

マルチメディア文書の標準データベース格納構造の考察

増永良文

図書館情報大学図書館情報学部

本稿ではODA(Open Document Architecture, 開放型文書体系)で規定されるマルチメディア文書をOODBMS(Object-Oriented database Management System, オブジェクト指向データベース管理システム)に格納しようという目的からODA文書の共通論理構造を分析し, その格納可能性を議論した. その結果, ODA共通論理構造中の, ある論理構成体の直属従属体が従属生成子(GFS)により, SEQ(順序), OPT(任意選択), CHO(択一), REP(反復), REP-CHO(反復-択一)関係で規定されているとき, それらの関係をどう表すかが問題であることを明かにし, それらの関係を各々どのようにすればOODBMSで表現可能かを明かにした. この結果ODAでは最も論理構造の複雑なAE.1136で規定されるレベル3の文書をOODBMSに格納する体系を明かにした. 本稿での結果の一部は現在オブジェクト指向DBMSであるONTOSを用いて実装中である.

Consideration of Standard Database Storage Structure of Multimedia Documents

Yoshifumi Masunaga

University of Library and Information Science

In this paper, the structure of multimedia documents characterized by ODA (Open Document Architecture) is analyzed to investigate the possibility of translating the ODA documents into the documents which can be stored in OODBMSs (Object-Oriented Database Management Systems). It is made clear that the relations such as SEQ, OPT, CHO, REP and REP-CHO which are used in the generator-for-subordinates (GFS) are not directly supported by OODBMSs. It is shown that these relations can be translated into equivalent representations of OODBMS database schemata so that ODA's most complex document, i.e. AE.1136-based documents can be stored into OODBMSs. The result of this study is partly under implementation using ONTOS, a commercial OODBMS.

1 まえがき

ODA(Open Document Architecture; 開放型文書体系)はマルチメディア文書の体系(architecture)と交換形式を規定した国際標準として知られ、ISO/IEC JTC1により作業がなされ、1989年にISO 8613として国際標準と制定された。ODA文書は共通及び特定論理構造と共通及び特定割付構造、割付体裁と表示体裁などを有する。AE.1111, AE.1126, AE.1136の3種に分類されたマルチメディア文書は文字内容、幾何図形内容、ラスタ図形内容を含む様々なマルチメディア文書を表現できるよう構造化することが出来き、大きな意義をもっている[1]。

しかし、ODAはマルチメディア文書の交換(ファイル転送や電子メールによる交換、あるいはフロッピディスクによる交換)を主目的としており、マルチメディア文書をデータベースに格納し、組織体での共有資源としてそれを有効利用していこうということを目的として制定されたものではない。従って、ODAによるマルチメディア文書の構造化は必ずしも我々が想定しているマルチメディア文書のデータベース化という観点からは適切な構造化とはなっていない。つまり、データベースの技術的な観点からはマルチメディア文書はオブジェクト指向データベースに格納することが適切であることが知られているが[2,3]、ODAが示しているマルチメディア文書の構造は、そのままの表現ではオブジェクト指向データベースには格納できないということである。

そこで、本研究ではマルチメディア文書の論理構造に視点をあて、ODAのどのような表現がオブジェクト指向データベースへの格納に際し問題となるのかを明かにし、問題となるODA表現をどのようにに変換することによってマルチメディア文書をオブジェクト指向データベースに格納することが出来るかを明かにする。本研究では様々なマルチメディア文書をデータベース化する目的で、ODAでは一番論理構造が複雑なレベル3の文書の構造を議論の対象としている。勿論レベル3文書のODA論理表現はそのままではオブジェクト指向データベースの格納表現と合わず、それに等価なオブジェクト指向データベースの格納表現(以下OODB表現という)を示す。

この結果は、ODA文書のデータベース格納によるマルチメディア文書の分散共有化システムを構築可能とするための基礎体系を提供するものである。なお、本研究でマルチメディア文書の論理構造に特に着目し議論しているのは、ODAは文書作成者が論理構造を指定すると文書処理システムが自動的に割付構造を作成する処理モデルを前提に文書の交換形式を規定しているからである[4]。

