

## 解説

## PCTE：ソフトウェアツールを移植可能にし、共通に利用するための環境

PCTE: Portable Common Tool Environment by Yoshihiro MATSUMOTO (OSAKA INSTITUTE OF TECHNOLOGY) and Matsuki YOSHINO (Hitachi, Ltd. Software Development Center, Artificial Intelligence Department).

松本吉弘<sup>1</sup> 吉野松樹<sup>2</sup>

<sup>1</sup> 大阪工業大学情報科学部情報システム学科

<sup>2</sup> (株)日立製作所ソフトウェア開発本部 AI 設計部

### 1. まえがき

PCTE (Portable Common Tool Environment の略称) とは、ソフトウェアの開発、製作、試験、保守などを支援するために使われるさまざまなソフトウェアツール (以下、ツールとよぶ) を統合的に利用するための環境の 1 つであり、この環境に組み込まれるソフトウェアツールが、移植可能性 (異なる計算機環境においても利用可能な性質)、および共通性 (多くのユーザによって共有される性質) をもつための基準を満たすように作られた環境を意味する。PCTE に関する国際規格 (ISO/IEC 13719-1) は、ここでいう基準を定めたものであり、ツールは、この目的に従って作られた環境に組み込み可能にすることによって、移植可能性および共通性をもつことが期待されている。

### 2. PCTE とは

PCTE という名称は、1983 年に、欧州連合委員会 (CEC: Commission of the European Communities) の ESPRIT-I (European Strategic Programme for Information Technology-I) プロジェクトのなかで、1 章で述べた基準の策定が取り上げられたときに作られたものである。現在では、PCTE という基準、もしくは基礎概念は、ツール以外のソフトウェアモジュール、たとえばコンポーネントウェア (一般的なインタフェース記述言語によってそのインタフェースが記述され、ソフトウェア共通基盤に組み込まれたソフトウェアモジュール) を統合的に利用するための環境に対しても、拡張して適用されている。

通常、ソフトウェアの開発、製作、試験、保守など (以下、ソフトウェア・エンジニアリングとよぶ) は、異なるツールを逐次、または並行に組み合わせ、実行しながら進められる。ツールの種類はきわめて多いが、さらにこれに加えて、同一の機能をもったツールでさえ、その種類は多岐に渡る。ユーザは、自分の目的や利用条件からみて最適なツールを選び、それらを組み合わせてソフトウェア・エンジニアリングを実施する。

ツールの組合せにあっては、ツールからツールに渡される情報 (たとえば、仕様記述、プログラム・ソーステキストなどを記憶したファイルなど) であり、以下、実体とよぶ) が存在し、ツールの間で交換される制御信号が存在する。ツールからツールへの実体および制御信号渡しが完全に遂行され、ツールが互いに同期して目的の機能を果たすためには、実体および制御内容について、それを表している表現規則、意味定義を、当該ツールが完全に受け入れ可能 (解釈可能) とする必要がある。しかし、製作者の異なる市販ツールを組み合わせるような場合には、実体および制御に関する表現規則、意味定義の不統一にしばしば遭遇し、異なる製作者の市販ツールを組み合わせることは、通常、きわめて難しいものと考えられている。

1980 年代に、開放型ソフトウェア環境へのニーズの高まりとともに、どのようなツールにも解釈できるような形式で実体を表現し、記憶し、管理するとともに、ツール制御およびツール間の連携制御をも司るようなリポジトリ (保管庫, repository) があれば、実体を複数のツールが共有することができ、ツールの連携も行うことがで

さるであろうとの発想が生まれた。ツール間で交換される実体はすべてこのリポジトリを経由して行うことによって、異なる製作者の市販ツール間でも実体の交換が可能となり、流通されているツールを組み合わせることで使う可能性が増大すると考えられた。

