量かつ長大になり得るため、インスタンスをいちいち視覚化するよりも、タイプによってインスタンスの一般記述を行なうことがデータベースの役割の一つでもあったからである。さらに、複数の従属性が存在するインスタンスを表示した場合には、それらのつながり方が複雑になり過ぎて有効に視覚化することは困難である。また、マルチメディアデータベースにおいては、ビデオ映像や音声などのメディアデータは大容量であることや時間依存性があるだけでなく、メディアごとにその表示方式がことなる。

しかし、コンテンツやその意味に基づいてデータベースを構築したり更新したいという場合には、インスタンスとインスタンス同士のつながりを、値まで含めて視覚化できることは極めて有効である。すなわち、メディアデータだけでなく、様々なメタ情報を同時に蓄積するマルチメディアデータベースを、インスタンスまで含めて視覚化するためには、

- タイプの定義 (スキーマ設計) ができること
- タイプ間のつながりを表示できること
- タイプに属するインスタンスを、タイプ間の制約を満たして表示できること
- 値データを表示できること
- マルチメディアデータを表示できること
- インスタンスの「簡単な」巡回や検索、更新操作を行なえること

の条件を満たしている視覚化インタフェースを構築し、データを直観的に視覚化できれば、有効に利用することができる。

そこで本研究ではこれらの条件のうち、スキーマ設計に重点を置き、さらにインスタンスの入力も可能であるようなシステム、AVIS (Advanced Visualizer of Instances and Schema) を設計し、その実装を行なった。AVISでは、マルチメディアデータベースのデータ定義を、視覚的なユーザインタフェースを用いてスキーマダイアグラムを描くことを行ない、さらに意味制約に従ってインスタンス入力を行なうことができる。

2 視覚的データ定義システム

AVISとはスキーマ、およびインスタンスを利用者にとって把握し易いように直観的に視覚化し、非常に簡単な操作でデータ定義が行なえることを目指したシステムである。このシステムは次のような機能を備える。

- 視覚的、直観的な操作を用いてダイアグラム (スキーマ図) を描くことによって、データベースのスキーマを定義する。
- スキーマ図上に、インスタンスを表現したインスタンス図を描き加えていくことによって、インスタンスを入力できる。
- 常に、制約に従った正しいデータ定義が行なえる。
- 様々なメディアデータを表示する。
- 実際のデータベースに反映させるためのインタフェースを有する。

このようなことが、このシステムの要件だと考えられる。以下にそれぞれについて詳細を述べる。

2.1 スキーマ定義

AVIS では、スキーマの定義はダイアグラムを描くことによって行なう。これにより利用者はデータベースのスキーマを、直観的な操作で定義することができる。

2.1.1 AIS ダイアグラム

データベース設計のうち、データベース化の対象となる世界の情報の構造などを細かく分析し、明確にする、概念設計を行なうための代表的な方法として、ER モデルに基づく ER ダイアグラムを用いる方法がある。

関係データベースのデータベース設計を行なう場合の多くは、ER モデルのスキーマを書くことで概念設計を行ない、それを関係データモデルのスキーマに変換することによって論理設計を行なっている。しかし、その変換を行なうことによって、概念設計の段階ではあった情報が、最終的なデータベースでは完全に反映されない可能性があるなど、不都合があることが知られている。

マルチメディアデータベースの設計においては、蓄積されるマルチメディアデータが、共通の性質を持つものの集合同士の間での相互参照が入り組んでいて、非常に複雑な情報構造を持つために、その概念設計には従来にない高い情報表現能力を持つデータモデルが必要となる。さらに、概念設計から論理設計への変換がなく、概念設計の段階で書かれたスキーマが、そのままデータベースのスキーマとなることが要求される。

本稿では、このようなことを満たすデータモデルとして、様々な制約をダイアグラムによって表現できるという特徴を持つことから、関数型データモデル AIS[4][5][6] を採用し、そのスキーマ表現法である AIS ダイアグラムを利用する。関数型データモデルは ER モデルなどに代表される意味データモデルをさらに発展させたものである。

