

## 解説

CASE 環境統合のための  
インタフェースの標準化現状†

松本 正雄†

## 1. はじめに

これまで、ツール提供者が必要なインタフェースを独自に設定し、ツール一式を作成してきたが、次第に、標準インタフェースに準拠して作られたツールが求められるようになってきた。標準化は、技術、産業、市場の発展の当然の姿であり、CASE ツールにおいても例外ではない。以前は、市場に出まわっているツールは限られていたが、最近の種類も豊富になってきた。異なる複数ベンダ製品のなかから、より目的に合うツールを選び、それらを開発環境として統合し使用したいとする要求のほうが、ローカルインタフェース準拠でも構わないから能力的により優れたツールを使いたいとする要求を上回るようになってきたからである。

プラットフォーム上に統合した環境を構築するのに要するコストや時間は、インタフェースが標準か否かによって大きく左右されることは明白であろう。ここでは、標準化のドメインをCASE環境(CASEEと略す)に限る。CASEEは、ツールとプラットフォームよりなり、プラットフォームはハードウェア、OS、データベース、通信、ユーザインタフェースなどよりなる。

標準とは公共性があり、相当広範なコミュニティで合意形成されたものを意味し、たとえアーキテクチャとして整っているものであっても私企業所有のものは本稿では除外する。標準化を論ずる場合、大きく2点あり、一つは何を標準化すべきかの議論で、CASEE内外のいかなる側面についてのインタフェースを問題とすべきか、またその理由を明らかにすること。二つはどうか標準化を達成してゆくのが良いかという過程上の議論である。これには、標準化対象全体を支える基

礎概念を検討することや、そのなかの個別問題側面ごとに具体的な構造モデルを論ずることなどが含まれる。本稿では、主に標準化すべき側面を述べた後、CASEE統合のためのインタフェース標準化検討例のいくつかを概観してみたい。

## 2. 背景

## (1) 統合化CASEEへの要求

なぜ統合化CASEEを必要とするのか。それは、ツール間の連結を有しない開発環境、つまり統合化できていないCASEEより利点があるからであろう。統合化CASEEに対し次のような要求がある。

## データ共有

ツール間でデータを共有できれば、あるツールで開発者がデータの変更を行ったとすると、その変更が他のツールに対しても反映されるようになる。もしこの機能がないとすると、変更後の内容をしかるべき形式で再入力しなければならない。それでは手間がかかり過ぎるしデータ矛盾をきたしてしまう危険性がある。

## 開発ツールと管理ツールの統合

開発ツールと管理ツールが統合され一つの環境を成すようになれば、プロジェクトをさらに良くコントロールできるようになる。開発ツールによってどのようなアクションが取られたのか、管理ツールは通告を受け、必要な管理機能を発揮できるようになる。たとえば、開発ツールの実行終了時点で、作成された中間生産物の品質をチェックする品質管理ツールを起動する。

## 起呼機能の統合

ツールの自動起呼機能のことで、たとえば詳細設計が完了した後コード生成ツールを起呼実行することができるようになる。ツール間での自動起呼や開発プロセス全体をあらかじめ記述し、そのプロセス記述を実働化させるなど幾通りかの方式が考えられるが、人間の介入を不要化する点が肝心である。

† An Overview of Standard Frameworks and Reference Models for Integrated CASE Environments by Masao MATSUMOTO (C & C Software Development Group, NEC Corporation).

† 日本電気(株) C & C ソフトウェア開発グループ

### 開発パス追跡可能性

ツールを使って開発が進められたとすると、迅速かつ容易に開発がたどったパスを追跡することができるようになり、作業し直すときに都合が良い。ツールを使わないで人手による作業も混じるので、一般にはプロセス記述に対する実行経緯を記録し、それをもとに追跡する仕方となる。

### 整合性のあるユーザインタフェース

使い方を逐一教わらなくても、あるツールから別のツールへ移行できるようにする。

#### (2) 統合化実現の基本的側面

これまで、ツールごとに独自の図形インタフェースや処理データの内部形式をもっていた。ツール統合とは、異なる開発元からのツールであろうとも、あたかもおのおのが単一ツールの部分であるかのごとく協働することを意味する。

