

システム開発のメタモデル —企業情報システムの場合—

奥悦史 葉木洋一 堀内 一

(株)日立製作所 ビジネスシステム開発センタ

企業情報システムについて、データ中心の方法論により、メタデータを組み立てる開発形態の利点を実現するためのメタモデルを提案する。まず、この成果物および開発自体のメタモデルを体系化するメタモデル枠組みを示す。この枠組みにメタデータの部品化要件を持つ基本メタモデルとモデリング対象の多様性に対応する応用メタモデルの枠を設定する。さらにこの応用メタモデルをカスタマイズする設計メタモデルを提案する。ここで特に実システムを集約オブジェクト群として表現することに着眼する。最後に設計メタモデルに基づき応用メタモデルを作成した事例により、システムの部品の管理やその組み立ての方法を規定できることを示す。

METAMODEL OF SYSTEM DEVELOPMENT ON BUSINESS APPLICATIONS

Etsushi Oku Youichi Hagi Hajime Horiuchi

Institute of Advanced Business Systems, Hitachi, Ltd.

We show the metamodel advantages of system development with Data Oriented Approach Methodology on business application. This paper proposes various metamodels and Metamodel Frame Work for giving a comprehension about their role's types. One of these types is Fundamental Metamodel with essential factors of metadata. And another is Applicable Metamodel that describes real system dealing with the various requirements. Furthermore it is given that Designing Metamodel yields customaization merits to various modeling domains. At last we show the same Applicable Metamodels that is created by using our Designing Metamodel as a sample case, and we regard that set of metamodels as the discription of parts management and construction for system development.

1 緒言

企業情報システムは企業の業務の領域（以下、業務領域）とコンピュータシステムの領域（以下、システム領域）にまたがるシステムである。業務領域とシステム領域を有機的に組織化し、企業にメリットをもたらす企業情報システムを構築することが望まれる。

今回は企業情報システムの開発において、データ中心の方法論に基づき、業務領域とシステム領域各々に固有の機能を有効に結び付けることを確実にする開発形態を目標に、このシステム開発のメタモデルを明らかにする。

本論文では、まず始めに一般的に受け入れられている機能分析中心の方法論に基づく開発形態と比較して、今回あるべき姿と考えるメタデータを部品として、それを組み立てる開発形態を示す。これによりシステム開発をメタモデルの開発という観点に置き換える。

次に、このシステム開発自体を表現するレベルに及ぶ一貫したメタモデルの体系を与えるメタモデル枠組みを提示する。このメタモデル枠組みに従い、まず与件としてデータの基本要件を満たす具体的メタモデルとシステム開発作業に直接機能する応用メタモデルの枠を設定する。さらに、応用メタモデルを具体的に得るため、設計メタモデルの枠組みを示す。

次に、応用メタモデルの設計メタモデルを提案する。これはシステム開発メタモデルの核となるものであり、特に開発の成果物としてのメタモデルに含まれるオブジェクトを集約物として与える機能を持つ。

さらに、提案した設計メタモデルに基づき、実際に応用モデルを設計し、具体化した事例を示す。

2 データ中心のシステム 開発形態

2.1 機能中心とデータ中心の 開発方法論

システム開発方法論は機能分析中心の方法論とデータ中心の方法論に大別できる。このうちデータ中心の方法論には堀内などが提案

しているものがある。[1] しかしその有効性が示されているにも関わらず、一般的には機能分析中心の開発形態の方が広く受け入れられている。

そこで、機能分析中心の方法論による開発形態と比較して、データ中心に基づいたあるべき姿としての開発形態を示す。

(1) 機能分析中心の方法論

まず、伝統的なシステム開発を図1に示す。ここでは業務領域からニーズの抽出を行ない、それに対応する機能を求め、その中からコンピュータシステムの中に組み込む機能を切り出す。そしてこれをプログラムなどの手段によりシステム領域の中に機能として実現する。