PCTEは、ソフトウェアリポジトリを用いて、次のような方法でツールを統合するシステムに対して適用される。

- (1) リポジトリは、ツールにとって必要なデータ集合の集まりを、データ集合と、データ集合間の関係を表したモデルに準じて記憶し、十分な版管理、構成管理、機密保護を施すとともに、それが情報ネットワーク上に分散して配置されていても、並行に実行されるツールに対して、配置位置透明、安全、かつ信頼のおけるデータサービスを提供する。
- (2) リポジトリは、ツールに対して、データの読み書きに関するインタフェースと、制御に関するインタフェースをあわせて提供する。後者、すなわち制御のためのインタフェースは、ツールがほかのツールを起動し、それらとの間でデータを授受するためのものである。これらインタフェースは、これを介して相互運用するツールが移植可能性、共通性をもつように定められる。

### 3. PCTEの仕様

#### 3.1 PCTEの機能

PCTEは、上に述べたツール統合のためのフレームワークを実現するために必要となるデータ統合、および制御統合サービスを提供する。具体的なサービスは次のとおりである。

- (1) データモデリングおよびデータ管理機能を提供し、複雑なデータのもつ意味制約や構造を保持できるようにする。
- (2) データモデルの拡張、修正、および可視範囲の多重選択を可能とするため、ERA (entity-relationship-attribute) モデルに基づいたスキーマ管理を行う。
- (3) 版および構成管理のために必要なサービスを行う。
- (4) ネットワーク上に配置された記憶装置に

分散して記憶されたデータに対する透明なアクセス (記憶位置に依存しないアクセス) を可能とし、ネットワークの故障に対処できるフレキシビリティと回復性を提供する。

- (5) 目的とする処理を実行するために、ネットワーク上に配置された複数の計算処理装置に分散し、並行するプロセスを生成し、これらに機能遂行を任せることができる。この場合、1つの処理装置上のプロセスは、ほかの処理装置上のプロセスによるサービスを受けることができるようにする。
- (6) データに対し、並行するプロセスによる同時アクセスがあった場合には、競合を回避し、データの一貫性を損なうことがないようにする。
- (7) トランザクションが、入れ子になったり分散されたりしていても、システムの故障に際して、データの原子性、一貫性、隔離性、不揮発性を保証するものとする。
- (8) データを保護するために必要な、アクセス制御、一貫性保護、機密保護、監査に関するサービスを提供する。
- (9) ツールが相互に通信するために必要な、パイプ (pipe)、メッセージキュー (message queue)、イベント (event)、通知 (notification) などに関するサービスを提供する。

#### 3.2 データ管理

PCTEリポジトリは、データを、オブジェクト (object)、リンク (link)、属性 (attribute) と称する3種類の要素によって管理する。これら要素の組立ては、ERAモデルの表現を基本とするが、オブジェクト指向データベースでしばしば採用されているデータモデルに近い拡張がなされている。

オブジェクトは、一般にPCTE内にある形をもったソフトウェア実体 (たとえばビジネス・データ集合、ソースコード・モジュール、実行可能コード、テスト結果などの記述およびそれらの記憶、ファイル、または個々のデータの単独記憶) を意味する。これに加えて、オブジェクトが、PCTE内の抽象 (たとえばタスクのような実体をもたない制御のための対象) や、PCTE外に

あって PCTE が扱う世界に含まれる実体の抽象表現 (たとえば人, 役割, 組織, 規格など) を表すこともある。

方向が異なり, 両端のオブジェクトを共有する 2 本の矢の組 (pair) をリンクとよび, リンクは両端のオブジェクト間の関係を表す。たとえば, 片方のオブジェクトが要求仕様, 他方のオブジェクトが設計仕様を表すとき, リンクは各方向へ向けてのトレース (trace: 対応の道) を表すことになる。個々のリンクには, 特性 (property) と称する注釈を与えることができる。特性は, リンクの用途に応じて定められるリンクカテゴリから決まる場合と, 明示的に与える場合がある。これによって, リンクの両端のオブジェクト間に存在する依存や制約関係が表される。オブジェクトとオブジェクト間の関係が, 当該オブジェクトを選

んで自由に張れるリンクと, リンクの性質を特性定義によって個別に管理できるところが, 関係型データベース (relational database) に比べてこのデータモデルの優れているとされる所以である。また, 直接にリンクされていないオブジェクトでも, それがリンクを辿っていきつくことができるものであれば, リンクの接続によって関係づけすることができる。