統合には少なくとも三つの側面がある。第一はプレゼンテーションあるいはユーザインタフェース統合と称されるもので、ユーザにとってツールが同じように見えるようにする、第二はデータ統合と称されるもので、どのツールからもデータを共有かつ操作可能なようにする、第三は制御統合と称されるもので、ツールを他のツールから呼び出せるようにする、である。

これ以外の側面もあろう。たとえば開発ツールと管理ツールの機能統合があるが、制御統合を利用して実現できるので基本的(必須)ではなからう。ツール移植性、つまり異プラットフォーム上で動いているツールを自分の環境に取り込むことは、第一から第三が満たされれば解決しそうである。ツール同士の情報受渡しとその正しい解釈は、第二と第三で解決しそうである。

#### (3) 標準案の背景

現在までに提唱されている案は、CAIS, PCTE, その拡張版である PCTE+, CIS, IRDS, シグマ, IEEE などいくつかある。構想の完成度、製品化程度、力点の置き方はまちまちである。PCTE+, IEEE は参照モデルも示している。参照モデルとは、標準化作業を手際良く進めることができるよう、どの側面がすでに標準化されているのかまたは未着手かを示すいわばガイドラインである。おのおの案はなにがしか標準案を掲げている。これら以外にもいくつかの案が提示されているが、私企業所有なので、取り上げない。

### 3. Common Apse Interface Set (CAIS)

CAIS は、Ada 関係のツールの移植性を問題視して開始された。Apse (Ada Programming Support Environment) への要求は、米国防総省 Stoneman 文書の解説<sup>1)</sup>に述べられている。1980年代初期、米国の陸軍と海軍はそれぞれ別に Apse 開発の契約を結んだ。その結果は、不幸にも ALS (Ada Language System) と AIE (Ada Integrated Environment) は非互換なものとなった。

1982年に、これを調整するためのチームが結成され、1983年に CAIS 仕様の第一版が出、ついで 1985年1月に CAIS 標準案が出された。その後、CAIS のインプリメンテーションのために幾多の努力が払われた。

しかし、CAIS-A<sup>2)</sup> にあるようなツール間データ交換などの問題を、CAIS は正しく取り上げなかった。Ada 開発環境でのインタフェース仕様の別案として PCTE があり、両者は、互いに深くかかわってきた。CAIS-A は、大規模で複雑な標準であり、Softech 社は、そのインプリメンテーションを 1989年秋ごろまで行った。PCTE と CAIS-A はどちらか一方が優位性を勝ちとることはないだろう。DOD/NATO 標準が一つにまとまるのは、少なくとも 10年先にならうと現在予測されている。暫時、軍用ソフトウェア開発者は、どちらかの標準に合わせるとすると、より実践的なほうへであり、現状では PCTE であろう。

### 4. Portable Common Tools Environment (PCTE)

#### 4.1 概要

PCTE は、欧州共同体 (EC) 拠出金による Esprit プロジェクトの一環として、ツールの公共インタフェース確立を旗印として始まった。この本質点は、ツールを当てはめ差し込めることのできる枠組で、ハードウェアおよびソフトウェアのアーキテクチャの影響をツールが受けないようにするためのものである。ツールの移植性は保証したが、ツール同士のコミュニケーションやユーザインタフェース統合は保証しなかった。

PCTE の心臓部は、オブジェクト管理システム (OMS) とタスク管理やユーザインタフェースなどのためのサービス機能一式である<sup>3)</sup>。プレゼンテーション統合やデータ統合を達成するための基礎となるものである。このようなメカニズムは Esprit プロジェクト

トの後身である Pact プロジェクトで開発され、現在少なくとも GIE Emeraude などの具体化例がある。

PCTE の OMS は批判されてきたが、CASEE のためのデータベースとしては初期の試みであり、そこから多くの教訓を得ることができる。NATO で受け入れられるように準備中の別の版である PCTE+ に教訓は生かされつつある。PCTE+ は 1984 年以来検討の続いている ECMA CASEE 標準 (1990 年制定予定) の基礎にもなっている。