以下、次章でODAの論理構造についてまとめ、3章でODA論理構造のオブジェクト指向データベース化における問題点の列挙、それを受け4章ではODA表現のOODB表現への変換法、そして5章で結論を述べる。

2 ODAの論理構造について

2.1 レベル3文書交換形式(AE.1136)

AE.1136は高度な論理的階層構造を持ち、テーブルやフォームを含む複雑な割付を持つ文書を交換するためのプロフィールである。AE.1136は文字内容、ラスタ図形内容、及び幾何図形内容を含む。

AE.1136は次に示すような論理構造上の特徴を有している：

- ・文書の論理構造(章、節など)の階層化
- ・章/節/脚注番号、図番号の参照機能
- ・論理構造としてのテーブル、フォームを持つ
- ・論理構造としてリストを持つ

図1にAE.1136の論理構造の概略を、図2にその例を示す。論理構造は基本的に部品展開の関係(IS-PART-OFの関係)、例えばドキュメントは番号付きセグメントとパッセージからなるという具合に読む、を表している。また図2の例では図番号の参照を表すためテキスト中に図番号が挿入されている様子や、図と同じ論理構造上の位置にテーブルが置かれている様子、あるいはリストが使用されている様子などが示されている。

2.2 文書構造—論理構造—

2.2.1 論理構造ダイアグラム

ODA文書(AE.1136)の構造、ここでは論理構造に着目、はダイアグラム(関係図)で表される。ダイアグラムはそのプロフィールで使用できる論理構造と構成体制約(constituent constraint)で使用できる割付体裁、表示体裁についての概要を示している。その様式をAE.1136文書に関してその論理構造の一部を示せば、図3に示されるようになる。図では構成体制約同士が(構成体制約の)従属生成子(generator-for-subordinates, GFS)に従って、木構造で従属している。

ここでダイアグラムの意味しているところをより具体的に把握するために更に説明を加える：

DocumentLogicalRoot(文書論理根)は文書の最上位の構成体制約であり、これにどのような構成体制約が従属するかはその従属生成子(GFS; generator-for-subordinates)、この詳細が以下で述べる抽象構文を用いて規定されている、によって決まる。AE.1136では一般に文書論理根の直属従属体はPassageとNumberedSegmentの一つ以上の構成体制約の任意列からなる。Passage(パッセージ)の典型的な使用法は；報告書の表題ページに置かれる内容、目次や序文の内容、文書の本文、付録や用語集や索引の内容、文書の章の中に異なった単位として挿入されるテキストを伴う図である。Passageの直属従属体は一般に任意選択のTitle構成体制約と、これに続く次に示す一つ以上の構成体制約の任意列からなる：

Paragraph, NumberedSegment, Paragraph, Figure, Table, NumberedList, UnnumberedList, DefinitionList, BodyText, BodyRaster, BodyGeometric.

一方、NumberedSegment(番号付きセグメント)は識別子で区別される文書内の論理的な単位を表す構成体制約である。この直属従属体は、必須のNumber構成体制約(これはNumberedSegmentの識別子を与えるもの)と任意選択のTitle構成体制約がこの順で続き、更に次に示す一つ以上の構成体制約の任意列が続く：

Paragraph, NumberedSegment, Paragraph, Figure, Table, NumberedList, UnnumberedList, DefinitionList, BodyText, BodyRaster, BodyGeometric.