この一連の順序にはバリエーションがあるが、いづれにしてもこの様な開発では、機能という抽象レベルの高い概念を用いて、業務領域とシステム領域の間を結び付けている。

ここでシステムの基本構成物は機能とプログラムである。この開発形態の問題点は、機能という概念は抽象性が高いため、機能の切り出しや、その表現（記述）に比較的高い技量が要求されることである。そしてこの作業はシステム化要求ごとに行なわなければならない。従って、機能の確定度の低い場合や解釈の食い違いなどが生じ、業務領域とシステム領域の結合が阻害される場合が多い。

(2) データ中心の方法論

上記のような機能分析中心の方法論に対してデータ中心の方法論がある。この方法論では、メタデータをシステムの部品と考える。メタデータとは業務領域で現われる実データを規定する立場のデータであり、以下のものがある。

- 実データの型
- データ固有のライフサイクル（プロセス側面）
- データ固有の制約
- 画面、帳票、データ格納や制御機能などシステム領域を表現したデータ

この開発形態では図2に示すようにメタデータを企業情報システムの資源として準備

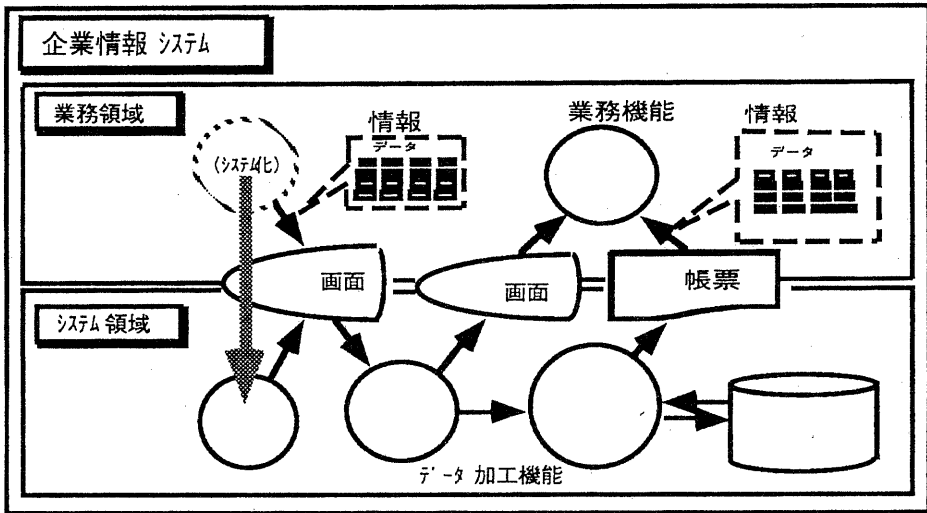


図1 機能分析中心のシステム開発概念

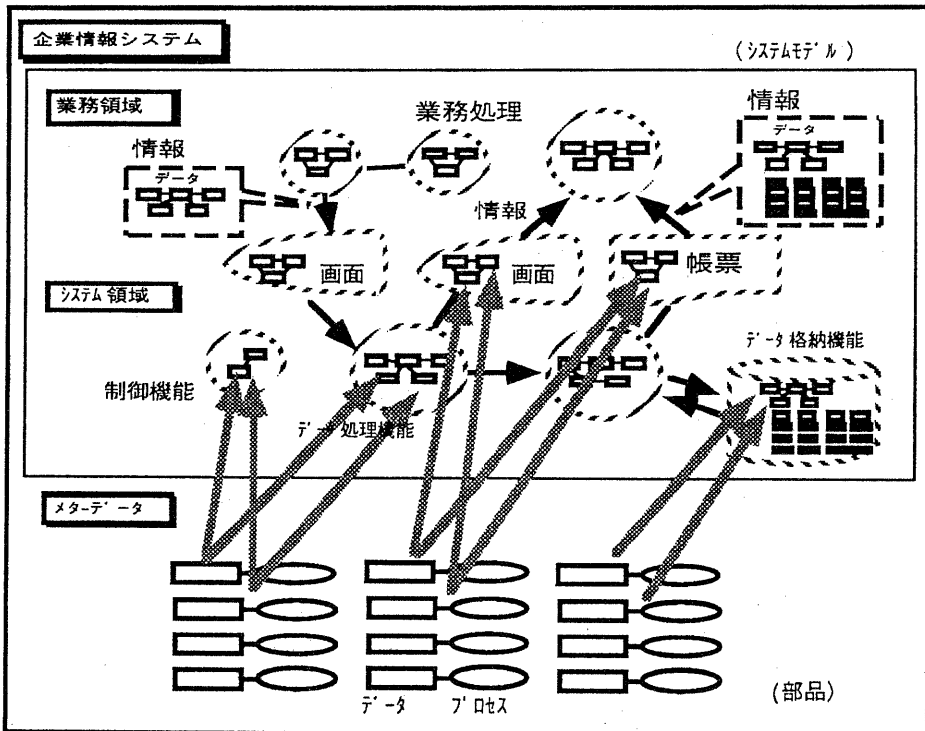


図2 データ中心のシステム開発概念

する。そしてシステム開発の段階でメタデータを部品として、これを組み立てる。

ここでシステムの基本構成物はメタデータである。メタデータは部品としての要件を具備したものであり、業務領域とシステム領域の両方にわたって洗い出されている。従って、システム化に当たっては、保証された部品の機能を組み立てるので、確実に全体としての機能を実現できる。

2. 2 開発成果物としてのメタモデル

上記の開発において、メタデータから組み立てられ、実システムをある側面から表現すべく組織化されたモデルをメタモデル（後で、システムモデルと定義する）と呼ぶ。これらメタモデルの全体は企業情報システムを表現したものである。データ中心の方法論でシステム開発（論理設計の段階）はメタデータを組み立て、メタモデルを作成することである。つまり、このメタモデルが開発成果物である。

3 システム開発のメタモデル枠組み

