解説：

メタモデル標準化の意義と最新動向

Current Trends on Metamodel Standardization :

前編：－基本的概念と歴史的経過－

Part-1: Basic Concept and Historical Evolution of Metamodel

堀内　一　東京国際大学　hori@tiu.ac.jp
大林正晴　（株）管理工学研究所　obayashi@kthree.co.jp
藤川泰之　富士通（株）　fujikawa.yasuyu@jp.fujitsu.com

グローバルなビジネス連携を支えるシステムでは，ソフトウェアやモデルの柔軟性，整合性，共有性，拡張性などの確保が重要となる．そのため，標準的な分類体系，識別体系，オブジェクトパターン，メタデータ，など標準モデル構成要素（Standard Modeling Constructs）への準拠と，業界ごとの規範的ビジネスベストプラクティスへの準拠が必須となってくる．

オブジェクトモデル化言語 UML のグローバルな普及とデファクト化は，それら標準モデル構成要素への準拠性確保を通じて複数異モデルの統合や変換が容易となることを意味する．昨今，特に，統合手段としての「UML メタモデル」と，「メタモデル機能：MOF（Meta Object Facility）」への関心が高まっている．

本稿では，前編で，まず，メタモデルの基本的概念を紹介してから標準化動向を紹介し，今後，情報技術分野においてメタモデルが重要性を持つことなどを解説したい．後編では，メタモデル機能としての MOF と XMI について詳しく解説する予定である．

メタモデルの基本概念

◆メタとは

メタ（meta）とは，「超」，「典型」，あるいは「変態」，「変身」を表す接頭語である．"meta" がつく言葉は，日常使う言葉としてはなじみが薄い．せいぜい，metaphor（暗喩），metamorphose（変態させる），metaphysical（形而上），metalogic（超論理），くらいであろう．それも哲学的である．英和辞典を見ると，metabolic（代謝），metastatic（転移）など医学用語が多い．病気を組織の「変態」と見るからであろう．

情報処理分野では，メタ言語，メタ論理，メタモルフォーゼ，メタデータ，メタモデル，などが使われている．

さて，メタ言語は「言語を記述する言語」として，メタデータは「データを記述するデータ」と定義されてきた．同様に，メタモデルは「モデルを記述するモデル」となるのか，詳しく見てみよう．

◆メタ階層

「記述するもの」と「記述されるもの」との関係に基づく階層は限りなく反復が可能である．そこで，強制的な停止が必要となる．

一般に，メタ階層を構成させる基準，つまり上位のオブジェクトに何を記述させるべきかについては，(1) 汎化型（型と実現値），(2) 生成・操作規則，(3) 抽象型，(4) 管理属性，(5) コメント記述，などが考えられる．多くは (1) に基づく階層を採用している．メタ概念とメタ階層について最初に標準化を試みたのは，国際規格，IRDS（Information Resource Dictionary System，情報資源辞書システム：ISO/IEC 10728，1993 年）であろう [1]．

IRDS は，メタ階層反復の強制的停止を意図して「記述するもの」と「記述されるもの」との関係を「型と実現値」により 4 階層で示した（**図 -1**）．その 4 階層のメタ階層概念は，OMG（Object Management Group）の MOF や UML に引き継がれている（**図 -2**）．

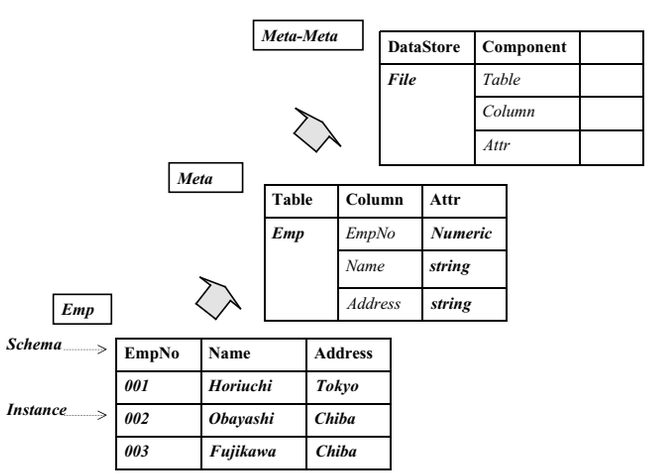

図 -1　IRDS におけるレベル対

層	役割	例
メタメタモデル（M3層）	メタモデルの定義	Metaclass MetaAttribute
メタモデル（M2層）	・メタメタモデルのインスタンス ・モデルの定義	Class Attribute
モデル（M1層）	情報ドメインを記述するための言語	社員 製品
オブジェクト（M0層）	・モデルのインスタンス ・情報ドメインの例	田中 TV