PCTE は、現時点では最も練成された標準案<sup>9)</sup>の一つと言えよう。ライバルとして CAIS-A があるが、実現のテンポはさほど早くはなさそうである。PCTE への批難は、良く理解されていない領域なのに標準化が早過ぎるというものである。PCTE 推進グループの主張は、もし欠陥があるとしても次世代の枠組への有用なフィードバックを得るためにあえて実践の場で使おうということである。

#### 4.2 PCTE の経緯

PCTE は 1983 年から 3 年間、欧州企業よりなるコンソーシアムによって開発された。1986 年 PCTE 仕様の初版が公開され、それに基づく実際の環境の開発が始まった。同年末、PIMB (PCTE Interface Management Board) と称する管理機構が設けられ、PCTE 定義の管理、使用促進、仕様へのコメントや変更要求の管理の任に当たっている。

GIE Emeraude (Bull SPS/7 機上に商用版リリース) のような PCTE の初期のインプリメンテーションは、Unix のカーネル部を変更したものであり、Unix OS の後続のリリースに追従しえないものであった。GIE Emeraude は、その後カーネル部はさわらない仕方で PCTE を再インプリメントした。他の例としては、E 2 S (ベルギー) によるアポロワークステーションへの移植がある。

PCTE への最大の批判は、軍事プロジェクトにとって機密性が不十分であるという点である。1987 年、NATO の IEPG (Independent European Programme Group) は TA-13 という作業グループを作り、民需と軍需の両方のニーズを満たすべく PCTE は最も良く進化しうるかどうか、特に大規模、分散型でしかも安全重視、機密保持重視のプロジェクト分野でどうか調査を進めている。IEPG TA-13 で作られたものが PCTE+<sup>9)</sup>で、PCTE 開発ならびに使用経験を通じて得られた点を考慮に入れたものである。PCTE+ は現在評価中であり、1991 年に定義作業を完

了する予定になっている。その結果により、PCTE+ を完全 NATO 仕様へ押し進め、ソフトウェア開発プロジェクトにおいて強制力あるものにしてようと IEPG は考えているようである<sup>9)</sup>。

1988 年、PIMB は ECMA に ISO 向けの枠組標準として PCTE 仕様案を準備するよう依頼した。ECMA の TC-33 は案の出発点として PCTE+ を用いつつある。ECMA は標準制定を 1990 年中葉に予定している。

#### 4.3 PCTE 枠組

PCTE は、ソフトウェアツールのための基盤であり、標準 OS によっては提供されない管理サービス、サポートサービスを提供するものである。これは、統合 CASE に必要な機能一式で枠組と称される。PCTE 仕様は、枠組とツールとのインタフェースを定義するものである。この枠組は、言語独立である (現在までは C と Ada のみサポートしている)。

PCTE で重視されていることの一つに、オブジェクト管理システム (OMS) がある。オブジェクトの粒度は、Emeraude 実現例では粗く、一般に適当な粒度水準は得られていない。また制御統合機能は PCTE ではさほど取り上げられてこなかった。制御統合機能は、7. の CIS に興味深い考え方がある。

### 5. PCTE+ 参照モデル

#### 5.1 背景

主たる経緯は、1988 年 ECMA TC 33/TGRM (Task Group on Reference Model) 発足、1989 年案初版作成、1990 年春モデル評価 (さまざまな CASE 統合枠組を初版モデルへ写像) 開始である。

図-1 の CIS 全体構造モデル (G. Tatge のトースターモデル) と本モデルは公開され関心を集め、TC 33 と CASEE コミュニティで現在評価中である。ECMA/TC 33 TGRM 案は難解な抽象度の高い概念である。分かりやすくするため、TGRM モデルへ PCTE+ を写像しながら簡潔に解説した文献があるので参照されたい<sup>7)-9)</sup>。

#### 5.2 参照モデルに対する要求

参照モデルに対する要求は次のとおりである。

- 参照モデルは、既存のあるいは提案中の環境を比較/対比/記述するのに適していること。
- 異機種環境でのツールの移植性/相互運用性/インタラオペラビリティに着目すること。
- 広範な CASEE の設計記述に使用され得るべきで

- ① オペレーティングシステム、
- ② データレポジトリ
- ③ データ統合サービス
- ④ ツール差込みスロット
- ⑤ タスク管理サービス
- ⑥ ユーザインタフェース
- ⑦ メッセージサーバ  
ネットワーク