AIS ダイアグラムによって表現される制約は図1のようになっている。

2.1.2 ダイアグラムを用いたスキーマ定義

このような AIS ダイアグラムを、データ定義を行なうための視覚的なユーザインタフェースに利用するのは有効な手段である。そこで、AIS ダイアグラムに基づくスキーマ図を非常に簡単なマウス操作を用いて描く、という行為を、AIS モデルのスキーマ

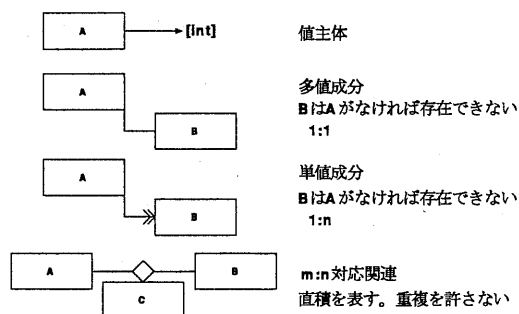


図1: AIS ダイアグラム

定義をする、ということに対応させるインタフェースが考えられる。例えば、画面上にアイコン化された主体型 (entity type) を描くと、その主体型は定義され、主体型と主体型を結ぶ線を描くと、その主体型間に関連が定義される、といったものである。

このような操作系であれば、画面上には現在定義されているスキーマが描かれているので、設計中のデータベースのスキーマ図を常に確認しながらスキーマ定義を行なうことができる。AVIS で実現した第一の要素として、このようなスキーマ定義があげられる。

2.2 インスタンス定義

AVIS では、スキーマ図上に重ねてインスタンス図を描くことによってインスタンスの入力を行なう。

2.2.1 インスタンスの視覚化と入力

インスタンスとは、情報の枠組であるスキーマに基づいてその枠内に蓄積された現実の情報のことである。スキーマを表現したダイアグラムは、データベースに与える制約を図示しているため、これによりどのようなタイプがあるのか、またどのタイプの間に個々のインスタンス同士のつながりを定義すべきであるかが示されている。つまり、スキーマダイアグラムが示す制約をインスタンスは必ず満たしていなければならない。

上述の AIS ダイアグラムはこのような特徴を持つことより、それを視覚的、直観的にインスタンス定義を行なうためのユーザインタフェースに利用するのは、有効な手段である。そこで、スキーマ定義の際に描かれた AIS ダイアグラムに基づくスキーマ図にさらに、簡単なマウス操作を加えて、インス

タンスを表現したインスタンス図をスキーマ図に重ねて描く、という行為を、AIS モデルに基づくインスタンスを定義する、ということに対応させるインタフェースが考えられる。例えば、すでに描かれたアイコン化された主体型の中に主体 (entity) を表すアイコンを描くと、その主体は定義され、主体と主体を結ぶ線を描くと、その主体間に関連が定義される、といったものである。この機能を AVIS で実装した際の、インスタンスの表示例を図 2 に示す。

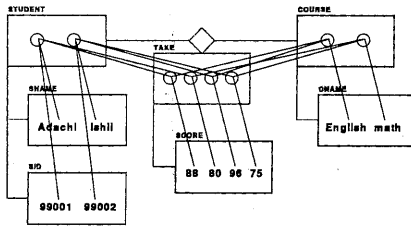


図 2: インスタンス図の例

このように AVIS では、すでに描かれたスキーマ図上にインスタンス図を描くことになるので、スキーマ、インスタンスの両方の情報構造を一覧することができ、常にスキーマを意識しながらインスタンス入力を行なうことができる。

2.2.2 大量データの表示

大量のデータがあるとき、画面上に表示できる主体の数には限りがあるため、それをどのようにして表示するかという問題がある。

これに対して AVIS では、タイプ表示にスクロール機能を付加することで実現することにした。

2.3 制約に従ったデータ定義

スキーマ定義については、AIS ダイアグラムの表記に関する制限、つまりどのようなスキーマ図を描いて良いか、ということに対しての制限がある。

また、インスタンス定義については、AIS ダイアグラムに基づくスキーマ図自体が個々のインスタンスが満たすべき制約を表現している。

このような制限や制約に反したデータ定義が、確認もされずに行なうことができてしまえば、非常に問題である。そのため AVIS では、制限、制約に反するデータ定義を行なおうとする操作は排除され、常に正しいデータ定義だけが行なわれるように

