

ソフトウェア環境におけるリポジトリへの要求 -- リポジトリ内容標準化の要件--

堀内 一
東京国際大学

リポジトリに関する国際規格は、1993年3月にIRDSとして、また、1994年3月にPCTEとして相次いで制定された。今後の標準化の主旨はリポジトリに登録すべきメタ情報（リポジトリコンテンツ）に移行することが想定される。しかしながら、リポジトリ内容の検討はソフトウェア開発プロセスからの論議が多く、クライアント/サーバシステムなど今後の情報システム形態を想定すると、ソフトウェアツールの稼働時（実行時）をもサポートするメタ情報の標準化が重要となる。

本稿では、伝統的なデータディクショナリの内容を考察しながら、今後のリポジトリ内容とその標準化への要求を述べた。

The Requirements for Repository in the Software Environment -- Repository Content Standardization Requirements --

Hajime HORIUCHI

Tokyo International University

Two ISO repository standards were established, one was the IRDS Services Interface in March 1993, and the other was the PCTE in March 1994. The focus of the repository standardization is expected to move the topics of repository contents. However, the most of discussion have been made from the view point of software development process for long time.

Supposing current trends of the information system architecture, especially, Client / Server architecture, the contents of repositories should be changed from the traditional concept. Not only for the software development, the execution of software should be supported by the meta-data in the repository.

This paper deal with the requirements for the repository content standardization, comparing with traditional data dictionary contents, and using the object orientation and the atomic object concept.

1 はじめに

リポジトリ (repository) とは、ものごとと貯蔵庫を指す言葉である。ソフトウェア開発やCASEツールの分野では「情報資源貯蔵庫」と呼ぶこともある。この用語が一般化したのは、1989年に発表されたIBM社のAD/CYCLEで用いられてからである。そもそも、その貯蔵庫が何を格納すべきか定説はない。データやソフトウェアに関するメタ情報だけを格納すべきとする意見、あるいは、直接ソフトウェアが操作するデータのインスタンスまでを含むべきであるとする意見もある。

リポジトリに関する標準化作業は1983年からISOにおいて進められ、1993年3月にIRDS (Information Resource Dictionary System) サービスインタフェース (IS10728) として規格が成立している。また、1994年には、ECMAが開発したPCTE (Portable Common tools Environment) もISOの規格として制定されている。それぞれ情報資源のメタ情報の格納方法、あるいは開発を前提としたソフトウェアのメタ情報の格納方法にその標準化の主旨をおいている。

さらに、ISO/IEC JTC1/SC7/WG11では、EIAのCDIF (CASE Data Interchange Format) プロジェクトと共同で、ソフトウェアエンジニアリングを支援する各種ドキュメントのメタ情報の標準化を進めている。

一方、コンピュータシステムアーキテクチャとソフトウェア構成概念は、リポジトリがデータディクショナリとして登場した頃に比して大きく変化している。しかし、リポジトリの用途や内容 (コンテンツ) に関する論議の多くは、依然として旧来のアーキテクチャとソフトウェア構成概念を前提にしているように見える。

つまり、パソコン、ワークステーションとネットワークによるシステムのオープン化は、ソフトウェアのコモディティ化を促進させ、手作りによるソフトウェアとその生産形態を急速に陳腐化させつつある。

さらに、実質標準として君臨するメジャーなソフトウェアコモディティの活用によるインターネットなどの自由な接続は、最新の情報にアクセスしなければならない、という「情報ギャップ強迫」を創造し、それがまた、コモディティソフトウェアの早期導入と迅速なインストールを希求するという環境を造り上げている。手作りソフトウェアなど想像もできない環境になりつつあるともいえる。

本稿では、何らかの対象物に関するメタ情報の集合をリポジトリと捉えながら、上記のような情報システム環境想定しながら、そこにおけるリポジトリの意義とそのためのメタ情報構造に関する要求を述べる。

2 リポジトリ概念の変遷

リポジトリのメタ情報管理としての概念は、古くからデータディクショナリあるいはDD/DS (Data Dictionary and Directory System) と呼ばれてきたものと変わりはない。しかし、その適用の目的あるいは概念は、その時代の要請にしたがって変化してきた。図1は、その変遷を示すものである。