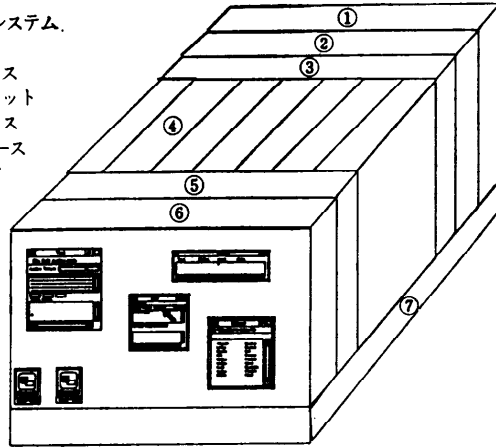


図-1 参照モデルの全体構造 (G. Tatge 図)

あるが、有用な標準を定義できなければならないという要求とも釣り合っていること。

- ソフトウェアおよびシステム技術者に CASEE を教育する基盤であること。

- 標準を円滑に進歩させるものでなければならない、特に後々の標準が上方向互換性をもって容易に達成されるように初期の標準が開発されること。

図-1 は、参照モデルの構造全体を示す。その各部と深い関係にある PCTE+ 各要素を項目ごとに対比させて説明しているのが文献 10)、11) で、一読をお勧めする。

## 6. IEEE

ツール接続性に主眼を置いて、IEEE タスクフォース P1175 によって検討されてきている内容を紹介する。その骨子は、人間組織とツールの接続、ツールとプラットフォームの接続、ツールとツールの接続の三面よりなる枠組をもつ。まずこの三面よりなる枠組に他案にはない特色をみることが出来る。前二者に関して IEEE の現時点の案は参照モデルで、最後者に関しては標準案であり内容的に性格の異なる二種類を一括して扱っている。今後、前二者について、参照モデルを念頭に、標準案の作成へ進む基礎固めが行われるであろうかたわら、最後者の標準案の適用・評価・改善を周期的に行ってゆくはずである。以下に背景と経緯、ツール間接続標準案、参照モデルを概観しよう。

### 6.1 背景と経緯

ツールを組み合わせて使おうとしている者は多い。市場に出ているツールの多くは、オープン・アーキテ

クチャ、インポート/エクスポート能力、ファイル形式の文書による提示を約束しているが、実際はツール接続はそう容易に行えるようになってはいない。困難になっている理由は、ツール間で転送する情報の意味、形式、転送方法が判然としないことによる。

意味の問題……ツール A と B があるとして、転送情報を同じ仕方で解釈しているか？ いかによく、ツール A と B が転送情報の形式と転送方法について合意しあっても、情報解釈の仕方が異なっていれば、それらツールは協働しえない。たとえば、ツール A は状態にデータ操作プロセスを含めているのに対して、B はデータ操作プロセスに状態を含める前提に立っているという具合である。

形式の問題……ツール A と B は同一のデータ構造や要素を認識できるか？ たとえば、ツール A は OSI の ASN.1 標準ファイル形式を、B は International Graphics Exchange 標準を使っているという具合である。ペンダ提示のファイル形式を見てトランスレータを作ることはできる。形式/構文の差異を解決するには、トランスレータ作成の工数とコストを見込まねばならない。

転送方法の問題……いかに情報が A から B へ転送されるか？ たとえば、A は情報をファイル、IRDS を介して中央データベースまたは通信チャンネルへ書く。B はそのどれから読むことができるか。物理転送を確実にする一つの方法は、オープンアーキテクチャとインポート/エクスポート能力に頼ることになる。

1986 年、IEEE の本タスクフォースが、ツール使用者、作成者を対象に実態調査をしたところ、標準が不

十分である旨の声が多かった。そこで、既存標準のうちツール接続に関係する 247 の標準（業界承認済）と、いろいろな機関で検討されている 11 通りの標準をレビューした。そこで分かったことは、既存標準は、媒体形式、ファイル形式、フィールド形式、図形表現、言語バインディング、OS インタフェース、データベースインタフェース、ヒューマンインタフェース、ハードウェアインタフェースについてであり、転送情報形式と転送方法については十分であった。情報の意味については、VHDL<sup>12),13)</sup>があるのみで、しかもそれはハードウェアモデルにのみ関与するだけである。1987年、本タスクフォースは、ソフトウェアの意味に焦点をあて、そこにおける標準案<sup>14)</sup>を作成することにした。その際の基本方針は、