図 -2　MOF におけるメタ階層

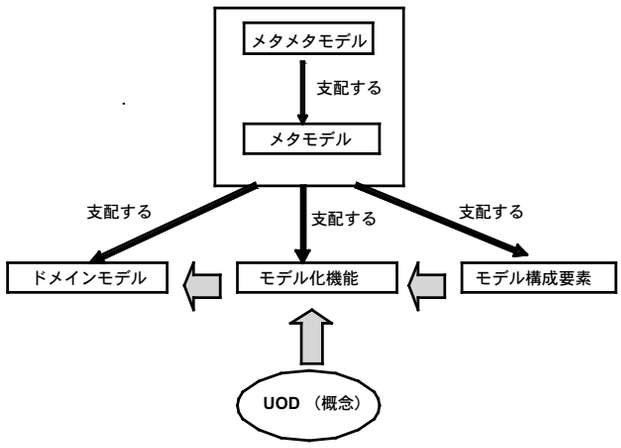

図 -3　モデルとメタモデル

◆メタモデルの位置づけ

メタモデル議論を整理するために，モデルとメタモデル，それに付帯する概念の相関を見ておこう（**図 -3**）．

ドメインモデル（Domain Model）

一般に，特定な概念（対象領域，UOD：Universe of Discourse）の本質を，定められた規則と記号により，規範的なモデル構成要素を用いて表現したものと定義される．UML1.3[2] の定義では，「特定な目的を持った物理システムの抽象」としている．ここでは，「UML モデル」，「IDEF1X モデル」などというモデル概念と峻別するために，実世界を対象としたモデルを「ドメインモデル」と呼ぶことにする．

モデル化機能（Modeling Facility）

UML あるいは IDEF1X など，モデルを作成するための規則や記法のセットを指す．UML はモデル化言語（Modeling Language）と呼ばれるが，他のモデル化の基準を考慮するとモデル化機能と呼んだほうがよい（ISO 概念スキーマ標準化プロジェクトで使われた言葉）．本稿では，この用語を使用する．

モデル構成要素（Modeling Constructs）

モデルを構成する要素．言語における単語，熟語に該当する．具体的には，標準識別･分類，標準データ要素，標準実体プロファイル，あるいはオブジェクトパターンなどを指す．メタモデルの一部ではなく，メタモデルの統治対象である．後述する UML メタモデルにおける「モデル要素（model element）」とは異なる．

メタモデル（Metamodel）

モデルを記述するモデル．モデルを構成する概念，規則を示すもの．MOF では「モデルを記述するための言語を定義するモデル」としている．つまり，モデルのセマンティクスとシンタックスを定義するモデル（ここでのシンタックスは，具体的な構文ではなく，抽象的な構文を指すことから，抽象構文と呼ぶ）．

その統治範囲（scope）は，図 -3 のように，モデル化機能と生成されたドメインモデル，さらにモデル構成要素である．それぞれの役割は次のようなものである．

- モデル化機能のためのメタモデル：
  モデル化機能が持つ概念や規則を，モデルセマンティクスとシンタックスとして記述したモデル．

- モデル構成要素のためのメタモデル：
  雛形やオブジェクトパターン，あるいは情報要素（メタデータ）などドメインモデルを構成するモデル構成要素のそれぞれを表現するモデル．その所在，型，制約を記述する．

- ドメインモデルのためのメタモデル：
  特定なモデル化機能に則って作成されたドメインモデルを表現するモデル．ドメイン固有のどのような雛形に沿ったものか，どのようなモデル構成要素を使ったものか，あるいはそのドメインモデルの管理属性などを記述したモデル．

メタメタモデル（Meta-meta Model）

メタモデルそのもののセマンティクスとシンタックスを記述するモデルであり，メタモデルを構成する要素間の関係やその規則を記述する．さらに，メタモデル相互やメタデータとの関係なども記述される．