した。それによって、画面上に表示されたスキーマ図、インスタンス図は常に制限、制約に従った状態であり、利用者が特に意識しなくても正しいデータ定義が行なわれる。

2.4 メディアデータの表示

マルチメディアデータベースにおいては、画像、映像、音声、描画など、様々なメディアデータを扱うが、それらメディアデータを視覚的、直観的に把握できることが必要とされる。

インスタンスとして値が入っているとき、そのようなメディアデータを表示、再生するために、例えば映像や音声のように時系列に沿った、値の列を一つの単位としなければならないことがある。

そのようなときは、メディアデータから表示、再生が可能な値の列 (値列) を生成する必要がある。また、場合によってはファイル形式にすることも考えられる。さらに、メディアごとにその値列を再生するためのプレーヤーを埋め込んで、メディアデータの表示を可能にしなければならない。

AVIS では、値列 (vauseq) の概念を導入することで、メディアデータの表示を可能にした。

2.5 データベースとのインタフェース

AVIS 上で行なったデータ定義、つまり、作成されたスキーマ図、インスタンス図で示されるスキーマおよびインスタンスを実際のデータベースに反映させるためには、データベースとの間に何らかのインタフェースを設ける必要がある。これは、AVIS の状態をデータベースに反映させるためのものと、データベースの状態を AVIS で表すためのものの、2つが考えられる。これらによって、AVIS を用いたデータベースの構築、更新などが可能となる。

3 AVIS の実装と機能

ここでは、前章に述べたような考えに基づいて設計した AVIS の実装とその機能について述べる。

3.1 実装手法

本稿で実装を行なった AVIS の構成は図 3 のようになっている。

このシステムは、Java 言語 [7][8] を用いて開発したが、このとき Java アプレットという形態をとった。そのためユーザが実際に AVIS を利用する際に

は、Web ブラウザ上に Java アプレットを表示して、操作を行なう。

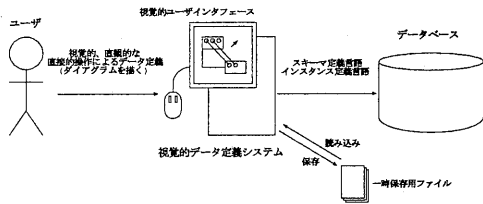


図 3: システム構成図

3.2 機能

画面上にスキーマ図のみ表示されるときはスキーマモードといい、主体型および関連の定義などを行ないスキーマ図を描くことでスキーマ定義をすることができる。一方、スキーマ図と同時に重ねてインスタンス図が表示されるときはインスタンスモードといい、主体を定義しインスタンスを入力し、インスタンス図を描くことでインスタンス定義をすることができる。

図 4 に実装した AVIS の実行画面を示す。

実装された AVIS は、それぞれのモードによって異なる次に挙げるような機能を有する。

3.2.1 スキーマモード

スキーマ定義を行なう際には、スキーマ図を描くために次のような機能を実装した。

Make	主体型を定義する
Delete	定義した主体型および関連を削除する
Move	定義した主体型および関連を移動する
S-compo	単値成分を定義する
M-compo	多値成分を定義する
Relate	関連を定義する

図 5: スキーマ定義のための機能

3.2.2 インスタンスモード

インスタンス定義を行なう際には、スキーマ図上に重ねてインスタンス図を描くために次のような機能を実装した。

Make	主体および成分主体を定義する
Delete	定義した主体を削除する
Move	定義した主体型および関連を移動する
Relate	関連を定義する
Cut	関連を削除する
Value	値を参照する (マルチメディアデータの場合は viewer により値を表示)
Valseq	値列を生成する
Restrict	表示させる主体を制限する

図 6: インスタンス定義のための機能

このような機能により、メディアデータの表示、大量のデータの表示が行なえるようになっている。

3.2.3 共通の機能

どちらのモードでも使用できる、共通の機能として次のようなものを実装した。

Sch	スキーマ定義テキストを生成する
Ins	インスタンス定義テキストを生成する
Save	現在の状態を一時的に保存する
Load	一時的に保存された状態を読み込む
Create Ins	スキーマを構築する
Create Sch	スキーマを構築する
From DB	データベースの状態を再現する
Schema	モードを変える
Instance	モードを変える