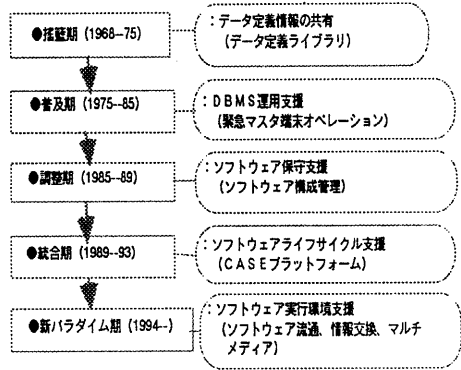


図1 リポジトリ適用概念の変遷

(1) 搖籃期 (1960末～)

プログラムを作成する目的、あるいはデータを管理する目的のために、データ形式などを記述したデータ定義を、コンピュータファイルに格納管理する概念として、データディクショナリなる概念が誕生した。その原型は、1960年代末の磁気テープベースのツールに遡ることができる。

(2) 普及期 (1975～)

データディクショナリが、一般にその意義を認められ、商用のパッケージとして登場したのは1970年代の中頃からである。当時、普及期にあったDBMSを支援するものとして、トランザクション、プログラム、及びデータの相互関連を蓄積し、主としてDBMS運用時に管理者端末オペレーションを支援するツールとして位置を占めた。

(3) 調整期 (1985～)

DBMS運用の支援、あるいはデータベース設計支援として定着したデータディクショナリは、ソフトウェア開発における変更制御ツールとも見なされようになり、プロジェクト管理ツールやソフトウェア開発支援ツールとの連動が試みられた。また、プログラム生成などにデータや設計図面のメタ情報が必要とされ、ソフトウェア開発におけるデータディクショナリの重要性が認識された。

データディクショナリツールのインタフェースに関する標準化が議論されるようになった。

(4) 統合期 (1989～)

ソフトウェア開発の各フェーズを統合する手段 (クロスライフサイクル支援)、あるいはCASEツール (ツール統合) そのものを

統合する手段として、その重要性が認識された。またソフトウェアエンジニアリングで必要とするドキュメントのメタ情報に関する標準化が議論されるようになった。

(5) 新パラダイム期 (1993~)

クライアント/サーバシステムやオブジェクト指向の普及、さらにインターネットなどグローバルネットワークの普及などにより、ソフトウェア環境やソフトウェア開発に対する概念的認識も大きく変わりつつある。

オブジェクト指向のクラスライブラリ概念などと統合した、新しい理念が求められている。また、1993年はIRDSの国際規格が制定された年でもある。

3 メタ情報とリポジトリ内容

3.1 メタ情報とは

メタ (meta)とは、「超」あるいは「一段高レベル」のような意味である。メタ情報とは、一段上において「対象を記述した情報」をさす。「記述される対象」と「記述するもの」の関係は無限に繰り返すことができる。そのような関係をメタ階層とよぶ。リポジトリはそのようなメタ情報のファイルである。ただし、メタ情報として対象のいかなる属性を記述するかは特定できない。