- 既存標準が適用可能なら、再利用するか参照する、
  - 既存ツールへの適用のみならず、今後開発されるであろうツールにも適用できる標準であること、
  - 人間に認識できる形式のみならず、機械認識可能であること、
- などである<sup>15)</sup>。以下に、IEEE 案の最新内容の骨子を述べる。

6.2 ツール間接続標準案

ツール対ツール接続において、転送される情報の記述の仕方を、標準案は与える。案の中身は、標準テキスト言語 (STL) で、データ、状態、イベント、条件

などのソフトウェア概念を扱えるのみならず、変換、制御遷移、状態遷移などのオペレーションも扱える。図式表現と別にテキスト表現をもっている。ツールはいずれかまたは両方を使えばよい。図-2 は、あることを表現する二通りの仕方を示す。ここでは、データフロ Df1, Df2, Df3, Df4, データストア Ds1, プロセス P1 それぞれの STL による記述例を示している。このように記述することにより、いかなるツールも情報の意味を正しく解釈できる。この例では現れないが、状態遷移オペレーション、制御オペレーション、データオペレーションなどほとんどすべてのソフトウェア概念を STL で記述できる。

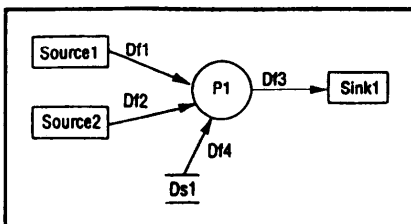
6.3 参照モデル

上述のツール間接続インタフェースは IEEE の現案では標準として提示されているが、ツール対人間組織およびツール対プラットフォームの各相互接続インタフェースについては参照モデルとして示されている。

6.3.1 人間組織対ツール相互接続の参照モデル

組織活動の主旨・目的に沿って、ツールが使われるなら、その効果は大きい。もしそうでなければ、1) ツールは使われない、2) 最適とは程遠い仕方でツールが使われる、3) ツールは改造される、4) 組織のほうを手直しする、のいずれかとなろう。ツールもしくは組織を変更しなければならないなら、そのコストを事前に評価せねばならない。

ツールの組織活動に対する相互接続性は、次の三つ



(a)

*Dataflow*  
is used by process  
is provided by source  
is pictured by graphic  
carries data item

Df1  
P1;  
Source1;  
DfGraphic1;  
Di1.

*Dataflow*  
is used by process  
is provided by source  
is pictured by graphic  
carries data item

Df2  
P1;  
Source2;  
DfGraphic2;  
Di2.

(b)

*Dataflow*  
is received by sink  
is produced by process  
is pictured by graphic  
carries data item

Df3  
Sink1;  
P1;  
DfGraphic3;  
Df5, Df7.

*Dataflow*  
is used by process  
is pictured by graphic  
carries data item

Df4  
P1;  
DfGraphic3;  
Df7, Df8.

*Data store*  
receives dataflow  
provides dataflow  
is pictured by graphic  
holds data item  
has retention time

Ds1  
Df6;  
Df7, Df8;  
DsGraphic1;  
Ds6, Df7, Df8;  
8 hours.

*Process*  
is type  
uses dataflow  
produces dataflow  
references data store  
contains operation  
is pictured by graphic

P1  
data transformation  
Df1, Df2;  
Df3;  
Ds1;  
Do1.1, Do1.2;  
ProcessGraphicP1.