メタメタモデルにより，メタモデルの特性を理解できる．メタメタモデルを標準化することで，メタモデルの定義，記述方法，読み方，解釈の標準化が可能となる．

◆メタモデルの基本例

ここで，一般的なオブジェクト指向における基本的なメタモデル概念の例を示そう．

オブジェクト指向では，実現値としてのオブジェクト

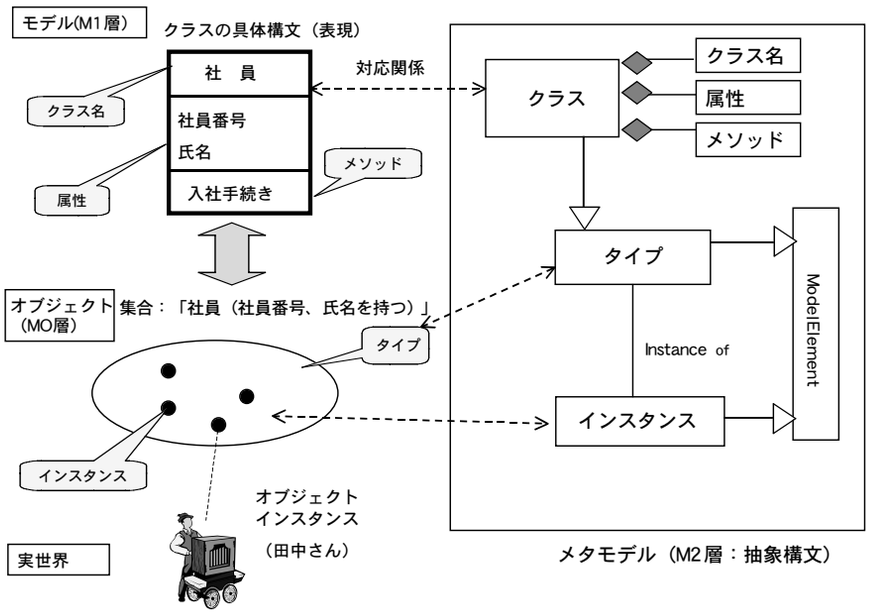

図 -4　オブジェクトの構造とメタモデルの例

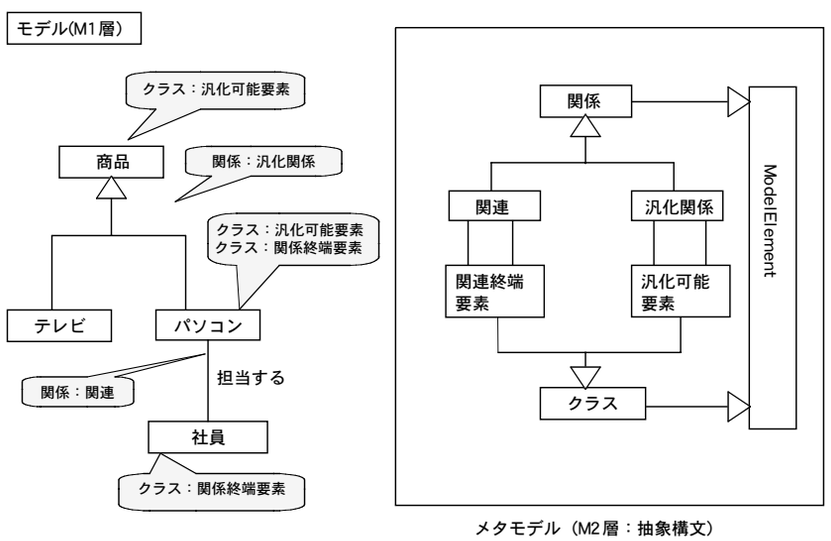

図 -5　関連を記述するメタモデルの例

はいずれかの「クラス」に属する．その「クラス」は，複数個の「属性」，「操作」および「クラス名」を持つことができる．そのような構造はメタモデルで規定され，オブジェクトモデル上のクラス「社員」は，メタモデル上クラス（メタクラスと呼ぶ）の具象構文（ここでは，UML のクラスの記法）を用いて表現される．（**図 -4**）．