図2は、ISOのIRDSサービスインタフェース規格 (IS10728) におけるメタ情報の概念を示すものである。

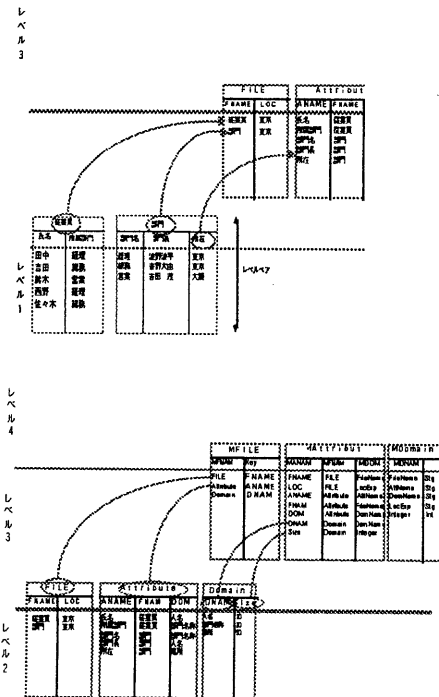


図2 ISO/IRDSにおけるメタ階層

3.2 これまでのリポジトリ内容

1980年代初頭にデータディクショナリとして登場したリポジトリは、その理論的な考察や枠組みを持たないまま、実務的な要請に応えながら広く浸透した。

当時のデータディクショナリの内容 (メタエンティティと呼んだ) は、概ね以下のようなものであった。

(1) ソフトウェア開発用メタ情報

1980年代の中頃、成長期にあった情報システムの開発で、大量の人員を投入して行われたソフトウェア開発のために、データ項目 (Item) の管理を中心に、それをアクセスする、レコード、ファイル、プログラム、サブシステム、システムなどの関連 (ソフトウェア構成) を管理するメタ情報が蓄積された (図3 参照)。

したがって、データ項目も「名称」「形式」に加えて、「所有者」、「有効期限」などのメタ情報をもつもので、その主旨はプロジェクト内で頻発する仕様変更の影響度分析などプロジェクト管理面の要請が色濃く反映された (図4 参照)。

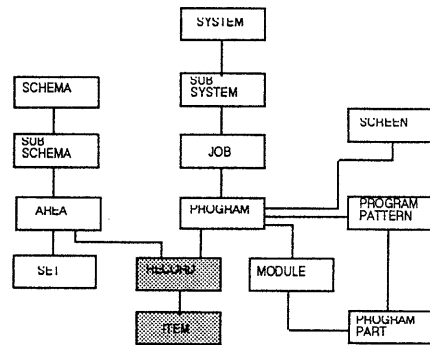


図3 ソフトウェア開発用メタ情報

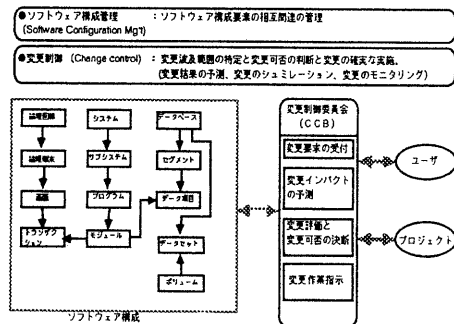


図4 変更制御用メタ情報

(2) システム運用管理用メタ情報

ソフトウェア開発メタ情報に加えて、ソフトウェアの運用を管理するためのメタ情報として、ジョブ構成、所要リソース（ファイルボリュームなど）との関連を蓄積したものの。

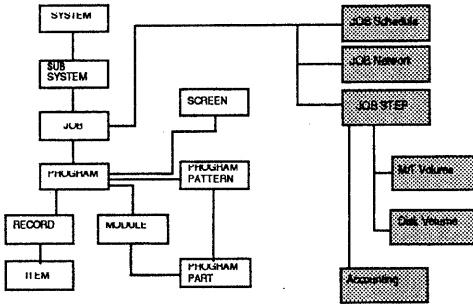


図5 システム運用管理メタ情報

(3) ソフトウェア工学用メタ情報

1980年代の後半から、それまでソフトウェア開発支援ツールと呼ばれたものがCASEツールと呼び換えられ、開発プロセスの上流から下流に至る様々なツールが提案され商品化された。そのようなツール乱立は、当然ながら異なるベンダツール相互の有機的結合を目的とした標準化を喚起した。

現在、ISO/IEC SC 7/WG 11がEIA/CDIFプロジェクトと共同で進めているソフトウェアエンジニアリングのためのメタ情報標準化では、ツール統合のためのデータ統合(Data Integration)の手段の一つとして、データフローダイアグラム、状態遷移図、あるいはデータモデル図など設計情報のリポジトリへの格納形式を検討している。

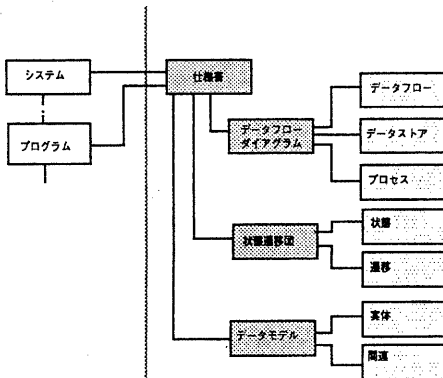


図6 ソフトウェアエンジニアリング用メタ情報

(4) システムモデリング用メタ情報

CASEツールやプログラム開発環境の整備によって、開発プロセスの下流工程（プログラム生成）の効率は大きく改善され、1990

年代に入ると、問題意識は上流工程の改善に向けられるようになった。その背景には、ビジネスとソフトウェア開発の整合性確保におかれ、特に、昨今話題のBPR (Business Process Reengineering) などでは、ビジネスプロセスの観点から、情報システムプロセスやデータモデルを眺め、業務プロセス改善に資する手段を求めている。