2. 2. 2 共通及び特定論理構造図

一方、ANS.1による抽象構文形式を用いた、処理可能(processable)文書(つまり論理構造が規定され編集可能な文書)体系(PDA)に従った記述法でレベル3文書交換形式を記述しておくことは共通(generic)及び特定(specific)論理構造を正確に知るうえで必要である。図4にa)例題文書の一部、b)その例題文書が対象とする文書のクラスを表すAE.1136共通論理構造図の一部、c)同特定論理構造図の一部、そして図5に上記共通論理構造を表す抽象構文の一部を示す。図4b)あるいは図5に見るように、ある文書のクラスが決まると、そのクラスに属するいかなる特定文書の論理構造も記述できるような共通論理構造が決まる。ここでは以下の議論のためその構造を記述している共通論理構造図のアークにSEQ, OPT, REP, REP-CHOという属性が付与していることに注目する。

3 ODA論理構造のオブジェクト指向データベース化における問題点

3. 1 概論

ODAマルチメディア文書の共通論理構造は、その一番レベルの高いレベル3文書交換形式(AE.1136)に着目すれば、例えば図5のように表現された。この表現は「共通」の名が示すとおり想定した文書のクラスに属するあらゆる文書の論理的構造をgenericに規定するものであり、データベースの観点から表現すればデータベースのスキーマ(schema, このインスタンスを語るのではなくそのメタ構造を表している、の意味)を規定していることになる。従って、もしその構造がそのままオブジェクト指向データベースのデータベーススキーマとして受け付けられるものであるならば、与えられたマルチメディア文書をその共通論理構造にしたがって分析し特定論理構造を見いだせばそれはそのままインスタンスとしてのオブジェクトとしてオブジェクト指向データベースに格納できることになり、理想的である。しかし、分析してみるとそうではない。マルチメディア文書の構造を論理的に捉えようとする基本的な考え方はODAとOODBで異なるものではないが、ODA表現は幾つかの点でそのままOODB表現とはならない。以下、具体的に一体どのようなODA表現がOODB表現となっていないのか、OODB表現となっていないODA表現は等価なOODB表現に変換可能か、もしそうなら等価なOODB表現はどのようにして得られるかなどについて議論する。

3. 2 オブジェクト指向DBMSへのODA文書格納上の問題点

現在市販のオブジェクト指向DBMS（以下OODBMS）は相当数にのぼる[3]。中でも応用プログラミング言語として、永続化したオブジェクト指向プログラミング言語C++をサポートするものが多数を占める。本稿ではODA文書のデータベース化の対象となるOODBMSも、このような状況から多数派のOODBMSを想定して議論を進めることとする。とくに議論を明確にすることが必要な場合はその中からONTOS[5]を用いそれを行う。

さて、ODAの共通論理構造を分析するとOODBMSがそれをデータベーススキーマとしてそのまま受け入れることが出来るかという問題は次のようにまとめられる：

- 1) 一般に論理構成体Aが論理構成体B, C, ..., Dを直属従属体として持つという関係をどう表すか。またそのとき、B, C, ..., Dのうち少なくとも一つは非空でなければならない、というODAの制約をどう表すか。
- 2) ある共通論理構造中、ある論理構成体の直属従属体が従属生成子(GFS)により、SEQ（順序）関係で規定されているという関係をどう表すか。
- 3) ある共通論理構造中、ある論理構成体の直属従属体が従属生成子(GFS)により、OPT（任意選択）関係で規定されているという関係をどう表すか。
- 4) ある共通論理構造中、ある論理構成体の直属従属体が従属生成子(GFS)により、CHO（択一）関係で規定されているという関係をどう表すか。
- 5) ある共通論理構造中、ある論理構成体の直属従属体が従属生成子(GFS)により、REP（反復）関係で規定されているという関係をどう表すか。
- 6) ある共通論理構造中、ある論理構成体の直属従属体が従属生成子(GFS)により、REP-CHO（反復-択一）関係で規定されているという関係をどう表すか。

次章でこれらの問題を逐一議論する。

4 ODA表現のOODB表現への変換

4. 1 IS-PART-OF関係の表現ほかの取扱

構成体制約とその直属従属体間の関係は、前述したがオブジェクト指向分析で知られているIS-PART-OFの関係（特にassembly-partsの関係[6]）と考えることが出来る。この関係はオブジェクトとオブジェクトの参照関係としてONTOSでもそのまま表現可能である。