図-2 IEEE 標準案：ソフトウェア概念記述例 図式表現 (a) とそのテキスト表現 (b)

の側面から考えられよう：

1. 任務、職務機能：だれがそのツールを使うか？  
何のために使うか？
2. ライフサイクル：プロジェクトのどの段階でそのツールは使われるのか？
3. 支援要素：いかにツール使用がプロダクティブになるか？

これらの三つの側面について、現 IEEE 案は、既存標準を整理して提示するとともに、標準がない側面については、ツールと組織活動の対応ぐあいをプロフィールとして記述することを勧告している。

### 6.3.2 ツール対プラットフォーム相互接続の参照モデル

以下の要素よりなるプラットフォームとツールとのインタフェースのための参照モデルを示す：

- ハードウェア
- OS およびその関連機能
- DBMS
- コミュニケーションシステムズ
- ヒューマンインタフェース
- 環境制御

組織が単一種類のプラットフォームしか使っていないのであれば、それ用のツールを使うことで問題はない。しかし、たいいてい組織では、さまざまなプラットフォームを使っている。

ツールに対する投資によって得られる効果は、そのツールが異なるプラットフォーム上でも、ツール/手続/使用データを変更することなく使用しうるなら、著しい。ツール改造や手続/使用データ再作成に時間と工数を要するなら、資源最適利用からほど遠くなる。もし、ツールが異なるプラットフォームに対しても相互接続され得るなら、そのツールは組織の必要に応じ使用でき、より好都合である。

いかに良い相互接続かを論ずる際、前提条件となることは次のとおりである：

1. プラットフォームのネットワークの構成
2. いろいろなプラットフォームへのツールの分散
3. 各プラットフォームの構成

また、ツール対プラットフォームの相互接続の良さは、

- そのツールがいかに標準に基づいているか
- そのツールがどのようにプラットフォームサービス制御機能を使っているか

といった見方から論ぜられよう。

## 7. CASE Integration Services (CIS)

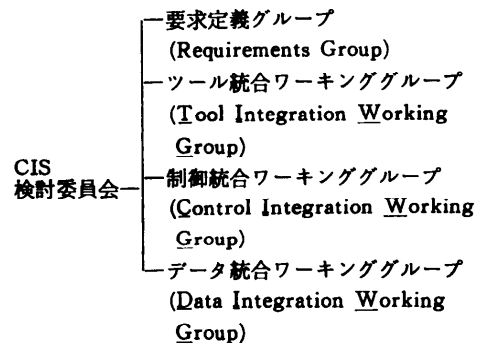
CIS は統合側面のうち、制御とデータの統合に力点を置いた CASEE 枠組<sup>16)</sup>である。実時間 CASEE 領域を対象としている。

### 7.1 経 緯

1987 年末、DEC 社と Atherton Technology 社が、後者の Software Back Plane™ と称される製品に基づいて CASE 統合サービスを提供できそうか否か検討したのが CIS の始まりである。1988 年 1 月、両者はツール統合のためのオブジェクト指向枠組の検討を開始し、同 4 月、ATIS (Atherton Tool Integration Services) と呼ばれる提案を欧州の PCTE+ ワークショップの場へ提示した。ATIS と PCTE+ の二つを対比するためである。同年 7 月、ATIS をツール統合のベースにできないかとの検討を一連のコンピュータメカ、ツールベンダ、ユーザに依頼した。同年後半、標準検討母体としての組織の名称を ATIS から CIS 委員会に変えた。この改称や主査を SPC (Software Productivity Consortium) から選出したことは中立性を高める配慮からであった。

### 7.2 検 討

事実、委員会内で主関心事の衝突が生じてきた。コンピュータメカは、より進んだ標準と互換性がありしかも私企業所有的な統合枠組を望んだが、ツールベンダは、極力痛みのない統合案を望んだ。突破口を開くグループが結成され、オブジェクト指向メッセージパッシングネットワークアプローチと称す提案 (もともと 'M' と呼ばれた) を出した。このグループは現在は CIS 内での正式地位を与えられ、CIS-CIWG (Control Integration Working Group) と称されている。検討は下図に示すように分担して行われているとのである。