そのため業務プロセスや実体、あるいは業務ルールなどに関するメタ情報をソフトウェア資源と結びつけてリポジトリで管理する概念が登場してきた。

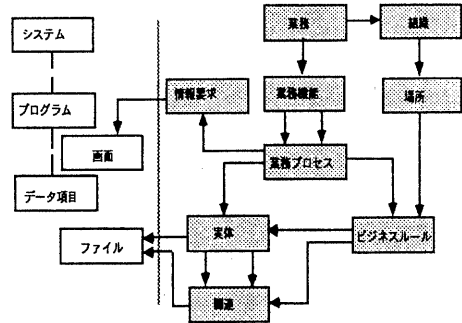


図7 システムモデリング用メタ情報

4 リポジトリ内容に関する2つの立場

リポジトリのインターフェース標準化を進めるISO/IEC SC 21 (IRDS)、SC 22 (PCTE)も第一段階の規格化を完了し、つぎのテーマとして採り上げているものがリポジトリ内容 (Repository Contents)の標準化である。

リポジトリに登録すべき内容の標準化という点では、SC 7 (SEDDI)、SC 14 (データエレメント)、あるいはTC 184 (EDFACT)などが進めている標準化作業も結果的に同様の主旨のものといえる。標準化団体をISOに限らず広く捉えるならば、OMGやX/OPEN、あるいはMicrosoft社とTI社の提携などで行われつつあることも、そのスコープ内に入る。しかしながら、リポジトリの概念は単純で、融通無碍なものだけに、同床異夢となりやすいリスクをもつ。

今日、オブジェクト指向分野でも、プログラム言語としてのオブジェクト指向と、データベースとしてのオブジェクト指向に若干の齟齬がみられ、必ずしも統一された概念となっていない。同様に、リポジトリ内容標準についても、大きく2つの立場が存在する。

その一つは、情報システムあるいはソフトウェアを開発作業のフローの観点から捉えるものである。もう一つは、開発作業の結果、ストックとして存在するソフトウェアあるいはデータの観点から捉えるものである。

両者は全く異なるメタ情報を要求する。

4.1 フロー重視の立場

ソフトウェアを設計、開発する人的作業、あるいは作業工程からメタ情報を決定する立場を指す。

システムライフサイクルと称して、特定システムの開発過程をフェーズ分けして、それぞれのフェーズの間を繋ぐ手段としてリポジトリを置き、作業情報としてのダイアグラムと、それに関するメタ情報の共有を重視している。そのような概念はクロスライフサイクル(Cross Lifecycle) サポートと呼ばれ、CASEツール統合の中核概念となった。

図8はIBM社のAD/CYCLEで示されたクロスライフサイクル概念である。また、図9はMSP社DATAMANAGERの概念として示されたものである。

IBM AD/Cycle の構想 (1989)

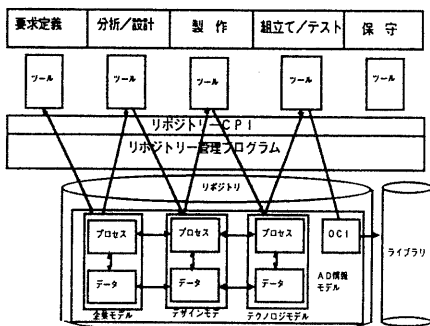


図8 IBM社のAD/CYCLEの概念

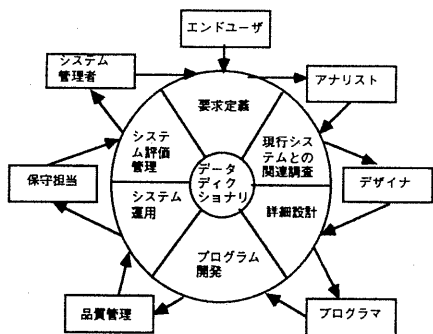


図9 MSP社DMRの概念

クロスライフサイクルサポートは設計者や開発者に、前工程での結果を引継ながら、次工程へ自らの作業結果を引き渡すことができる点で、紙などのメンテラッグを持つ図面での授受に比して大きな利益をもたらす。