ところで、前記1)項で述べた、ある論理構成体制約の直属従属体の少なくとも1つは非空でないといけないという制約は、文書がODA文書であろうがなかろうが実世界の文書であるならば持たねばならない制約であり、もっともな制約である。しかし、一般にOODBMSではあるオブジェクトのメンバ変数の値が全て空であってはいけない、という制約をデータベーススキーマ定義時に表現できない。つまりOODBMSでは現在その制約を、指定したいならばアプリケーションプログラムとして書くしかない。一方、OODBMS中の文書オブジェクトはもしそれが実世界の文書を忠実に表現しているものならばODAのいうその制約は自ずと満たされているのではないかと期待される。従って、ODAが提示しているこの制約の記述はこれからのOODBMS機能への要求事項と捉えるてここでの結論としたい。

4. 2 SEQ 関係のサポート

文書の共通論理構造で構成体制約間の論理的順序を表しているのがSEQ（順序）関係である。例えばあるクラスのODA文書では番号の次に（章）タイトルが来て、それに続いてパラグラフや図などが来るというような順序が本質的であるなら、このような順序を表すのがSEQである。しかし、一般にOODBMSではメンバ変数の順序を指定したりそれに意味を持たせたりすることは出来ない。例えば、クラスAがクラスBとクラスCを値域とするメンバ変数v1, v2を持っていたとした場合、OODBMSでは変数v1, v2の順にそれらを定義しようが、v2, v1の順に定義しようがどちらでもよい。しかし、文書を表すにはその構成体制約の論理的順序が大事なわけで、それを規定しているSEQ関係を無視してOODBMSに格納しては問題が生じよう。そこで何らかの対策が必要になる。

本稿ではこの論理的順序を表すための目的に、新しくメンバ変数をして導入して問題解決を図る。例えば、ある共通論理構造のクラスNumberedSegmentの定義においては、メンバ変数priv_number, priv_title, priv_01xに加えて、それら直属従属体の論理順序を格納するための変数priv_seqを新しく導入する。この変数の型は具体的にはONTOSのClient Libraryの中からAggregate classesの一つであるclass Dictionaryを選ぶこととする。ONTOSでのクラスDictionaryの定義例は次のようである：

```
Dictionary (Type* indexSpec,
            Type* memberSpec,
            Boolean isOrdered=TRUE,
            Boolean hasDuplicates=FALSE,
            char* name=0);
```

つまり、Dictionaryはaggregateクラスの一つであるクラスArrayとどこが違うかと言えば、cardinalityを予め決めておく必要がない、ランダムアクセスが出来る、キー空間を別に持つ、キーは整数でなくてもよい、点を挙げられる。本稿ではとくにDictionaryのcardinalityを予め決めておく必要がない特徴に着目して、ArrayではなくDictionaryを採用している。(例えば一般にNumberedSegmentが何章からなるかは事前に決めようがない。)

さて、ある文書クラスのあるマルチメディア文書が与えられると特定論理構造が得られる。例えばその特定論理構造のNumberedSegmentの部分が次のようであったとしよう：

- 第1章
 - 第1章番号(Number)
 - 第1章タイトル(Tiele)
 - 第1章序文(Paragraph)
 - 1. 1節(NumberedSegment)
 - 1. 2節(NumberedSegment)

このとき、この文書を表しているクラスNumberedSegmentのインスタンスのメンバ変数priv_seqの値は次のようになろう(ここでindexSpecのTypeはInteger, memberSpecのTypeはObjectとする)：

indexSpec	memberSpec
1	第1章番号(Number)
2	第1章タイトル(Tiele)
3	第1章序文(Paragraph)
4	1. 1節(NumberedSegment)
5	1. 2節(NumberedSegment)

このようなpriv_seqの値は文書が特定されれば決まるので、それにしたがって入力すればよい。

4.3 OPT関係のサポート

OPT(任意選択)関係はある共通論理構造が与えられたとき、それにより規定されるある特定論理構造で直属従属体が従属生成子のOPT関係で規定されている時、その特定文書でその直属従属体(である論理構成体)があればあってよく、なければなくてもよいという関係をいっている。典型的な例として論理構成体パッセージ(passage)をとれば、ある共通論理構造ではPassageは直属従属体Titleとparagraph他をこの順序で持つが、Titleは任意選択でタイトルがないパッセージもあればあるパッセージがあってもよい。