図 7: 共通の機能

このような機能により、このシステム上でなされたデータ定義を実際のデータベースに反映させることなどができる。

4 まとめ

本研究では、マルチメディアデータベースにおけるデータ定義を、非常に簡単なマウス操作を用いてダイアグラムを描くという、視覚的、直観的な直接操作によって行なうことができるようなシステム、AVIS の設計および実装を行なった。

AVIS ではデータベースのスキーマ定義をダイアグラム表記によって視覚的に行なうことができるう

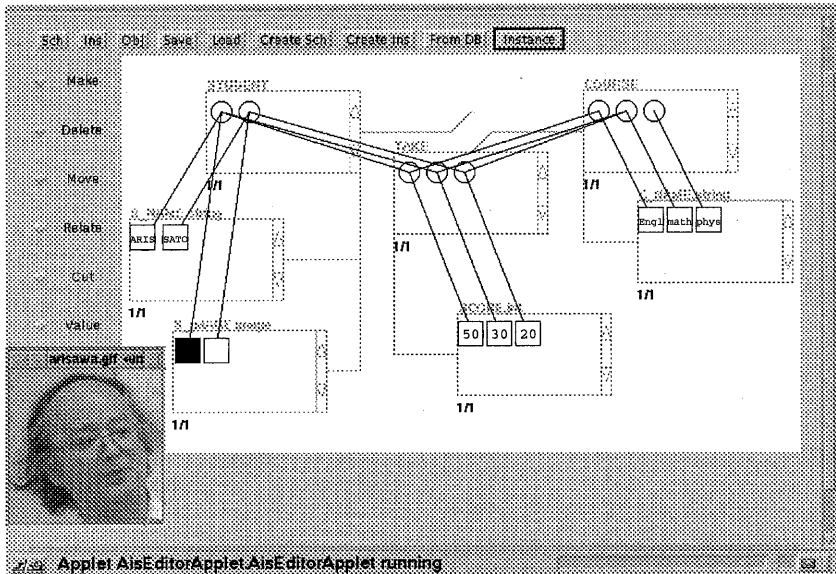


図 4: AVIS の実行画面

え、さらにそこに示される、インスタンスが満たすべき制約に従って、スキーマ図上にインスタンス図を描くことによってインスタンス定義を行なうことができるため、常に正しいデータ定義が行なわれるようにしている。

これにより、複雑な構造を持つ情報をモデル化し、データ定義を行なう際、データベースの内部の構造をよく理解しながら、制約に従った正しいデータ定義が行なえるので、データベース設計者の負担を小さくすることができる。

参考文献

- [1] P.P.Chen "The Entity-Relationship Model : Toward a Unified View of Data", ACM Transactions on Database Systems, Vol.1, No.1, 1976.
- [2] J.Rumbargh, et al.: "Object-Oriented Modeling and Design", Prentice Hall, New York, 1991.
- [3] A.Grau, et al.: "The TROLL Approach to Conceptual Modelling: Syntax, Semantics and Tools", Proceedings of the International Conference on Conceptual Modelling, pp.277-290, 1998.
- [4] H. Arisawa, H. Nagae, Y. Mochizuki, "Representation of Complex Objects in Semantic Data Model "AIS" and Implementation of Set Operators", IEICE TRANSACTIONS, vol.E74, No.1, 1991.
- [5] H. Arisawa, T. Tomii, H. Yui, and H. Ishikawa, "Data Model and Architecture of Multimedia Database for Engineering Applications", IEICE TRANS. INF. & SYST. , vol E78-D, No.11, 1995.
- [6] 石川 英彦, 有澤 博, "データベーススキーマの効果的な図示手法", 情報処理学会データベースシステム研究会, 1996.
- [7] "JAVA プログラミング講座", 日本サンマイクロシステムズ, 1996.
- [8] Patrick Naughton 著, 豊田 光智衣, 細井 一雄 訳, "Java プログラミング徹底解説", 日経 BP 社, 1991.