

JDMFにおけるマルチメディア機能拡張の考察

2E-9

田中 聡

穂鷹 良介

筑波大学 経営・政策科学研究科

筑波大学 社会工学研究科

1. はじめに

データベースはビジネス分野で業務目的別につくられていたデータを、共有するために必要となり生まれた。将来、マルチメディアデータでも同様に共有化の必要性が生じてくると考えられる。

本研究では、ビジネスデータ（既存の数値や文字で表されるもの）用の関係モデルの発展型であるJDMF/MODEL-1992[1]（以下JDMF/M-92とする）に基づく抽象モデルによって、QuickTimeなどの既存のマルチメディアデータを記述し、マルチメディアデータの本質を明らかにするとともに、マルチメディアデータを表現する上でJDMF/M-92及びそれに付随する設計ガイドラインにおいて、問題が生じるかを考察し、改良すべき点を検討する。

2. JDMFとセマンティックダイアグラム

一般的に用いられているバックマンダイアグラム[2]ではJDMF/M-92の属性と定義域の明確な関係が明記できない。一方、セマンティックダイアグラム[3][4]では、必要に応じた詳しさを、データモデルの設計結果を見せることが可能である。

本稿では、データ構造をセマンティックダイアグラムとバックマンダイアグラムを用いて記述する。セマンティックダイアグラムの記述法は図1に示す。

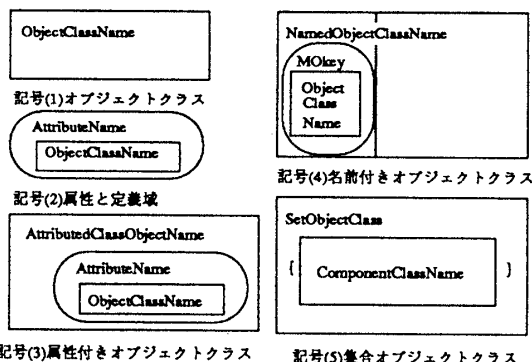


図1 セマンティックダイアグラムの書き方

3. マルチメディアデータの性質

検索対象となるデータを、データ自身の性質によってレベル分けをすると、「マルチメディア・作品」と「マルチメディア・リソース」の2つに分類される。

(イ) マルチメディア・作品（ムービー）

これは、いくつかの映像と音（リソース）によって構成された時間的な変化を持つデータストリームである。構成されたもの全体で一つの意味を持つ。たとえば、映画作品、オーケストラによるコンサートなどがこれにあたる。

(ロ) マルチメディア・リソース

これは時間的な変化を持つ集合として管理される最小単位のデータストリームである。たとえば、オーケストラの演奏の1楽章、映画の1カットなどがこれにあたる。このレベル内で時間的指定をすれば、サンプリングされた映像や音を操作することも可能であると考えられる。

4. JDMFマルチメディアデータモデルの設計

既存のモデルを参考に以下のようなモデルを考えた。なお、ムービーはいくつかのリソースから構成される。

(1) リソース非共有型モデル

リソースをそれぞれのムービーに取り込んだ型。

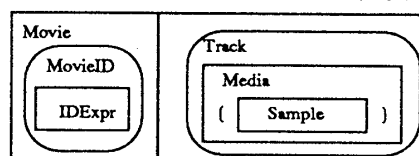


図2 リソース非共有型モデル(Semantic Diagram)

(2) 一方向リソース共有型モデル

リソースを分離して、他のムービーからの共有を可能にする型（図3 a,b）。各オブジェクト間には参照関係があるが、バックマンダイアグラムでは記述できない。また、このデータ構造は自然であると思われるが、JDMF/M-92および、その設計ガイドラインでは禁止されているデータ構造である。