前章ではアプリケーションデータのメタレベルに位置するメタモデルについて述べた。このメタモデルは開発された実システムを表現している。これに対してシステム開発自体を表現するモデルを考え、システム開発の枠組み（Fram Work）により体系化する。

3. 1 システムモデルと機能メタモデル

開発成果物のメタモデルに対してある機能を持ったモデルがいくつかあるとして、これら各種の機能を備えたモデルの集合体でシステム開発自体を表現できると考える。

この開発自体のモデルは開発成果物を規定しているとして、この開発自体を表現したモデルは開発成果物（メタモデル）のメタモデルと考える。

ここで、実システムを表現したメタモデル（開発成果物）をシステムモデル、システ

ムモデルに機能するメタモデル（開発自体のモデル）を機能メタモデルと呼び区別する。

以下、この機能メタモデルを洗い出していく。

(1) 基本メタモデル

先の図2で示したようにシステムモデルはメタデータから組み立てるので、メタデータは部品としての要件を具備しなければならない。そこでシステムモデルを構成するメタデータの要件を持ったメタモデルが必要である。これを基本メタモデルとする。

メタデータの要件を持ったモデルとして堀内が示すメタ-メタモデルがある。[2]

これは以下の要件を備えている。

- データの構造
- データに固有のライフサイクル（プロセス側面）
- データに固有の制約

これらの要件を具備したメタデータは次の部品としての要件を持つといえる。

- 部品単位に構造とプロセス側面をカプセル化しているので、これを組み立てて機能を実現できる。
- 制約によりデータの一貫性が保証されているので、この部品により組み立てた機能はシステムとして整合性がとれる。

従って、上記のメタメタモデルを基本メタモデルとする。基本メタモデルの最も単純な運用は次のようになる。

- 一つの企業情報システムに唯一の基本メタモデルがある。
- 一つの基本メタモデルが多種のシステムモデルに対応して機能する。

(2) 応用メタモデル

システムモデルには実システムを記述する側面によって多種のバリエーションがあるが、システム開発でシステムモデルを作成する際には、この種類に応じた技法を用いる。この技法自体を表現するモデルがあるとすれば、これはシステムモデルに対して技法としての機能を持つと考える。