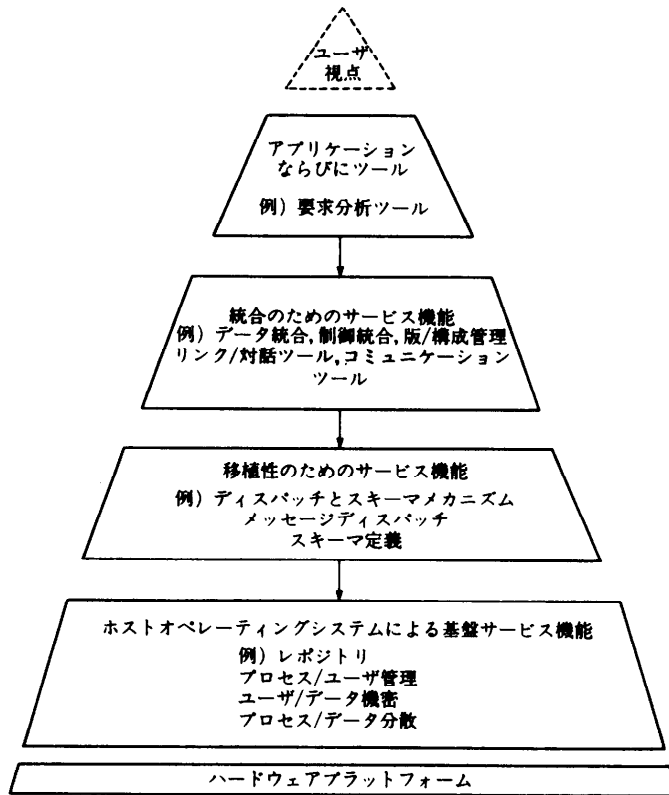


図-3 CIS 参照モデル

### 7.3 CIS 参照モデル

CIS 要求定義グループは、ATIS を評価し、CIS 活動スコープ設定の責任のもとで、参照モデルを草案した<sup>17)</sup>。サービス機能は、大略三つの層を成している。

図-3 参照。それらは、以下のとおりである。

- 1) 基盤サービス：ホスト環境に備わっていると想定される主要サービスである。たとえば PCTE またはホスト機オペレーティング環境には備わっているようなサービス。
- 2) 統合サービス：データ統合、タスク管理、メッセージサブネットワークなどのサービスである。
- 3) クライアントアプリケーション：この場合、ツール提供者からの開発および管理ツールが相当する。

CIS はプレゼンテーション統合は扱っていない。他で検討されている標準と重複することを CIS は極力避け、ユーザインタフェース標準は OSF (Open Software Foundation) や UI (UNIX International) に

ゆだねている。

## 8. Information Resource Dictionary System (IRDS)

情報(処理)システムや CASEE からのニーズに応えるべく、IRDS は開発されてきた。物理的にデータ格納を行う PCTE の OMS 的なものとしてでなく、より進んだデータディクショナリとしてである。IRDS とデータベース管理システムを一緒にして、レポジトリと呼ぶ。レポジトリにデータを物理的に置く。

IRDS は、共通データモデル標準であり、それに対しツールを適合させる。IRDS は、環境枠組に関心をもつ人でなく、ツールベンダによる標準として推進されてきたので、どちらかという成熟したデータモデルが、主に取り上げられてきた。

IRDS の ISO<sup>18)</sup>や ANSI<sup>19)</sup>案はともに、3層枠組である。

1) 最下位層 データベース自身であり、データベースに格納されるものは、オブジェクトの個々のインスタンスである。

2) 第2層 (IRD 定義レベル) ある環境で許されるオブジェクトの型、属性、関係などを含む DB データモデルを記述。

3) 第3層 (IRD スキーマ記述レベル) レポジトリシステムの構築者に必要な抽象レベルで、第2層の解釈の仕方を教えるものである。

ISO の IRDS 枠組は、カスタマイズ性の高いレポジトリを追求しており、いろいろなデータモデルを扱えるようにしようとしている。しかし、CASE ツールの移植性や移植にともなうデータ統合には、これまであまり配慮していない。

## 9. おわりに

紙数の制限から割愛せざるをえなかった標準(案)も多かった。米国電子産業協会 (EIA) による ANSI 向け標準である CASE Data Interchange Format (CDIF), National Institute of Standards and Technology (NIST, 旧 NBS) の ISEE 枠組,  $\Sigma$  などがそれである。内容水準や影響力が低いとみただけではなく、良い時宜を得て取り上げてみたいと思っている。今回取り上げたもののさらに詳細についても同様である。