さらに，複数の「クラス」の「関係」や，「クラス」間の「継承関係」を設定することができる．それらを規定するのもメタモデルである（**図 -5**）．

これらのメタモデルは，その上位のメタメタモデルによって，たとえば，**図 -6** のように記述される．

メタモデルの意義

メタモデルに何を期待するのか，その意義と役割は次のようなものとなる．

◆明快なモデル定義基盤の確立

メタモデルの最も根幹的機能である．あるモデル化機能（UML，IDEF1X など）を用いてモデルを開発する者へ，矛盾のないモデル化基準をメタモデルとして示すことにあり，その標準化はモデル開発やソフトウェア開発に次のようなメリットをもたらす．

- モデル化機能の理解促進と共通理解
- モデル化観点の共有とモデル再利用
- モデルの一意性確保と自動化の促進
- システムライフサイクル可視化と管理の容易化

◆共通モデル構成要素への準拠促進

ビジネスにおける識別コードや分類体系，あるいは「日付」，「金額」といった情報要素，さらに，「企業」，「顧客」といった実体プロファイルなどのモデル構成要素に関するメタデータの所在や意味，形式をメタモデルとして定義し，その利用と準拠の促進を図ること．企業間連携の基盤として標準の重要性が再認識されている．それらの標準として，MDR（Meta Data Registry: ISO/IEC 11179）[3] や ebXML のコアコンポーネント（UN/CEFACT）[4] などがある．

オブジェクト関連の雛形であるオブジェクトパターンも共通モデル構成要素として再認識され，UML メタモデルでの記述に関する議論が OMG でなされている．

◆モデル拡張性のメカニズム

モデルあるいはモデル構成要素の再利用における課題は個別事情による特殊化の容易性にある．メタモデルを参照しながら，継承したモデルに個別要素を付加して特化させるメカニズムをメタモデルの機能とすること．精神論や手順，規則による再利用でなく，自ずと再利用せざるを得ないメカニズムとしてメタモデルとそれに基づ