このモデルはシステムモデルを構成す

るオブジェクトの型を表現するデータ，すなわちシステムモデルに対するメタデータを組織化して与える。

従ってこのモデルはシステムモデルのメタモデルである。このメタモデルを応用メタモデルとする。応用メタモデルの最も単純な運用は次のとおりである。

- 一つの企業情報システムに多種類の応用メタモデルがある。
- 一つの種類の応用メタモデルが一つの種類のシステムモデルに対応して機能する。
- 応用メタモデルは企業情報システムあるいはシステム開発ごとにカスタマイズして準備される。

応用メタモデルは上記第三項のような特質を持つため，この段階では具体的なモデルを与えず，その枠組みのみを明らかにする。

以上のシステムモデル，基本メタモデルおよび応用メタモデルの枠組み全体を図3に示す。同図で垂直方向の軸はデータレベル，水平方向の軸はモデルの表現する対象領域である。基本メタモデルはデータについての最も一般的な要件であり，応用メタモデルよりも上位のデータレベルにあるので，その枠組みを最上位に配置する。そして，データレベルとして上位より機能レベルとシステムモデルレベル，機能レベルのサブレベルとして基本機能サブレベルと応用機能サブレベルを設定する。

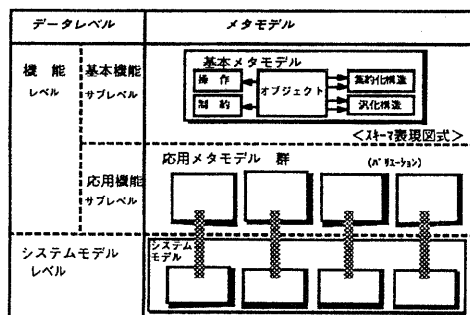


図3 メタモデル枠組み

3.3 応用メタモデル枠組み

さらに機能レベルの枠組みを検討する。この枠組みの中で，基本メタモデルは開発作業者が直接用いることを意図したものではなく，開発作業のためには応用メタモデルを与えている。これら二種のモデルのシステムモデルへの機能の仕方は次のように異なる。

- 基本メタモデルはシステムモデル中のオブジェクトの要件のみを持つことを目標に，最も単純な構造で構成する。
 - ◎システムモデルに対してはオブジェクトの要件として間接的に機能する。
- 応用メタモデルは実システムの多様な側面を写すことを目標に，多様なオブジェクト（システムモデルのオブジェクトの型）から構成する。
 - ◎システムモデルに対しては，その中にあるオブジェクトの型としてメタインスタンス対応し，直接的に機能する。

上のことを資源管理的視点から見ると，基本メタモデルとシステムモデルを直接メタインスタンス対応させたほうが冗長性が低い利点がある。しかし，本メタモデル枠組みではシステムモデルの直接のメタレベルに，基本メタモデルより多様なオブジェクトを持つ応用メタモデルの枠を設ける。この応用メタモデルはシステムモデルのオブジェクトの型から構成する。これによりシステムモデル作成の作業性確保を狙う。

システムモデルは多様なバリエーションを持ち，またこれは企業情報システムごとにカスタマイズされるものである。従って，応用メタモデルを自在に設計して応用機能レベルの枠に設定することが望まれる。

そこで図4に示すように，応用機能レベル枠組みの上位に，応用メタモデル設計のための設計メタモデルを配置する設計機能サブレベルの枠を設ける。設計メタモデルは以下の課題を持つ。