しかしながら、フェーズ別のメタ情報は、開発対象となっている同一のシステムまたはソフトウェアについて、異なる立場が必要とするビュー情報を意味し、それをそれぞれのフェーズ別に格納することは重複以外のなものでもない。

同様の発想(設計情報のメタ情報重視)はSC7/WG11のDDSEIにもみられる。図10は、CDIFプロジェクトと共同で進めている構想である。リポジトリによって共用の対象となっているのはダイアグラムである。

- ISO/IEC JTC1 SC7/WG11: SEDDR (SEDDR: Software Data Definition and Representation)
- EIA/CDIF (Case Data Interchange Format)

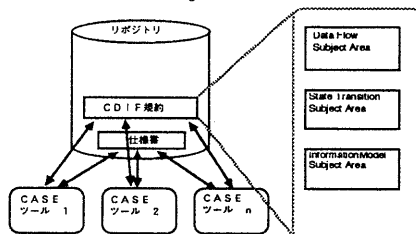


図10 SC7/CDIFの構想

もともと、ソフトウェアエンジニアリングを構成する多くの技法は、ソフトウェア生産工程のエンジニアリングに関するものである。そこには設計、開発という作業は人間の侵しがたい知的生産活動である、という暗黙の前提を置いている。

設計者や開発者というブラックボックスを支援する情報やツールがエンジニアリングの対象となり、設計基準、手順、ドキュメント、あるいはレビューポイントなどが標準化の対象となる。

4.2 ストック重視の立場

人間の知的活動として設計作業や開発作業が捉えるフロー重視の最大の問題は、開発の対象となるターゲットシステムやソフトウェアの「意味」には何等介入できないことである。最終成果物として完成させるべきものを構成する要素や、その要素を支配する原則や制約などは事前に列挙できないと考えるからである。それらは全て設計者の尊厳に触れるものとするからかもしれない。

ストック重視の立場とは、設計あるいは開発の対象とするシステム(設計結果としてのストック)そのものを構成する要素をエンジニアリングの対象とするものである。

今日、多くの企業は、すでに数メガから数十メガステップのアプリケーションソフトウェアを保有している。ビジネス領域のことごとくはコンピュータ化され、基本的なデータもほぼ網羅されている。

新システムといっても、構成要素の全てが新しいわけではない。新たに開発すべきソフトウェアも、その原子的な構成要素を調べれば、ことごとくを既に存在するシステムと共有していることが多い。

数十兆個に及ぶといわれる分子の数も、僅か数百個の原子からできている。同様に、

種々雑多、多様に見えるアプリケーションシステムもその構成要素の多くは互いに共通である。その原子的要素の数は極めて限られたものであり、多くの場合、ただ組み合わせが異なるだけである。

したがって、リポジトリに格納すべきメタ情報も、作業支援の図面のメタではなく、開発されたシステムを分解して帰納される構成要素に関するメタ情報である。

システムまたはソフトウェアを構成する要素は、それ以上分解を要しない基本的な原子要素と、それらの集約 (Aggregate) としての半加工品として要素であり、いずれもデータを核に、そのデータ固有の制約と操作をカプセル化したオブジェクトである。

図11は、既存のソフトウェアを解析し、原子オブジェクトや共通集約オブジェクトを抽出して、リポジトリに登録し、それを再利用しながら新ソフトウェアを構築するソフトウェアリエンジニアリングの概念を示すものである。リポジトリに求めるものはオブジェクトのメタ情報である。

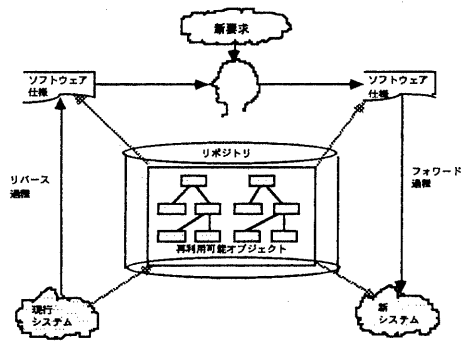


図11 オブジェクトによるソフトウェアリエンジニアリング

5 リポジトリへの要求

リポジトリの内容に関する標準化が議論され始めた今日、そのアプローチにいくつかの立場があることを述べた。そこで、ストック重視の立場から、オブジェクトに関するメタ情報を格納するリポジトリに対する要求を整理してみたい。