オブジェクト, リンク, および属性は, 型をもつことができる。表示された内容が, 特定のオブジェクト, リンク, または属性だけを指す場合には, そのオブジェクト, リンク, または属性を, 該当する型のインスタンスであるという。

以上に述べたことを, 人と, 人それぞれに割り当てられたタスク (仕事) との関係を表すオブジェクトモデルの例を用いて説明する。図-1 の上

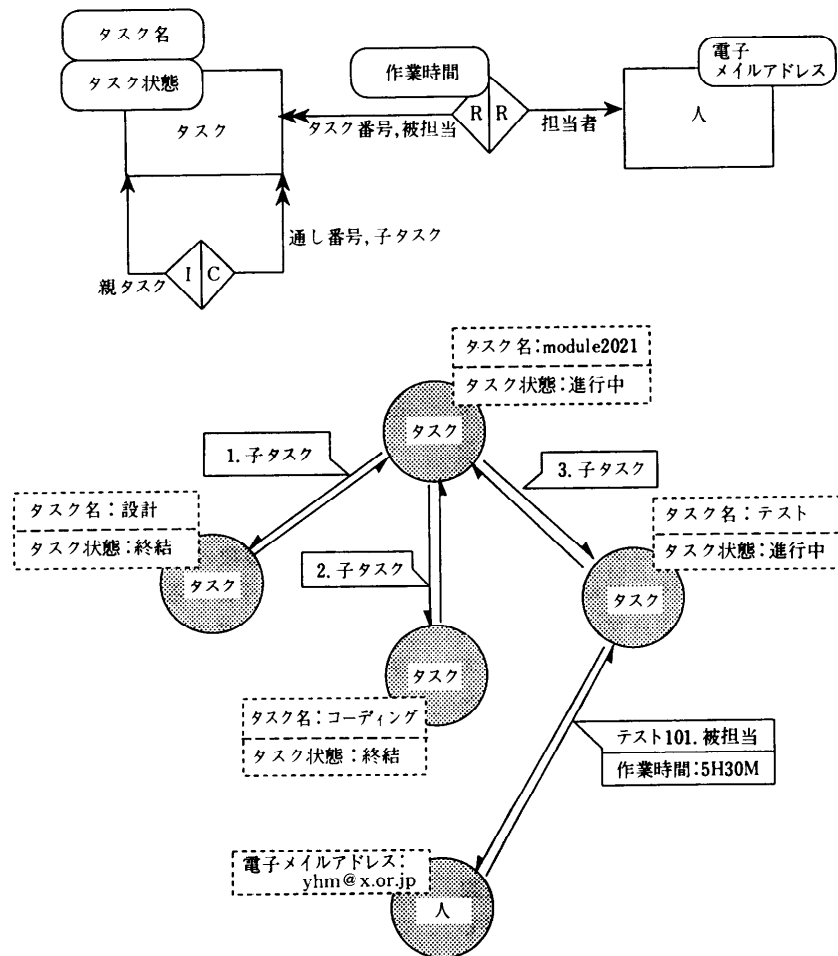


図-1 データ型とインスタンスの例

半分は、このモデルに組み込まれたオブジェクト型を示し、図-1の下半分はこのオブジェクト型に当てはまる1つのインスタンスを示している。図-1の上半分で、長方形はオブジェクト型、長円は属性型、菱形はリンク型を表している。矢に付随する記述はリンク名を表し、記述が終止符を含む場合には、終止符の前におかれた記述がキー属性、後ろがリンク名を表す。菱形のなかに記入されたローマ字は、リンク・カテゴリを表す。

図-1の下半分は、図-1の上半分で示されたタスク型に属する1つのインスタンス“Module 2021”が、3つの子タスク（いずれもタスク型に属する）をもち、第1の子タスクが設計、第2がコーディング、第3がテストであり、テストのなかの1つのインスタンスであるテスト101は、yhm@x.or.jpというメールアドレスをもつ人に割りつけられていることを示している。