- システムモデルに基本メタモデルの持つ基本要件を具備させる機能を持つ応用メタモデルを設計する。
- システムモデルの多様性に応じて応用

メタモデルを設計する。

- 上記二点を両立させる。

特に、結果としてシステムモデルの部品が基本メタモデルの基本要件を具備したものになることが絶対条件である。次の章でこの設計メタモデルを具体的に提案する。

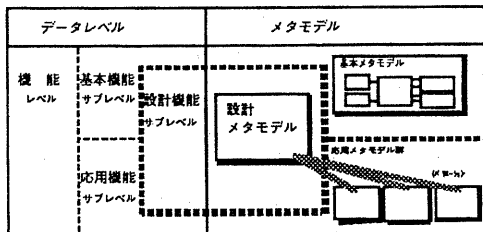


図4 設計メタモデル枠組み

4 応用メタモデルの設計メタモデル

前の章で示したニーズに基づき応用メタモデルの設計メタモデルを提案する。この設計メタモデルは本論文であるべき姿と考えるシステム開発のメタモデルの核になる。

4.1 設計メタモデルの着眼点

本設計メタモデルは、応用メタモデルに対し、集約 (Aggrigation) に重点をおいたシステムモデル作成機能を付与することに着眼したものである。この着眼点について図5の概念図で説明する。

(1) 集約オブジェクト

第一の方針として、後述するドメイン以外、全てのオブジェクトは集約されたオブジェクト、すなわち集約オブジェクトであると考える。集約とは次の性質を合わせたものである。

- 一つ以上のオブジェクトの機能を継承 (Inheritance) し、それらの機能の和以外の機能を新たに加えたオブジェクトを形成する。
- 集約オブジェクトとして、次の属性を

持つ。

◎下位構造：集約オブジェクト自らの一段下位のオブジェクトを指すデータ (ポインタ)

◎実行命題：集約オブジェクトの振る舞いの能動的記述

集約の構造自体は集約オブジェクトと同等の実体とは考えず、その属性と考える。

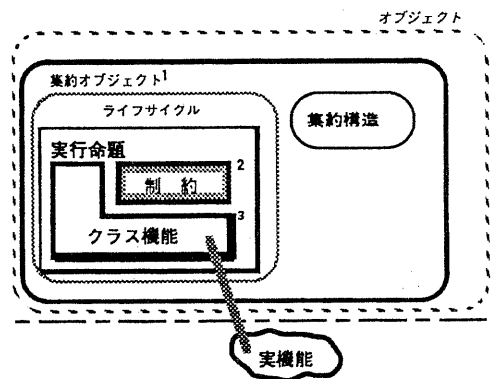


図5 設計メタモデルの基本概念

(2) オブジェクトの機能

集約オブジェクトの振る舞い (Life Cycle), すなわち動的側面を記述するデータを実行命題と呼ぶ。実行命題はオブジェクトが持つ機能を特定するものになる。

このようなオブジェクトの機能のうち、ある特定の種類 (タイプ) にはそのタイプに固有の機能 (オブジェクトの振る舞い) が存在していると考えられる。このような集約オブジェクト全体にわたる一般的な機能を記述したものを特にクラス機能と呼ぶ。本設計メタモデルでは、このクラス機能をシステム全般にわたり洗い出し、応用メタモデルのメタ情報として管理する方針をとる。また、クラス機能は、実世界の機能との関連を明確にして管理する。

(3) 制約

オブジェクトの一貫性を保証する制約条件の記述を制約と呼ぶ。制約はメタモデル (メタデータ) に欠くことのできないものとする方針をる。従って、応用メタモデルに次の制

5 応用メタモデルの設計事例

前章で示した設計メタモデルに従って
 応用メタモデルを設計した。この事例について
 設計の検討過程と設計結果に分けて説明する。

5.1 検討過程

応用メタモデルの設計事例を文末の付
 図に示すインスタンス表現図式を用いて説明す
 る。

(1) 業務側

まず、業務の側に着目して検討を始めた。
 図7では以下の検討過程を示している。

- ① ドメインを基底に置いた。
- ② 業務データをRDBで実装する方針でドメ
 ーンの集約オブジェクトとしてカラムを設定
 した。これは以下のようにRDBのカラム同
 様の機能を持つものとした。