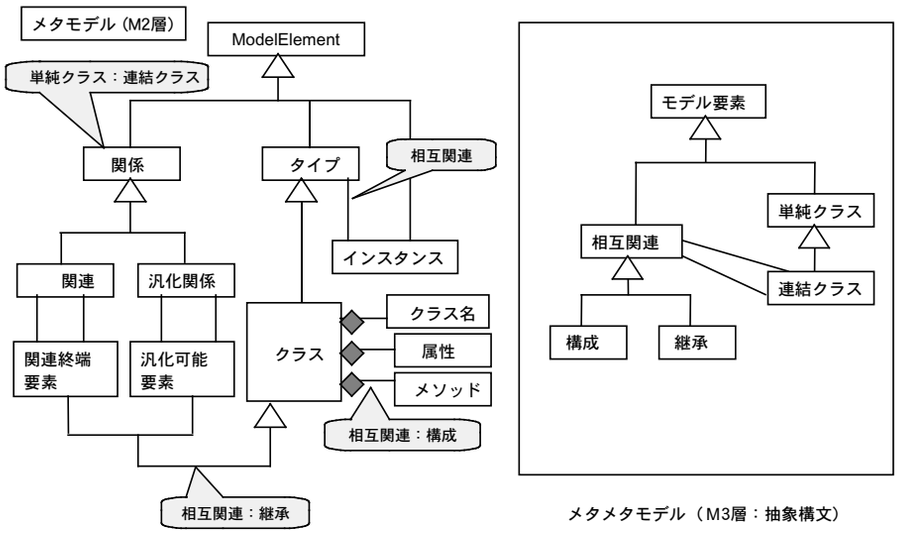

図 -6　メタメタモデルの例

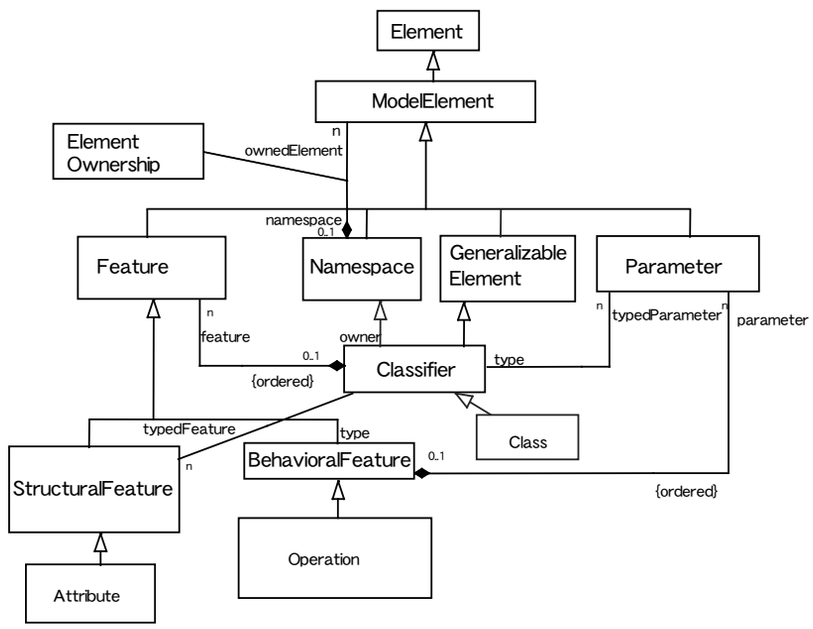

図 -7　UML メタモデルコア部分

く開発ツールに対する期待は大きい．

◆モデル多相性のメカニズム

一度作成したモデルやオブジェクトを，局面に応じて変化させ，状況に対応させること．あるいは別なオブジェクトに変態させること．メタモデルによりポリモアフィズム，あるいはイントロスペクションなどと呼ばれる機能の実現が容易となる．メタモデル標準化はそれらプログラミングシステムの改善やモデルベース開発ツールの進化に貢献する．

◆モデルの統合と変換

異なるモデルまたはオブジェクトを対応付けて，統合的に扱うこと．異なる概念から構成されるモデルやオブジェクト相互の対応・変換，あるいはモデル間の関連をメタモデルで記述すること．特に，データウェアハウスやデータマイニングでは，異部門や異企業のデータを統合して扱うことを求められ，そのために，それぞれが持つメタデータ間の変換や統合の規則や形式の記述をメタモデルに期待する．そのようなメタモデル標準には，OMG の CWM[5] などがある．

◆モデルの並行制御

モデル定義・作成段階ではなく，モデルの実行制御を，メタモデルを基に行うこと．エージェントなど複数オブジェクト（マルチオブジェクト）の並行制御情報としてメタモデルを捉えるもの．

◆動的モデル化と柔軟性

同じく，モデルやオブジェクトの実行時に，メタモデルを参照しながら，時にはメタモデルそのものを書き換えることで動的な変態（インターセッション，リフレクションなどと呼ばれることもある）を実現させること．

UML におけるメタモデル

ここで，メタモデルの具体例として，UML メタモデルを見てみよう．

◆ UML メタモデル

UML では，メタモデルによってシンタクス（abstract syntax）やセマンティクスを規定している．特にシンタクスは UML のクラス図（のサブセット）を用いて表現されている（後述するように，このシンタクスは MOF を用いても記述されている）．**図 -7** は，UML メタモデルの一部である．

上記メタモデルが，UML 記述能力をどのように規定しているかを，"Window" クラスを例として説明する（**図 -8**）．

UML クラスは，たとえば以下のような記述能力を持つ．

(a) クラスには名前がある（この例では Window）．

(b) クラスは構成要素（属性，操作）を持つことができる（size や maxSize）．

(c) 属性，操作は，可視性（public'+', protected'#', private'-'），スコープ（class 下線付き /instance 下線なし）を指定できる．

(d) 属性には型が指定できる．（Area, Rectangle など）．

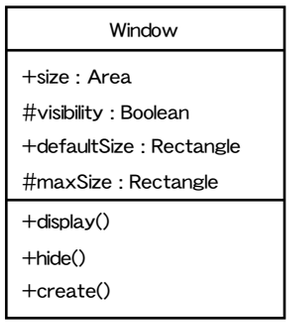

図 -8 "Window" クラスの例

これらの性質はメタモデル上ですべて表現される．たとえば，クラスを表現する Class は，ModelElement を先祖（上位クラス）に持ち，ModelElement は属性 name を持つため，Class は名前を持つことができる．また，Class（上位クラスは，Classifier）は Attribute や Operation を構成要素 Feature として持つことができる．さらに Feature は visibility や ownerScope を通して可視性やスコープの性質を備えることができる．また，Attribute（上位クラスは，Feature）は，Class（上位クラスは，Classifier）と役割 type の関連があり，属性（Attribute）は型を持つことになる．