この関係はOODBMSでは問題なく表現できると考えられる。例えば論理構成体Passageを表すためOODBMSでクラスPassageを定義するが、それはメンバ変数priv_titleなどを持つ。このとき文書が特定されると、もしその文書のあるパッセージがタイトルを持っていればメンバ変数priv_titleはそのタイトルに対応するオブジェクトをその値として持ち、そうでなければ空値を持つべき。このようすればOPTの関係はOODBMSではそのサポートに特に問題はないと考えられる。

4.4 CHO関係のサポート

CHO(択一)の関係はある共通論理構造が与えられたとき、ある論理構成体の直属構成体として複数

の論理構成体の候補の中から一つを選んでそれとすることが出来る、という関係をいつている。CHOのODAでの典型的な使用例は、REP-CHO（後述）ではなくCHOが単独に使われている場合、ある共通論理構造で論理構成体Figureは論理構成体Artworkか論理構成体Formのどちらかであるという択一の関係にあることを示す場合に見られる。

さて、ある共通論理構造でCHOが指定されている場合、それに対応するOODBMSでのCHO関係のサポートには、オブジェクト指向プログラミング言語C++に基づいたOODBMSでは次の2通りの方法が考えられる：

- a) 共用体(union)を使用する方法
 - b) 仮想関数(virtual function)を使用する方法
- 以下この二つの代替案について議論する。

4. 4. 1 共用体を使用する方法

共用体Unionは同時には1つのタイプのデータしか採れないが、様々なタイプのデータをメンバ変数に採らせたときに使用する。したがってODAでのCHO関係を表現しようとするとき自然に考えられる方法である。ただし、Unionでは現在メンバ変数がどのタイプのオブジェクトをとっているかは、プログラマがその責任で管理しなければならないという構造になっている。

さて、上記の例を用いて議論すれば、論理構成体Figureを表すためOODBMSにクラスFigureを定義するが、そのための定義を一般的にC++の構文で書けば次のようになる：

```
enum figtype {ARTWORK, FORM};
class Figure : public Object {
public:
    figtype type;
    union{
        Artwork* art;
        Form* form; };
    Number* num;
    Caption* cap;
    .....
};
```

すなわち、クラスFigureの中で、ArtworkとFormが共用体として宣言されている。figtype型の変数typeで値としてARTWORKかFORMを記憶し、同じメモリ領域をARTWORKとして記憶した場合はArtwork*型、FORMとして記憶した場合はForm*型として変数typeが使用できる。

b) 仮想関数を使用する方法.

Artwork型かForm型のオブジェクトのアドレスを入れる変数をsfigとする。つまりsfigにはArtwork*型の値でもForm*型の値でもどちらでも入ることが出来る変数sfigを導入する。そうするために、人工的にArtwork型とForm型の共通のスーパークラスを定義し、sfigをそのスーパークラスのオブジェクトへのポインタを値として採るように定義する。ここではその共通のスーパークラスをSelectFigとする。しかし、このままの状態ではsfigが、例えばArtwork型の変数を示していてもsfig (のオブジェクト) にメッセージを送るとC++はSelectFigのクラスの中の対応するメソッドを実行しようとする。これを避けるために仮想関数を用いる。つまりArtworkとFormで共通のメンバ関数があればそれらをクラスSelectFigでvirtualとして宣言する。こうすることによって実質的に共用体で行うことと同じことを実行できる。その結果OODBMSでODAのCHO関係をサポートできる。図6にクラスFigure, SelectFig, Artwork, Form間のクラス階層関係を示す。