- 実データを格納、保持する
- ドメインと対応し、格納に際してドメ
 ーン制約を受ける
- データの型を持ち、格納に際して型制
 約を受ける

- ③ カラムの集約オブジェクトとしてテー
 ブルを設定した。テーブルは以下のように
 RDBのテーブルと同様の機能を持つもの
 とした。

- カラムから表を構成し、これを
 一意に識別する
- タプル（行）を操作する

- ④ カラムの集約オブジェクトとして参照
 制約検証オブジェクトを設定した。

- 参照制約の検証を実行する
- この集約オブジェクトの制約の枠には
 参照制約を記述する

- ⑤ カラムの集約オブジェクトとして導出
 制約検証オブジェクトを設定した。

- 導出制約の検証を実行する
- この集約オブジェクトの制約の枠には
 導出制約を記述する

- ⑥ 上記二つの集約オブジェクトはテー
 ブル

に集約されるものとした。

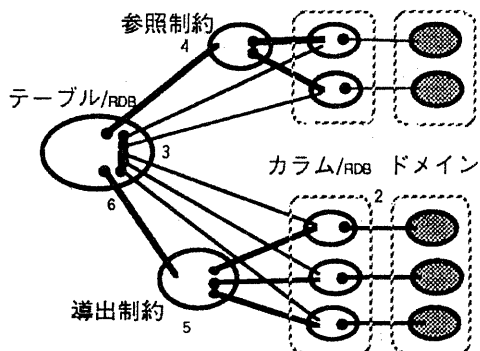


図7 応用機能の設計過程-1 / 4

(2) システム側

続いてシステム側に着目して検討した。

まず画面による入出力機能が必要とさ
 れるので、図8に示すように画面のモデルを決
 めた。

- ① 画面は画面構成、画面項目および画
 フィールドの各集約オブジェクトから構成
 されるものとした。
- ② 画面フィールドはカラム、制約検証のオブ
 ジェクトを集約するものとした。これによ
 り、画面フィールドでカラムその他のオブ
 ジェクトが持つ制約を継承するものとした。
- ③ 全ての画面フィールドは画面項目に対応す
 るデータを保持する機能を持つが、各々の
 画面項目で入力、出力その他の機能が異な
 っているので、更に画面フィールドの機能を
 洗い出し、数種類のサブタイプを規定した。

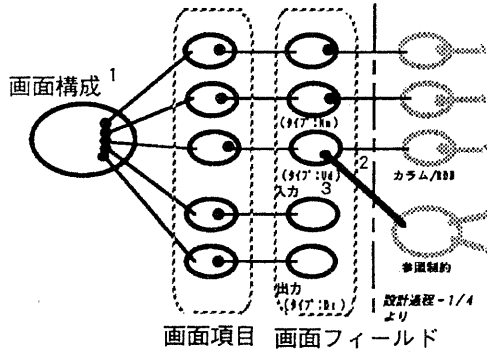


図8 応用機能の設計過程--2 / 4

さらに図9に移り、

- ④ システムに与えられる外部のイベントに着目し、これに対応する集約オブジェクトとして外部トランザクションを設定した。
- ⑤ 外部の下位集約物としてトランザクションフィールドを設定した。これは先の画面フィールド同様、データを保持する機能を持ち、特に入力タイプの画面フィールドからデータを受け取るものと規定した。
- ⑥ トランザクションフィールドは画面フィールドを集約したものとし、これによって画面フィールドがカラムなどから継承している制約をさらに自らに継承するものとした。