A Proposal for a Multimedia Extension to JDMF

Satoshi Tanaka, Ryosuke Hotaka

University of Tsukuba, Tsukuba, Ibaraki, Japan

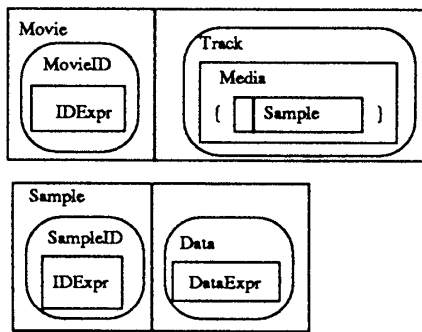


図 3 a 一方向リソース共有型モデル(SD)



図 3 b 一方向リソース共有型モデル (Bachman Diagram)

(3) 双方向リソース共有型モデル

共有されたリソースがどのムービーに属しているかまで記述された型 (図 4 a,b)。

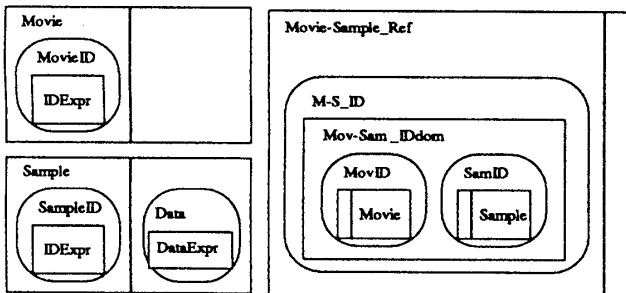


図 4 a 双方向リソース共有型モデル(SD)

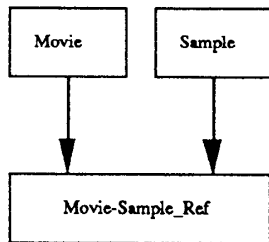


図 4 b 双方向リソース共有型モデル(BD)

5. 問題点

JDMF/M-92でマルチメディアデータを記述するにあたって、図 3 aのMovieのように、集合オブジェクトの要素クラスとして名前付きオブジェクトが現れる、JDMF/M-92の設計ガイドラインに違反する記述が現れた。この記述は設計ガイドラインに、敢えて従って記述することも可能であるが、図 3 aの記述はモデル的には、いかにも自然であったため、なぜこのような問題が起きたかを考察してみた。

6. 考察および結論

まず、共有について考える。共有は何のために行なわれるのであろうか。我々は以下の理由からだと考える。

(a) 更新のため

(b) 大きすぎる情報を扱うため

つまり、このために使用される側が情報を一元管理し、使用する側がそれを参照することが共有である。このとき、参照したりされたりするために「名前」を付けたほうがよい。この「名前」をつけられた対象をJDMF/M-92では名前付きオブジェクトと呼ぶ。

つぎに、共有の観点から以下のことが言える。

(1) リソース非共有型では共通のものを参照しているが、共有されていない。

(2) 一方向リソース共有型では共通のものを参照、共有している。

(3) 双方向リソース共有型ではお互いを参照しあっている。

ここで、一方向型と双方向型と異なる点は、双方向型の「このムービーはどのサンプルが必要である」という関係は必要であるが、「サンプルがどのムービーに必要とされているか」という関係は必要ないという点である。

JDMF/M-92では、その設計ガイドラインで、このような単方向の参照関係の場合、図 4 のような双方向の参照関係に設計しなおすように指示しているが、このような場合は、敢えて双方向の関係に設計し直す必要はないと我々は考える。

7. 今後の展望

本報告ではマルチメディアデータをJDMFに実装する際のモデル化に関する問題点を挙げた。今後は、この問題点を考慮しつつ、JDMFのDBMSとともに、マルチメディアデータの実装システムを作成したいと思う。

参考文献

[1]JSA; データモデル機能 JDMF/MODEL-1992, 1992
 [2]Charles W Bachman; Data Structure Diagrams, SIGBDP, Vol.1, No.2, 1969
 [3]Ryosuke Hotaka, Michael Björn; Data Oriented Approach to Business Information Modelling, ICODP-93, 1993
 [4] Yao Hui, Graphic Multi-Precision Dynamic Schema Design Interface, 情報処理学会第48回全国大会講演論文集掲載予定稿, 1994