4. 4. 3 比較

さて、共用体、仮想関数いずれを用いてもODAのCHO（択一）の関係は表現できることが明らかとなっ

た。しかし、ここであえて両者を比較してどちらを使用してその表現をOODB表現化するかを議論する。まず、共用体を用いるところはCHO関係をどう表現するかというこの場合だけである。共用体は仮想関数を用いる方法に比べてクラス階層に手を加える必要はないが、前述のようにメンバ変数が現在どのタイプのオブジェクトをとっているかは、プログラマがその責任で管理しなければならない。一方仮想関数を使用する方法では人工的に一つスーパークラスを導入することは必要であるがプログラマへの負担は一切ない。また、例えばCoad & Yourdonのオブジェクト指向分析でのOOA記法[6]で共有体を表すことは出来ないが仮想関数アプローチでのクラススキーマは無理なく表現できる。本稿ではこのような理由から仮想関数を使用する方法を採用する。

4.5 REP関係のサポート

REP（反復）の典型例は共通論理構造での論理構成体Titleと同BodyTextの関係、あるいは論理構成体Tableと同Rowの関係などがある。さて、反復には論理的順序があり、また何回反復するかは不定である（少なくとも一回以上）。従ってODAでのCHO関係の表現には、OODBMSではあらかじめcardinalityを決めないと定義できないクラスArrayではなくクラスDictionaryを用い、そのサブクラスとしてクラスBodyTextDictionaryを定義してそれを使用することを考える。論理構成体Titleが論理構成体BodyTextを反復して持つ場合、OODBMSではクラスTitleにクラスBodyTextDictionaryを値域とする変数（例えばpriv_bodyText）を導入してODAの反復関係をサポートする。以下にクラスTitleの定義の概略を示す（ここではONTOSの記法を用いて、まずオブジェクトの直接参照の書き方、次いで透過参照の書き方に従った概略を示す）：

```
class Title: public Object {
    private: BodyTextDictionary* priv_bodyText;
};
```

もし、オブジェクトの透過参照の書き方に従えば次のようになろう。

```
class Title: public Object {
    private: Ref* priv_bodyText;
    public: BodyTextDictionary* getPrivBodyText();
};
```

```
BodyTextDictionary* getPrivBodyText()
{
    return ((BodyTextDictionary*) priv_bodyText->Binding());
};
```

4.6 REP-CHO関係のサポート

REPとCHO関係の組み合わせであるが、ODA文書では出現する場所が多い。例えば、ある共通論理構造では論理構成体paragraphは同bodyText, bodyRaster, bodyGeometric, phrase, footnote, referenceを反復-択一しよう。つまり、ある特定文書の序文(paragraph)が序文1 (bodyText), 脚注1 (footnote), 序文2 (bodyText)からなる場合、その構造は上述の共通論理構造の論理構成体paragraphがその直属構成体としてbodyTextを最初に、次いでfootnoteを、更に再びbodyTextを採ったということである。

この論理的順序関係をOODBMSで表現するには、SEQ関係の場合と同様に（この例題ではクラスParagraphで）メンバ変数priv_seqを定義してデータベース中に保持するようにすれば実現出来る。

5 おわりに

本稿ではODAで規定されるマルチメディア文書をOODBMSに格納しようという目的からODA文書の共通論理構造を分析し、その格納可能性を議論した。その結果、共通論理構造中のある論理構成体の直属従属体から従属生成子(GFS)により、SEQ（順序）、OPT（任意選択）、CHO（択一）、REP（反復）、REP-CHO（反復-択一）関係で規定されているという関係をどう表すかという問題が本質的であることを明かにし、各々それらの関係をどのようにすればOODBMSで表現可能かを明かにした。この結果ODAで

は一番論理構造の複雑なAE.1136で規定される文書をOODBMSに格納する体系が明かになった。本稿での結果の一部は現在オブジェクト指向DBMSであるONTOSを用いて実装中である。

本研究を今後どのように発展させてゆくかには幾つかの方向がある。まず最初にOODBMSに格納されたODA文書を自由に検索・更新できる体系を作り上げることがある。このためには文書の論理構造を自由に辿れる文書アクセスのモデル、更に文書の意味構造[8]を表現しそれを利用したアクセスモデルの開発などが課題となる。