メタモデルの構成要素は，接頭語に meta- を付けた用語で呼ばれることが多い．たとえば，meta-class, meta-attribute, meta-association などである．先の例では，Class, Attribute, Feature, Operation など，メタモデル上でクラスとして示されているものが meta-class であり，それらの属性（name, visibility など）は meta-attribute である．Attribute が型（type）として Classifier と結ばれていることを指定した関連は，meta-association である．

このように，メタモデルは UML などのモデル化機能のシンタクスを規定する．さらに，モデル化機能の利用側からみると，モデル化機能のメタモデル図を一瞥することで，そのモデル化機能の記述能力の範囲，記述ルールについての概要が理解できる利点がある．

◆ UML における拡張メカニズム

UML は汎用的なモデル化言語であり，対象領域や実行環境などから独立である．しかし，特定の対象領域（たとえば，特定の業務システムや応答制約の厳しいリアルタイムシステムなど）や特定の言語（Java/C++ など），あるいは特定アーキテクチャ（RDB，EJB など）を対象とするモデル化では，対象領域に特化した概念をモデル作成時に追加できたほうが便利な場合がある．

たとえば，企業システムを構築する場合，オブジェクトを，RDB の表として実装されるようなデータ資源を表現するもの，制御プログラムを表現するもの，などの種類を設け，モデル作成時に，たとえば前者を "Entity 型クラス "，後者を "Control 型クラス " と明確化しておくと，個々のクラスの性質がより明らかになる．この場合，元の UML の概念である汎用的なクラス概念に，新たな概念（"Entity 型クラス " など）を追加することになる．

また，ターゲットが RDB である場合，表で実装されると予想されるクラスの属性にキー項目かどうかの指定をすることができれば便利である．この場合，Attribute に新たな属性（キー項目かどうか）を追加していることになる．また，このような詳細化は，モデルからソースコードを自動生成するようなツールを実装する場合にも活用できる．

このように UML を特定分野あるいは用途向けに「カスタマイズ」する方法として，UML 自体が「拡張メカニズム」を備えている．拡張メカニズムにより，元来，UML に備わっているメタモデル構成要素に，新たなメタモデル構成要素や新たな属性を追加できる．

UML の拡張メカニズムは**図 -9** のとおりである．

図 -10 に，ステレオタイプを用いて先の例 "Entity 型クラス " をメタモデル上に追加した様子を示す．

拡張メカニズム	用途
ステレオタイプ	UMLメタモデル要素をベースとして，メタモデルに新たなモデル要素を追加する
タグ付き値	UMLメタモデル要素に，新たな属性と値を付加する
制約	UMLメタモデル要素に，新たな制約を付加する

図 -9 UML の拡張メカニズム

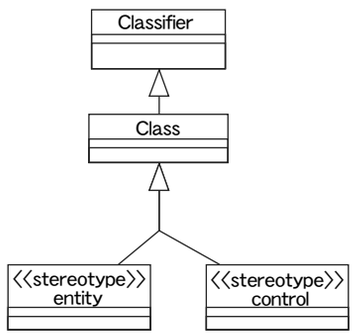

図 -10 ステレオタイプの定義例

UML profile

拡張メカニズムを用いてカスタマイズされた UML を UML profile と呼ぶ．UML profile は，UML 仕様内に例示されている UML profile for Software Development Processes や UML profile for Business Modeling 以外にも，現在，OMG やその他の団体でさまざまな UML profile が検討されている（**図 -11**）．

仕様名	対象領域／対象プラットホーム
UML profile for EDOC	企業システム
UML profile for EAI	EAI（企業システム統合）
UML Profile for Schedulabilty, Performance and Time	リアルタイムシステム
UML profile for EJB	EJBプラットフォーム

図 -11　UML Profile の例

メタモデル機能：MOF

UML profile は，UML メタモデル（メタモデル要素）をベースにして新たなメタモデルを作成する方法である．これに対し，MOF[6] は，より一般のメタモデルを記述する枠組み，すなわちメタモデルを対象としたモデル化機能であり，UML メタモデルのメタモデル要素（meta-class, meta-attribute, meta-association）などを定義している．

これらのメタモデル要素により，ユーザはさまざまな分野のメタモデルを自由に作成することができる．

OMG では MOF とペアで XMI（XML metadata interchange）仕様を提供している．XMI は，MOF で定義されたメタモデルと，そのメタモデルによって記述されたモデルを XML 形式で交換するための仕様である．