5.1 リポジトリがカバーすべき対象領域

リポジトリはソフトウェアエンジニアリング領域だけをサポートするものではない。情報システムに関するあらゆる領域がサポートされるべきと言っても過言でない。少なくとも次のような領域におけるリポジトリ内容の統一と共存が必須である。

- ソフトウェア開発
- プロジェクト管理
- ソフトウェア稼働

- システム管理 (分散環境管理)
- 文書交換及びマルチメディア交換
- EDI及びデータ交換

5.2 次期ソフトウェア環境と実行時リポジトリ

フロー重視のリポジトリ内容は、ソフトウェア開発という作業工程が対象となったことから、開発局面だけが考慮され、開発されたソフトウェアの実行局面は考慮されなかった。つまり、ソフトウェアを設計しプログラムを生成するまでがリポジトリの役割であった。

PCやWSなどによるダウンサイジングあるいはクライアント/サーバシステムの普及浸透を考えると、クライアント側でのプログラム開発は非現実的であり、表計算ソフトウェアなどのコモディティソフトウェアを実行させ、個々のエンドユーザの情報要求を支える環境が不可欠となる。その環境は、サーバとの連携を制御するだけでなく、多様なエンドユーザによる個別情報生産の一貫性を確保できるものでなければならない。そのためにはリポジトリに登録されたメタ情報をエンドユーザツール実行時に、必要とするオブジェクトとその制約などを引き渡すことが必要となる。つまり、リポジトリ内容のメタ情報を直接、解釈実行することが求められる。

図12は、クライアント側にリポジトリ内容を引き渡し、表計算ソフトでそのメタ情報を解釈しながら実行することで、これまでのソフトウェアエンジニアリング技法で開発されたプログラムの中に潜まれる重複したプロセスを回避できることを示すものである。

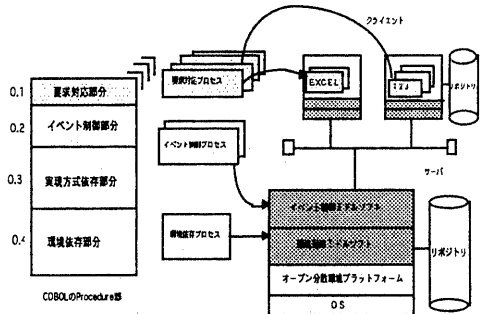


図12 実行時リポジトリ

5.3 階層リポジトリ

リポジトリの階層概念には、一つにメタ階層に関するもの、もう一つに第1階のメタレベルの実装上の階層に関するものがある。

(1) メタ階層

IRDS規格は図2に示したようなメタ階層概念をもつ。第2階のメタ (メタメタ)、第3階のメタ (メタメタメタ) をもち、リポジトリのスキーマ記述の標準化とリポジトリ概念のカスタマイズを可能としている。

特定な目的のためのリポジトリならば、第

2階は固定でよい。しかし、リポジトリの対象領域が拡大されたときは、いかなる概念のメタ情報を格納させるべきかを指定できるように第3階のメタ情報を用意すべきである。

(2) メタ情報実装上の階層化

IRDS規格は実装されるマシンを問わないが、PCTE規格はUNIXをベースとしたワークステーションを想定している。

サーバとクライアントの間、あるいはPCとメインフレームの間でリポジトリを階層化したパッケージもあるが、多くは中央側にメタ情報の全体、分散側に部分のコピーを持ちインポート/エクスポートにより整合性を維持している。

しかし、先に述べたリポジトリの対象領域を考えると、図13に示すように、全体/部分関係だけではなく、集約概念に基づく階層化が必要となる。

つまり、ローカルな特殊領域における特定のオブジェクトを、共通、普遍的なオブジェクトに基づき組み合わせて定義できるための、オブジェクトとモデル化機能 (Modeling Facility) のメタ情報が上位のリポジトリになければならない。