ほかにOODBMSをODAマルチメディア文書の格納基地であると捉えるとき、一旦格納された、あるいはOODBMSのもとで生成された文書をどのようにしてODAのフォーマットで提供できるかという問題がある。この問題はマルチメディア文書を表すOODBMSデータベーススキーマをODA共通論理構造に変換する問題であり、本稿で議論した変換の逆変換を議論する問題である。

別の問題として、ODA文書はODIF(Open Document Interchange Format, 開放型文書交換形式)に従ってファイルに格納されたりネットワークを介して文書交換される。このデータ列(datastream)はヘッダ部と本体部からなり、ヘッダ部の情報を使った文書のフィルタリングや検索などが考えられる。このような観点からODA文書をOODBMSにどのように格納すべきかを考察することも重要になる。

AE.1136でカバーされるマルチメディア文書は相当に複雑でその記述能力は強力であるものの、今日のマルチメディア技術の発展からみれば、より記述能力の高い体系が望まれる。そのため超ODA制定の作業が行われている[7]。現在マルチメディアDBMS自体確立した体系をなしているものではないが、本稿で行った議論を超ODAにまで拡張することを試みることは意義があると考えられる。

なお、本稿図1から図5は文献[1]、図6は文献[9]からの抜粋であることを末尾ながら明記する。

【謝辞】

本研究を進めるにあたり研究の一部を遂行した図書館情報大学卒業生杉山清美氏、その過程でCHO関係のサポートで仮想関数を使用する方法を提案した本学助手鎮目浩輔氏、およびODAにつき貴重なご助言と資料を提供下さった日本ユニシス(株)若鳥陸夫氏に深謝する。

【文献】

- [1] 開放型システム間相互接続の基本参照モデル JIS X 5003-1987 参考 S 007(V2.0) 文書交換形式(O DA)実装規約 規約編, (財)日本規格協会, 平成3年3月1日. 203p.
- [2] 増永良文: マルチメディアデータベース総論, 情報処理, Vol.28, No.6, pp.671-687, 1987.
- [3] 増永良文: 次世代データベースシステムとしてのオブジェクト指向データベースシステム, 情報処理, Vol.32, No.5, pp.602-613, 1991.
- [4] 棟上昭男(編): OSIの応用: 高度情報化の実現に向けて 開放型システム間相互接続, 日本規格協会, 308p., 1987.
- [5] ONTOS DB2.2 Developer's Guide, Ontologic Inc. 282p. 1992.
- [6] Coad, P., Yourdon, E.: Object-Oriented Analysis (book), Prentice-Hall International, Inc., 233p., 1991.
- [7] 若鳥陸夫: 超ODAによる意味関係及び時系列関係の交換, 電子情報通信学会技術研究報告(信学技報) DE92-41, 1993.3.
- [8] 増永良文: 教科書のマルチメディアデータベース化による電子図書館の作成研究, 情報処理学会第46回(平成5年前期)全国大会講演論文集(4), 1G-7, p.4-129, 1993.3.
- [9] 杉山清美: オブジェクト指向マルチメディアデータベースシステムの研究, 図書館情報大学卒業論文, 34p. 1993.
- [10] 佐藤和洋, 絹川博之, 大町一彦: オフィス文書の標準化と文書データベースの研究動向, 情報処理, Vol.28, No.6, pp.710-720, 1987.
- [11] Rosenberg, J., Sherman, M.S., Marks, A., Giuffrida, F.: Translating Among Processable Multi-media Document Formats Using ODA, Proc. of the ACM Conference on Document Processing Systems, pp.61-70, Dec. 1988.