### 3.3 スキーマ管理

3.2節で述べたようなオブジェクト、リンク、属性の定義をERAモデルで表現したスキーマをPCTEではSDS (schema definition set) とよぶ。SDSは、たとえばツールの供給者によってそのツールが扱う情報のメタ情報としてツールとともに供給される。

ユーザが複数のツールを組み合わせる開発環境を構築しようとする場合には、複数のSDSを組み合わせる必要がある。PCTEでは、複数のSDSからなる集合体を、ワーキングスキーマとして定義することができる。複数のSDSをまとめた場合に、SDSに含まれる型の統合が行われる。利用者は、自分が利用するSDSの集合を実行時に定義し、ワーキングスキーマとすることによって、より効率的にオブジェクトを操作することができる。ワーキングスキーマは、通常データベース管理というサブスキーマ、またはビューに相当する。通常データベースでは、サブスキーマ、またはビューを、実行時に追加することはできないが、PCTEのワーキングスキーマは、これを可能とする。

子型 (subtype) は、型のもつ性質を継承し、その性質のある部分をより特殊化して新しい型として定義したいときに利用できる。子型は、複数の異なる型 (親の型) の性質を継承することができる。継承機能は、オブジェクトのもつ性質をさ

まざま異なるレベルに具体化して用いたいときに役に立つ。ユーザマニュアルと設計仕様書があり、これらを文書という同一の型として扱いたい場合とそれぞれ別の型として扱いたい場合には、文書という型を定義したSDSと、文書という型を取り込んでその子型としてユーザマニュアルと設計仕様書という型を定義したSDSを用意し、必要に応じてどちらかのSDSをワーキングスキーマに取り込めばよい。

### 3.4 版および構成管理

個々のソフトウェア部品、またはその構成は、時間の経過とともに何回も変更されるであろう。また、それぞれはそのまま維持されるが、自分と並行して別の版をもち、それぞれが維持されて、次の機会に利用されることになるかもしれない。PCTEでは、個々の部品、版、構成を、独立に、またはそれを含む複合体とともに管理し、扱うことを可能にする。

### 3.5 分散データ管理

PCTEリポジトリは、ネットワーク上に分散された複数の記憶装置にまたがって配置されたオブジェクトを、それらの物理的配置を意識しなくても扱えるようにする便宜を提供する。オブジェクトは、場合によってその存在位置を変更するが、ユーザはこれも意識する必要がない。

複製 (replication) 機能により、共用される重要な情報を、異なる装置上に適切に分散配置することによって、部分的な装置の故障に際して、耐故障性を保持することができる。

### 3.6 分散プロセス管理

PCTEは、実行可能、または解釈可能なコードを、オブジェクトベースに記憶させ、これを1つ、または複数のプロセスとして実行、または解釈させるようにすることができる。さらに、オブジェクトベースの表現のなかで、プロセスに対する演算処理装置への割つけを指定することができる。

プロセス、およびプロセス環境 (動作中のプロセスの状態、たとえばオープンされているファイル、環境変数の値、現在のワーキングスキーマなど) もまたオブジェクトベースのなかで表現されている。そのためPCTEの制御下で、プロセスの状態を観測したり、サービスの制御を行うことが可能である。

### 3.7 並行制御

リポジトリに対する要求の1つに、多くのユーザやツールからのアクセスに対して、オブジェクトに対する高性能、かつ安全なアクセスを許すということがある。PCTEではロック制御によりこの要求を満足させている。PCTEでは、ロックは、アクティビティという単位が取得する。アクティビティは、プロセスの一部であっても、複数のプロセスにまたがってもよい。ロックの対象となる単位は、オブジェクト、またはリンクである。故障が起きたときには、ロックは自動的に解除される。

### 3.8 分散トランザクション管理

トランザクションは、前節で述べたアクティビティの1つのタイプである。アクティビティのオブジェクト集合に対するアクセスの結果の一貫性を保証したいとき、アクティビティをトランザクションとして定義する。