MOF によるメタモデル定義例

UML メタモデルは MOF の中でも定義されており，そのため MOF/XMI 機構により異なるツール間でのモデル情報交換が容易となっている．

前述した CWM は，データウェアハウス（DWH）構築におけるさまざまなデータ資源（リレーショナル，レコード，多次元，XML など）のメタモデル，DWH 分析ツールをモデル化するための変換やデータマイニング，あるいは OLAP のためのメタモデルなど，幅広い技術分野のメタモデル仕様を提供している．

このため，CWM メタモデル記述では，一部 UML メタモデルを流用してはいるものの，それに納まりきらず，直接 MOF 機構を利用してメタモデル定義をしている．MOF によるメタモデル定義機構については，後編で詳しく解説する予定である．

メタモデル標準化の経緯

メタモデル標準化の歴史は，1970 年代中頃の ANSI/X3/SPARC の「概念スキーマ」に遡ることができる．「概念スキーマ」は，実世界概念を表現するだけでなく実装と利用との橋渡しをするものであることから，メタモデル機能を持つものであったと考えられる．

ISO/TC97 による「概念スキーマに関するテクニカルレポート：ISO/TR-9007，1987 年」[7] でも，異なるデータモデル（網目型, 階層型, 関係型など）の変換を「概念スキーマ」の役割の 1 つとしていた．

1994 年に ISO/IEC JTC1　SC21 でスタートした「概念スキーマモデル化機能：CSMF」プロジェクトも，EDI におけるデータ交換，CASE ツール統合，異データモデル化機能間の変換，などの基盤として「概念スキーマ」標準化を目指したが，プロジェクトは中止された．

当時は，メタモデルよりも概念スキーマによって問題解決を試みていたといえる．

ソフトウェア仕様のメタモデル標準化を明示的に試みたのは，1990 年代初頭，CASE ツール間のデータ交換標準化を目的として開始された CDIF（CASE Data Interchange Format）プロジェクト（ISO/IEC JTC1 SC7WG11）であろう．　データモデル，データフローダイアグラム，状態遷移図などに関するメタモデルを，「メタ実体」と「メタ関係」，「メタ属性」で構成される汎型と規則で示している．「CDIF Framework（ISO/IEC 15474）」，「CDIF Transfer Format（ISO/IEC 15475）」，「CDIF Semantic Metamodel （ISO/IEC 15476）」などの規格化が進められている．

今日のメタモデル議論の多くは，UML と MOF をベースとして行われている．UML は目下，国際規格化が進められており（JIS 化もそれを睨んで準備待機中）．MOF も ISO への提案準備が進められている．

一方，ISO の情報技術分野（ISO/IEC JTC1）以外でも，TC184（製造業），TC127（土木建築），TC154（EDI），また，ISO 以外の UN/CEFACT，あるいは業界団体の OASIS，EAN などで，メタモデル標準化が議論されている．いずれも，製造装置間連携，土木機械間連携，あるいは EC（電子商取引）による企業間連携などが急務となり，各種ドメイン固有モデルの共有が必須となっているからといえる．

メタモデル標準化議論は，昨今の情報技術議論のほぼ全面に及ぶといってよい．「グローバル連携」への対応が求められているからであろう．一方で，ソケットインタフェースなど物理的規格化から始まった標準化の主眼が，長期的トレンドの中で抽象化された領域に到達したという見方もある．

今後のメタモデル標準化動向

◆ OMG における議論

MDA

OMG では，前述の UML，MOF，CWM を中核技術として，MDA（Model　Driven Architecture）の確立を目指し，関連仕様の整備に注力している．特に，MDA では，プラットホーム（実行環境）非依存モデル（PIM: Platform

Independent Model) から，プラットホーム依存モデル (PSM: Platform Specific Model) を，モデル変換を通じて，自動的，あるいは半自動的に実現しようとするものである．これらは，多様化するアプリケーション開発，特に，業務の変化やシステム環境の変化等に，柔軟に追従できる仕組み，すなわち，過去の資産や投資を有効に生かしながら情報システムの発展を促す仕組みの必要性から生まれたものである．