例題文書

この文書は、文書交換形式(004)実体的書解読用のための例題文書(*)であり、レベル3文書交換形式(AE, 1136)で交換されることを対象としている。
この例題文書は全体2ページの文書で、第1ページは文章のみ、第2ページは文章とラスタ図形とテーブルで構成されている。

1. 第1章

この章では、例題文書の概要を示す。

1.1 論理構成

この例題文書は、次の論理的構成要素から構成されている。

- 一期番号と期タイトル
- 一期の本文
- タイトル
- 図形
- 序文
- テーブル
- 章番号と章タイトル
- 脚注
- 一期の本文
- フッタ (ページ番号)

1.2 特徴

文章中には脚注番号が指定でき、その脚注はページの下部に割付けられる。文章中の脚注番号とページ下部の脚注内の番号とは自動的に対応するようになっている。

本文中には、属性「連結」が指定されている。本文では、文章と図形(*)の存在が許されており、レベル1文書交換形式・レベル2文書交換形式では、図形を並行に割付けられている。また、割付けられるようページ番号はフッタと期タイトルが一つの区画に割付けられるように、章及び期のタイトルの中央に位置付けられるようになっている。

注1 四角の枠はページを示しており、この枠内が交換対象の文書である。
注2 ラスタ図形。

図 4 a) 例題文書の一部

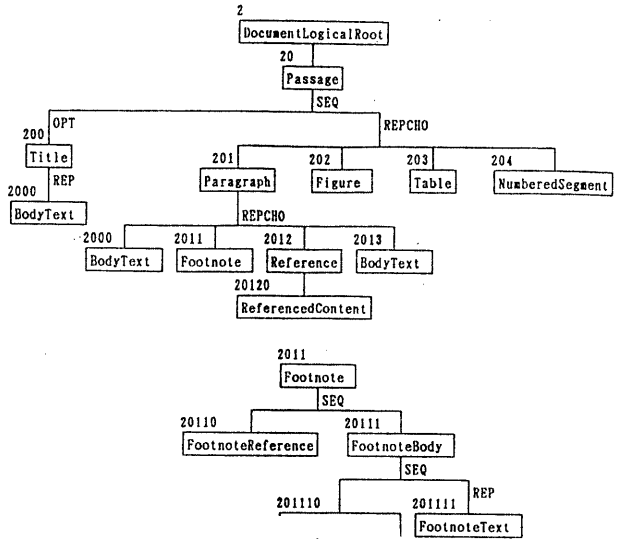


図 4 b) 例題文書の共通論理構造図の一部

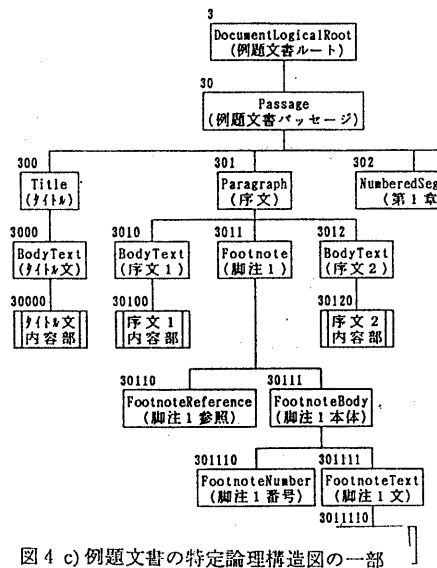


図 4 c) 例題文書の特定論理構造図の一部

```
logical-object-class {
  object-type      composite-logical-object(1),
  descriptor-body {
    object-class-identifier  "2 0 1 1",
    generator-for-subordinates { "SEQ("2 0 1 1 0","2 0 1 1 1") },
    bindings {
      [{"numbers-0", INC(B_REF(PREC(CURR_OBJ))
        ("numbers-0")) },
      {"numberstring-0", MK_STR(B_REF(CURR_OBJ)
        ("numbers-0"))}],
    application-comments  "6"
  }
}
```

図 5 図 4 例題文書の抽象構文の一部

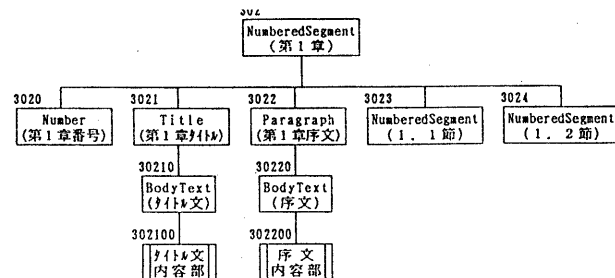


図 6 クラスFigure関連クラス階層