PCTEでは完全ロールバック方式、すなわち、トランザクションのなかで故障が起きた場合にはトランザクションの始点における状態をそのまま持続し、トランザクションが成功し、完結した場合にだけ、オブジェクト集合を更新する。トランザクションは入れ子になっていても構わない。また、トランザクションに含まれるPCTEプロセスやデータアクセスは、どの物理的位置で行われてもよい。

### 3.9 セキュリティ管理

軍が利用する目的（たとえば、米国国防省標準CSC-STD-001-83を満たすような）から生まれる高度なセキュリティを満たすように考えられているので、銀行など高度な民生用システムにも十分利用可能であると考えられる。セキュリティとは、不正な行為、または予期しない事態に遭遇した場合に、システムがオブジェクトの破壊および不正アクセスを防ぐ能力をもつ度合いを意味する。PCTEでは、次の方法で、セキュリティを維持する。

任意アクセス制御：個々のオブジェクトごとに任意にアクセス権を定義し、定義された権利の内容に従って、ユーザ個人ごと、グループごと、またはプログラムごとにアクセス自由度を制限することができる。

強制的な一貫性制御：一貫性レベルの高い情報が

一貫性レベルの低い情報源（ユーザ、ツール、ほかの情報）などによって汚染されることを防ぐ機構である。たとえば、一貫性レベル「中」のプロセスが、一貫性レベル「高」のオブジェクトにアクセスする場合、書き込みは許されず、読み出すことは可能とされる。一方、このプロセスは一貫性レベル「低」のオブジェクトからは読み出すことは許されない。

強制的機密性制御：情報およびプロセスに機密レベルを与え機密情報の流出を防ぐ機構である。機密レベルを用いた強制的保護は、前記の強制的な一貫性制御とほぼ同じ方法で行われる。しかし、プロセスは、あるオブジェクトから読み出したデータを、それより機密レベルの低いオブジェクトに書き込むことは許されない。

監査：セキュリティに関連するイベントが発生した場合、それらを必要に応じて記録する。

### 3.10 プロセス間通信

PCTEは、メッセージ転送、およびメッセージキューのための機能を備えることによって、プロセス間の通信をサービスする。特定のデータがアクセスによって変更されると、PCTEのオブジェクト管理システムがこれを検出し、自動的にその内容をメッセージ転送によって関連するユーザやツールに伝達する通知の機能もあり、ツール間の連絡などに利用できる。

## 4. 国際標準となるまでの経過

2章で述べたESPRITのPCTEプロジェクトは、1988年に当時の最新版であるPCTE 1.5をECMA (European Computer Manufacturer Association) に提案した。ECMA/TC 33は、これを審議し、ECMA-149 (抽象仕様)、ECMA-158 (C言語へのバインディング)、ECMA-162 (Ada言語へのバインディング) を制定し、続けて1993年には、これら規格の第2版を、ISOに対しファーストトラック手順でISO標準にするよう提案した。ISOでは、SC 22が担当となり、1994年3月を期限として郵便投票を行った。日本ほか1カ国が反対し、2カ国が棄権、20カ国が賛成した。

日本は、投票に以下のようなコメントを付記した。

(1) マルチバイト文字の扱いが不明確で

- ある。
- (2) 当時、すでに ISO/IEC 10728 として ISO 標準となっていた IRDS (Information Resource Dictionary System) との位置づけが不明確である。
  - (3) 仕様の範囲が広範であるので、サブセット仕様が必要である。
  - (4) 仕様の形式が ISO 標準の形式となっていない。
  - (5) 仕様案の内容に曖昧な点、矛盾する点など不備がある。(64 件を指摘)

日本の投票は、条件つき反対(条件が満足されたなら賛成する)であり、上記コメントに対し納得のゆく回答が得られることがその条件であった。1994年7月にジュネーブで会議がもたれ、日本からの提起を含む問題点の解決がはかられ、日本も最終的に賛成の意を表した。同年9月のハーグでの SC 22 総会でその解決結果が承認された。また、PCTE 規格の拡張、改訂を担当する新しい WG 22 の設立が決定された。