枠組や参照モデルをいくつか眺めて思うことは、狙ったスコープに対して概念枠組が妥当であるか、特にデータあるいは制御などの統合技術が先行して枠組をゆがめていることはないか、参照モデルは既存標準が対応できていない側面をどうすべきか端的に言及しているか、標準化すべきことの根拠と方向づけの本質点が薄れてはいないか、検討され提示された標準案は良いかどうか標準を使用する側からの評価が切望されるなどである。言うは易し答うるは難しである。CASEE の標準化は、CASEE 開発者主体でなく、CASEE 使用者または構築者 (インテグレータ) がイニシアチブを取り、皆が協力するのが有用な良い標準を生む近道ではないかと思う。

**謝辞** 本稿をまとめるに際し、François Coallier (Bell Canada), Robert Poston (Programming Environments), Margaret Law, William Wong (NIST), Frank Davis (US Navy), Marv Zerkowitz (University of Maryland), Anthony Earl (HP Lab in Bristol), Caroline Chappell (Ovum) とは貴重な議論をも

つことができ、ここに深謝します。本件のために時間をさき高所からコメントをくださった日本電気(株)専務取締役水野幸男博士ほか多くの方々特に感謝します。

本稿に記した参照モデルや標準は検討途次のものであり、今後変更される可能性は多分にあると思われるので、念のため付記します。

## 参考文献

- 1) Buxton, J. N. and Druffel, L. E.: Requirements for an Ada Programming Environment: Rationale for STONEMAN, Proceedings of IEEE Computer Society's International Computer Software and Applications Conference, pp. 66-72 (1980).
- 2) U. S. Department of Defense: Common Ada Programming Support Environment Interface Set (CAIS), Revision A, Proposed DOD-STD-1838 A (1988).
- 3) CEC, PCTE: A Basis for a Portable Common Tool Environment, Functional Specification Version 1.5 C Volume 1, Basic Mechanisms including Object Management System, Draft for Public Review, Nov. 15 (1988).
- 4) Boudier, G., Gallo, F., Minot, R. and Thomas, I. M.: An Overview of PCTE and PCTE+, Proc. of the ACM SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environments, SIGSOFT Software Engineering Notes, Vol. 13, No. 5 (1988).
- 5) ECMA/TC 33/89/48: Introducing PCTE+, IEPG TA-13 (1989).
- 6) Chappell, C., Downes, V. and Tully, C.: Real-time CASE: The Integration Battle, Ovum (1989).
- 7) Earl, A.: A Reference Model for CASEE, A Conceptual View Points (draft 2.0), ECMA/TC 33/TGRM, 89/032 (1989).
- 8) Earl, A.: A Review of PCTE+ (Issue 3) According to the CASEE Reference Model (draft 3.0) ECMA/TC 33/TGRM, 90/001 (1990).
- 9) Earl, A.: Principles of a Reference Model for Computer Aided Software Engineering Environments, in Software Engineering Environments, Proc. of International Workshop on Environments, Chinon September, Springer-Verlag Lecture Notes in Computer Science Series (1989).
- 10) Crispin, R. and Earl, A.: A Reference Model for CASE Environments-Part 1, PCTE Newsletter No. 3, pp. 15-20 (1990).
- 11) —: A Reference Model for CASE Environ-



- ments, Part 2, PCTE Newsletter No. 4, pp. 15-19 (1990).
- 12) IEEE Std-1076, Standard VHDL Language Reference Manual, Catalog No. 983 (1987).
  - 13) Armstrong, J. R. : Chip-Level Modeling with VHDL, Englewood Cliffs, Prentice Hall (1988).
  - 14) IEEE : A Standard Reference Model for Computing System Tool Interconnections, P 1175 Draft D 6.3 (1990).
  - 15) Tice, G. (Editor), Poston, R. : Proposed Standard Eases Tool Interconnection, in Software Standards IEEE Software, pp. 69-70 (1989).
  - 16) ANSI X 3 H 4 Working Draft, Information Resource Dictionary System ATIS (1990).
  - 17) Nolan, C. J. : CASE Integration Services Technical Description, Doc. No. CIS 90-008, Revision 1.0 (1990).
  - 18) ISO/IEC DIS 10027 Information Processing Systems IRDS Framework (1989).
  - 19) Technical Committee X 3 H4, Information Resource Dictionary System (IRDS), X 3. 138-198 X BASE Project 336-D (1986).

(平成 2 年 6 月 14 日受付)

---