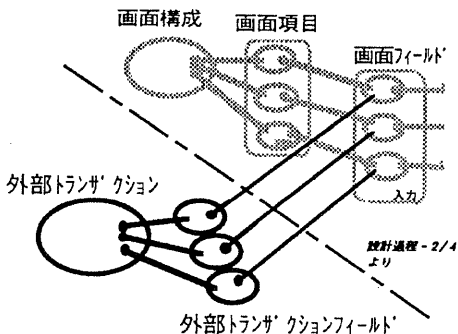


図9 応用機能の設計過程 -3 / 4

5. 2 設計結果

前の項で検討した結果を付図のスキーマ図式に従い、データモデル化した。その一例を図10及び図11に示す。

ここで前項の検討時よりもさらにリファインした点を以下に記す。

- ① 集約オブジェクトをデータタイプとして規定した。

- 前項で示した参照、導出の検証を行なう二つのオブジェクトは整理、統合して新たに制約チェッカーとして規定した。

- 画面フィールドとトランザクションフィールドなど、同種の機能タイプを整理して、サブタイプ構造で関連づけた。

- ② データタイプをグルーピングし、このグループに業務データ、スクリーンオブジェクトなどのオブジェクト-グループ-タイプを規定し、システムモデル対象領域のカテゴリ-を設定した。

これにより応用メタモデルが完成した。応用メタモデルは機能メタモデルとして、以下のようにシステムモデルに機能するものになった。

- 集約オブジェクトのタイプにより部品を規定している。
- 各々の集約オブジェクトはドメインを基本部品とした中間部品として規定している。
- 各集約オブジェクト相互の集約関連により、部品の組み立て方を規定している。
- システムモデル全体にわたる制約の継承関係を規定している。

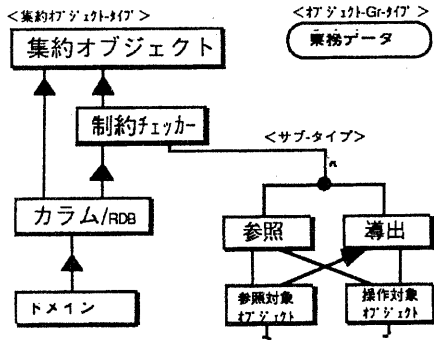


図10 業務データの応用メタモデル

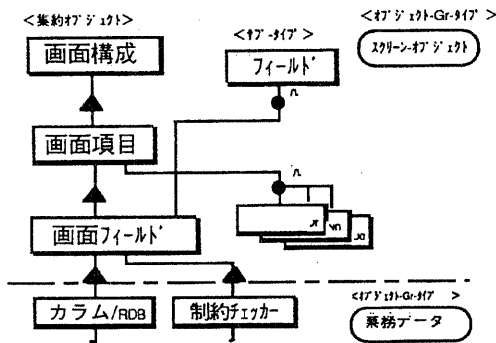


図11 スクリーンオブジェクトの応用メタモデル

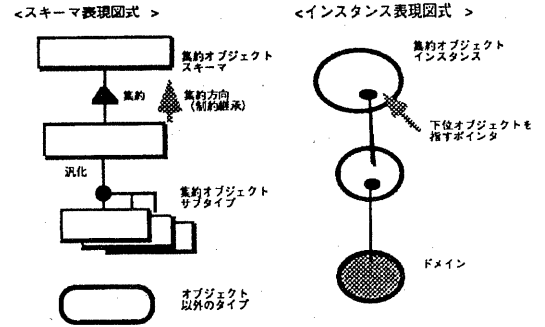
このように応用メタモデルは部品と組み立て方の一般形としてシステムモデルを規定する。また、制約の継承関係が一意に定まり、システムの整合性を保証できる。

6 まとめ

データ中心の方法論に基づき、メタデータを部品としてこれを組み立てる開発形態を示し、このシステム開発自体のメタモデル枠組みを明らかにした。さらにこのメタモデル枠組みの核となる、応用メタモデルの設計メタモデルを提案し、これによって応用メタモデルを具体的に設計し、この事例を示した。最後に設計した応用メタモデルが部品-組立て型の開発形態を規定していることを示した。

今回提案したメタモデル枠組みにより、

メタデータが企業情報システムの基底として位置づけられる。これにより適用領域とシステム領域をより確実に結び付けるシステム開発方法論の展開が示唆される。



付図 図式表現の凡例

文献

- [1] 堀内 一：データ中心システム設計，オーム社（1988）
- [2] 堀内 一：システム開発パラダイムと高水準データモデル，情報処理，32，9，pp.1014-1022（1991-9）