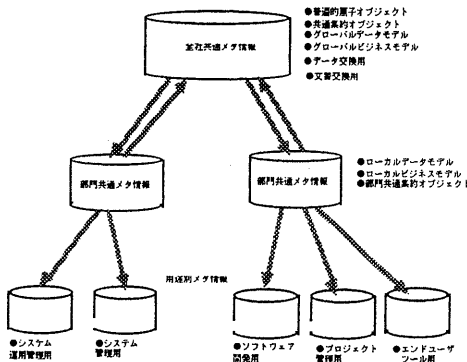


図13 階層リポジトリ

メタレベル	N + 1	N
全社共通	<ul style="list-style-type: none"> ●モデル化機能定義 ●原子オブジェクト定義 ●集約的オブジェクト定義 ●グローバルビジネスモデル定義 ●グローバルデータモデル定義 ●交換用オブジェクト定義 	<ul style="list-style-type: none"> ●モデル化要素 ●原子オブジェクト列挙 ●集約オブジェクト列挙 ●ビジネスオブジェクト列挙 ●交換用オブジェクト列挙
部門共通	<ul style="list-style-type: none"> ●モデル化機能定義 ●ローカルビジネスモデル定義 (集約分+追加分) ●ローカルデータモデル定義 (集約分+追加分) ●部門共通オブジェクト定義 (集約分+追加分) 	<ul style="list-style-type: none"> ●モデル化要素 (集約分+追加分) ●ローカルビジネスオブジェクト (集約分+追加分) ●ローカルデータオブジェクト (集約分+追加分) ●部門共通オブジェクト
用途別	<ul style="list-style-type: none"> ●実行形式及び編成定義 ●用途別オブジェクト定義 	<ul style="list-style-type: none"> ●実行形式オブジェクト

表1 リポジトリ階層におけるメタ階層

ここで、モデル化機能とは、リポジトリに登録されるオブジェクトが、どのようなモデル概念に従ったものかを定義したものである。ある領域では、E/Rモデル概念が必要にな

り、別な領域や用途では、NIAMモデルやリレーショナルモデルが必要となるかもしれない。

表1は、各階層のリポジトリのメタ情報(N+1)とそのメタ情報に基づくインスタンス(N)を示すものである。このほかにこの表の概念自身を記述するためのメタ(N+2)が存在する。

5.4 オブジェクトとしてのメタ情報

これまで、リポジトリの内容はデータの形式や名称など静的な定義情報に止まった。データの操作や制約検証はプログラムの課題と考えたからである。しかし、データは型だけでなく、それに固有な制約と固有な操作(ライフサイクル操作)を持つ。それぞれのデータ(原子及びその集約)に固有の制約と操作をカプセル化できるならば、オブジェクトとしての要件を整える。したがって、メタ情報には、それぞれの記述対象に関する次のような定義が必要となる。

- 名称
- 形式制約
- 値制約
- 存在制約
- ライフサイクル操作

また、それぞれの制約にはフォーマルな記述が望まれる。それにより直接、解釈実行が可能となる。特に、クライアント側に存在する用途別リポジトリでは、そのメタ情報としてのオブジェクトを直接、EXCELやLOTUS123などの表計算ソフトに引き渡し、表に埋め込ませることで、データ操作手続きを隠蔽させるだけでなく、値の集約などに個人差が生じない均質な情報加工が可能となる。

IRDSやPCTEでもオブジェクト指向化が検討されているが、自分自身のメカニズムをオブジェクト指向化するためのもので、オブジェクト指向モデル化機能をメタ情報として支援するものではない。

5.5 原子オブジェクトと

集約メカニズムのサポート

リポジトリの内容を標準化して共有させる上で、データやオブジェクトの概念を伝えるためのモデル化基準としてモデル化機能は必須のものであり、SC21/WG3でもCSMF (Conceptual Schema Modeling Facility) プロジェクトが作業を開始した。

しかし、モデル化機能は文法のようなもので、文章を構成するために必須ものであるが、意味を相手に伝えるためには共通の単語に変わなければならない(勿論、これもCSMFのスコープに入ると思われるが)。