MDA の狙いを実現するには，PIM や PSM が精密にモデル定義されていなければならない．さらに，各モデルの意味を厳密に解釈できるメタモデルが共通化，標準化されている必要がある．そのような意味からも，ますます，メタモデル関連技術の重要性が高まっている．

UML2.0 と MOF2.0 における議論

OMG では，現在，UML1.4 の改定作業と同時に，MDA を支えるモデル化の枠組みを目指して UML2.0 および MOF2.0 への改定作業が進められている．

UML 改定のねらいは，UML1.4 までの仕様策定の中で議論されていた，たとえば，UML 言語定義としてのあいまい性，言語としての系統的な拡張性，MOF との整合性，モデル情報相互運用性などを解決することである．さらに，MDA 指針に従った，メタモデルやモデルの定義，管理，変換などの方式を構築するための言語基盤を確立し，仕様化することである．

◆メタモデル言語の動向

OMG の UML2.0 および MOF2.0 の仕様開発で注目されるのは，Tony Clark，Andy Evens，Stuart Kent らの pUML group の提案である [8]．UML の 4 階層のメタ階層概念を基礎に，より厳密な UML を目的に，メタモデルを記述するための言語 MML（Meta-Modeling Language）と，それを用いた UML 仕様の枠組みを提案している．UML1.4 までで使われていたメタモデル記法では，メタモデルとそれに従って描かれたモデルとの対応を厳密に意味付けできていない．彼らのアプローチでは，MML を正確に定義し，モデルが，MML で表現された，どのメタモデルのインスタンスであるかをツールでチェックできるような厳密さを要請している．

つまり，MML では，モデル側とインスタンス側を明確に識別し，それぞれが“概念”（クラス，属性など）と“構文”（矩形，矢印など）を持つ．同じ概念は，共通の構文を持つことができ，一般に，構文は多くの概念で共有化される．一方，“意味”は，モデル (部分) の“概念”とインスタンス（部分）の“概念”間のマッピングとして定義される．このような枠組みを UML 仕様定義の再構築に適用することにより，UML profile から派生する多様なモデル構成要素を統一的に UML 言語族として定義，管理することが可能になる．

◆メタモデルフレームワーク標準化

本稿で概観したように，メタモデル技術の意義と重要性は広く認識されつつあり，関連技術の整備と啓蒙が急務となっている．

たとえば，UML とそのメタモデルを用いたビジネスオブジェクト標準化が，OMG，ISO，UN/CEFACT などの標準化団体で活発に行われているが，必ずしも共通の方法，基盤の上で行われているわけではない．

それらの成果を，相互に連携させ，有効に活用，普及させるためにも，メタモデルに関する共通のフレームワークを定めることが重要となってきた．

このような動向を踏まえ，日本からメタモデルの相互運用のためのフレームワーク標準化（Framework for metamodel interoperability）を ISO/IEC/JTC1　SC32 に提案し，新規プロジェクトとして，2002 年の 5 月ソウル会議で正式に発足した．

メタモデルはモデル開発やソフトウェア開発自動化だけのものではない．ネットワークを通じて動的に連携する企業システムは，連携の意味的整合性を保障する基盤を必要とする．その基盤は，オントロジー，メタデータ，ドメインモデル，ビジネスプロセス，あるいはそれらを規定する規格などに関するメタモデルの交換を可能とするものとなる．

メタモデル体系化に関する議論は緒に就いたばかりである．

参考文献

1）堀内　一：情報資源辞書システム IRDS の標準化，情報処理，Vol.37, No.7（July 1996）．
2）OMG Japan-SIG, 翻訳委員会編：UML 仕様書，ASCII（2001）．
3）ISO/IEC JTC1 11179-3: Information Technology: Meta Data Registry-Registry Metamodel．
4）ebXML Core Component: http://www.ebxml.org/
5）Poole, J. et al.: Common Warehouse Metamodel, OMG Press（2001）．
6）OMG: Meta Object Facility（MOF）Specification Version1.3（2000）．
7）ISO/TR9007: Information Processing Systems-Concepts and Terminology for Conceptual Schema and the Information Base（1987）．
8）Clark, T. et al.: A Feasibility Study in　Rearchitecting UML as a Family of Languages Using a Precise OO Meta-Modeling Approach（Sep. 2000）-available from　http://www.puml.org/

（平成 14 年 9 月 30 日受付）