以上の経緯を経て ISO から発行された文書は次のとおりである。

- (1) ISO/IEC 13719-1 Information technology-Portable Common Tool Environment (PCTE) - Part 1: Abstract specification (1995年6月1日)
- (2) ISO/IEC 13719-2 Information technology-Portable Common Tool Environment (PCTE) -Part 2: C programming language binding (1995年6月1日)
- (3) ISO/IEC 13719-3 Information technology-Part 3: Ada programming language binding (1995年9月15日)

通産省工業技術院は、上記(1)に関する PCTE 要約 JIS 制定を策する方針を起案し、JIS 原案作成を日本規格協会情報技術標準化センターに委託し、その結果、同センター内作業委員会(委員長:松本吉弘)が、1995年度に原案を作成した。PCTE が国際標準になった経過の詳細は、この JIS の解説に詳しく記載されている。

## 5. 実用化へ

PCTE は、実際に利用されている例からみて、次のような目的で実用化がはかられている。

### (1) ソフトウェアプロセスの管理

プロジェクトごとに、PCTE リポジトリを用いて、ソフトウェアプロセスの定義と、ソフトウェアプロセス間の関係を SDS によって記述し、各プロセスの進捗状態などをその属性によって管理することができる。また、プロセスによって作られる成果物(ソフトウェアプロダクト)を、SDS 上でプロセスと関係づけることによって、管理することができる。このような方法で、ISO 9000 標準に従う品質管理のためのソフトウェアプロセス支援手段の1つとして、PCTE の利用が考えられる。

### (2) 統合型 CASE 環境の実現

複数のツールを、ECMA/NIST 参照モデルで示されたフレームワークに従って組み合わせ、ユーザの行うエンジニアリングの目的に合致したサービスを提供するように構築されたエンジニアリ

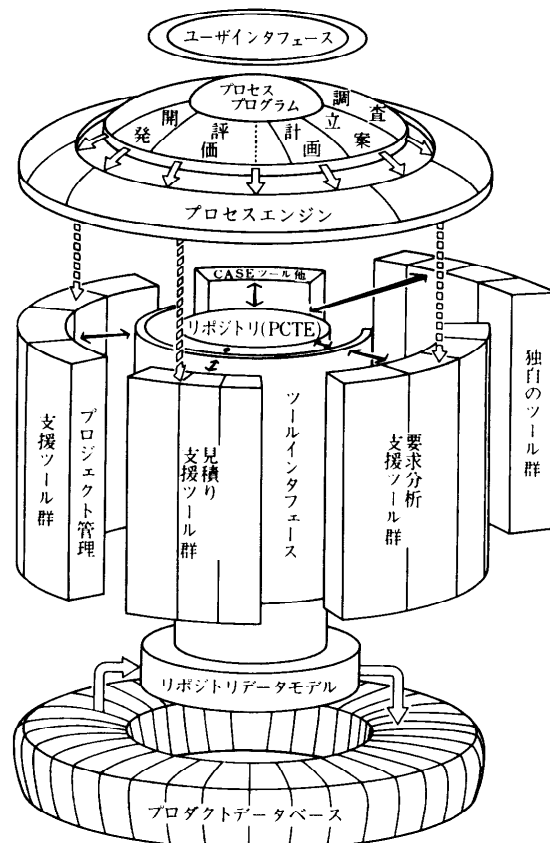


図2 PCTE を中心とする統合型 CASE 環境のイメージ (本図は、情報処理振興事業協会・システムインテグレーション技術開発調査専門委員会(委員長:松本吉弘)で作成した図を、一部修正したものである。)

ング環境を、統合型CASE環境、またはI-CASE (Integrated Computer Aided Software Engineering) 環境という。図-2にPCTEを中心とする統合型CASE環境のイメージを示す。PCTEリポジトリは、オブジェクト管理サービスを提供することによって、ツールが互いにデータを共有することを可能とする。また、PCTEリポジトリから通常のデータベース管理システムに対して、ツールで使用したデータをエクスポートすることができる。必要なツールをUNIX、またはパーソナルコンピュータの開放型分散環境上で組み合わせて実行することを可能にし、マルチユーザ、マルチプロセスに対する並行制御、版および構成管理、セキュリティ維持もあわせてサービスする。