情報システムの成果物としての情報(画面や帳票など様々な形態をとる)を分析すると、極めて限られた数の原子オブジェクトによって構成されていることがわかる。

そのような原子オブジェクト、例えば「日付」、「数量」、「金額」などを洗い出すと、企業の個別性はなく、極めて一般的なものである。したがって、その標準化の意義は大きく、原子オブジェクトのそれぞれに制約が定義された上で標準として公布されれば、それをベースに定義（特殊化）されるオブジェクトは、共通の性格を維持しながら利用、操作されるものとなる。

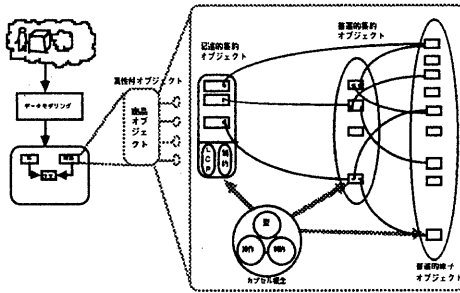


図14 原子オブジェクトによる集約概念

リポジトリ内容の標準化は、まず共通の標準化された原子オブジェクト群を、リポジトリ内容の最も低位に位置づけながら、それに基づく集約物として上位のオブジェクトを構成させるモデル化機能と、集約のメカニズムを提供しなければならない。またリポジトリのメタ情報定義自身も共通の原子オブジェクトに従わなければならない。

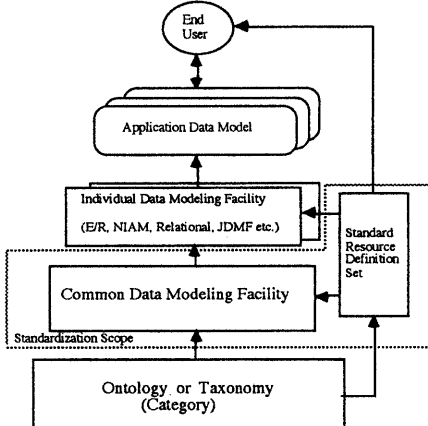


図15 DMFと原子オブジェクトの標準化

原子オブジェクトの標準化について、既に、わが国ではINSTAC（財：日本規格協会情報標準研究所）の情報資源スキーマ調査研究委員会（委員長：穂鷹良介）でもCSMF候補の一つとして提案したJDMF（JSA Data Modeling Facility）/model-1992と共に、標準化の準備を進めている。また、ISO本部でも、SC14（データエレメント）と連携して、

BSR（Basic Semantic Repository）として原子オブジェクトの標準化をすすめるプロジェクトを発足させている。

5.6 共通原子オブジェクトに基づく業界別オブジェクト標準

リポジトリ内容標準化のもう一つの課題は、共通の原子オブジェクトライブラリに基づく、業界別オブジェクトライブラリであろう。SC21/WG3では、IRDSの内容を開発する者に対するガイドラインを審議している。図16は、同ガイドラインに対する日本からの提案の一つである。

内容標準を開発するグループに対して、標準的オブジェクトライブラリと登録手続きを示している。

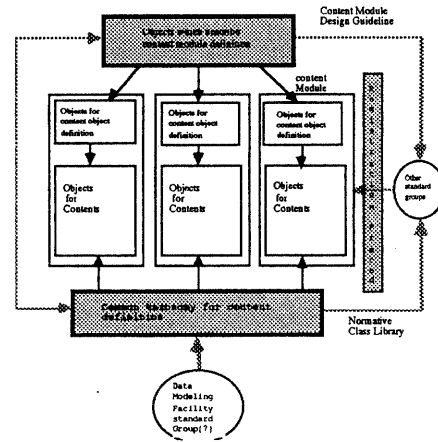


図16 IRDS内容標準化ガイドライン案

6 終わりに

リポジトリの内容標準化に対する要求を述べた。オブジェクトのメタ情報管理としてのリポジトリの基礎的な性格を考えると、DBMS、特にODBなどへの統合が検討されるものとなろう。

【参考文献】

- (1) ISO/IEC JTC1 IS10728: IRDS Services Interface, 1993.
- (2) ISO/IEC IS13719-1: Portable Common Tool Environment (PCTE): Abstract Specification: Part 1 & 2, 1993.
- (3) L. Wakeman et al., PCTE: The Standard for Open Repositories, Prentice-Hall, 1993.