#### (3) 企業内部門統合情報システムを支援

大きな企業のなかでは、部門ごとに異なるデータモデルによって管理されているデータ集合があり、互いにほかのデータモデルのなかに存在するデータを共有できないでいる状況がしばしば観察される。分散環境のなかで、異なるデータモデルによって管理されているデータを、共有することができれば、企業全体としてデータの多重管理による不都合を軽減することができる。分散環境に、統合されたデータモデルをもつPCTEリポジトリを組み込むことによって、データ管理方法の透明化を実現することができる。

#### (4) 情報ウェブの実現を支援

ネットワーク上に分散して管理されているメディア(文書、イメージ、音声信号、動画コード、など)が互いに関連をもつシステムにおいて、1つのメディアに埋め込まれたアンカ(anchor)から、ほかのメディアの指定箇所に至るリンクを設定し、全体を1つのハイパメディアとして実現する環境を形成する場合がある。このような場合に、PCTEは、メディア、アンカ、リンクなどをSDSによって定義し、ハイパメディアとして全体を作動するために必要な制御を提供することができる。

## 6. む す び

PCTEがISO標準として制定される過程において、次にあげる3つの問題が懸案として残された。

- (1) 開放型システム環境(open system environment)への対応
- (2) オブジェクトとして記述されている対象への対応
- (3) ファイン・グレイン(fine grain)対象の扱い

第1、第2の問題に関しては、OMG(Object Management Group)が標準化を進めているCORBA(Common Object Request Broker Architecture)との整合をはかり、CORBAが提示しているIDL(Interface Description Language)へのバインディング(IDLによる記述のなかでPCTEのメタ表現が使えるようにすること)がすでにECMA-230として制定されている。また、CORBA環境のような分散オブジェクト環境で稼働することを意識して作られたツールをPCTE環境に適合して使えるようにするためのオブジェクト指向拡張がECMA標準制定に向けて作業されている。

第3の問題は、上の2つに比べて、性質が異なる内容を扱っている。ツール間でより緊密な連携を行おうとするとファイル単位というようなコース・グレイン(coarse grain)ではなく、ファイルに記述されている細かいエレメント単位のファイングレインでツールが扱う情報をモデリングする必要がある。この場合PCTEが提供するきめ細かいロック制御や各オブジェクトにシステムで定義している属性の管理などが性能上の問題点となる。このような性能上の問題点を回避するためクラスタというキャッシュに類似した概念を導入したファイングレイン仕様が、ECMA-227(抽象仕様)、228(Cバインディング)、229(Adaバインディング)として制定されている。

これらの拡張仕様のISO化の議論がISO/IEC JTC 1/SC 22/WG 22で現在進められている。

(平成8年5月30日受付)

**松本 吉弘 (正会員)**

1932年生。1954年東京大学電気工学科卒業。同年(株)東芝に入社。1974年東京大学より工学博士を受領。1985年東芝・理事, 1989年定年扱い退職後, 京都大学工学部および同大学院工学研究科(情報工学専攻)教授に転出, 定年退官とともに1995年より大阪工業大学に転じ, 情報科学部教授。この間, 1994年より(財)京都高度技術研究所所長および1996年より武蔵工業大学講師を兼任, 現在に至る。工業用計算機制御システムの開発, ソフトウェア生産技術の開発と実用化により日本電機工業会進歩賞, 発明協会全国発明表彰発明賞, 科学技術庁研究功績者表彰, IEEEフェロー表彰を受賞。「ソフトウェア工学」(丸善)など著書18点。電気学会終身員, IEEEフェロー, 日本ソフトウェア科学会各会員。

**吉野 松樹 (正会員)**

1958年生。1982年東京大学理学部数学科卒業。同年(株)日立製作所入社。1987-1988年, 米国コロンビア大学大学院修士課程, MSCS取得。現在, ソフトウェア開発本部 AI 設計部主任技師。言語処理系, ソフトウェア開発支援環境の開発に従事。情報規格調査会 SC 22 専門委員会オブザーバ(PCTE 担当)。