

システムの構築を容易にする オブジェクト指向開発環境

Object Oriented System Development Environment

広瀬 正*

Tadashi Hirose

葉木 洋一***

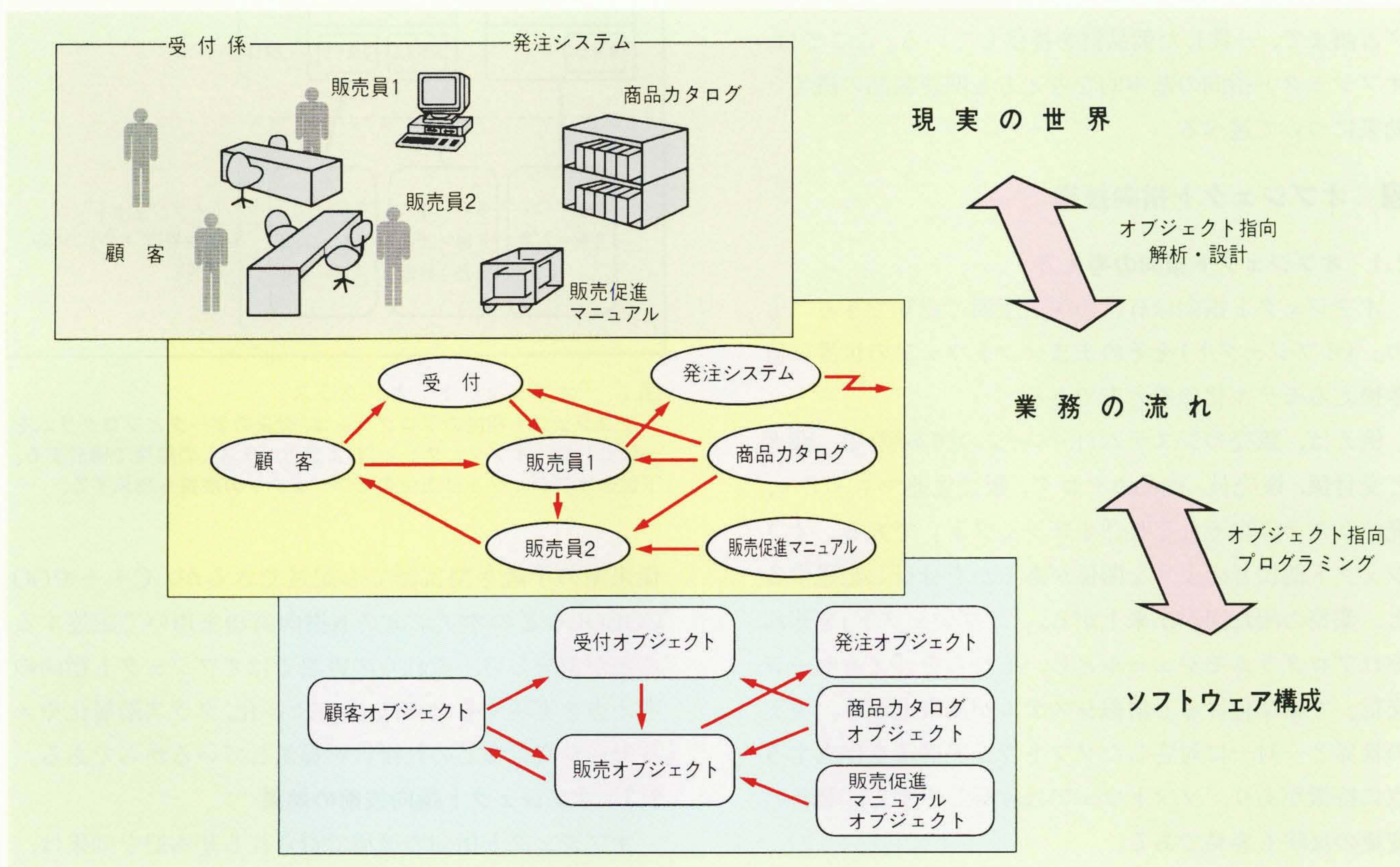
Yōichi Hagi

増位 庄一**

Shōichi Masui

山中 止志郎****

Toshirō Yamanaka



オブジェクト指向による情報システムの構築

「実世界に即したモデル化」がオブジェクト指向の考え方である。業務内容の解析からソフトウェア構成まで一貫した対応関係を保つことで生産性・保守性が向上する。

最近の激しい環境変化に柔軟に対応できる企業情報システムを構築していくうえで、オブジェクト指向の考え方が望まれている。

オブジェクト指向の考え方では、システム化の対象となる実世界に存在する「もの」を「オブジェクト」と呼ぶプログラムの基本単位として表現する。オブジェクトは相互に独立した形で作成されるため、システムの変更は関連するオブジェクトの追加・変更だけで済み、システムの変更に対する柔軟性の

飛躍的向上が期待できる。

日立製作所は、高信頼性を要求される基幹システム向けのオブジェクト指向開発環境と、エンドユーザーと共同で設計・開発するプロトタイピング型システム向けのオブジェクト指向開発環境を提供しており、いずれも、システムの解析・設計からプログラム開発まで一貫してオブジェクト指向技術を適用する製品群を提供しているところに特徴がある。

* 日立製作所 情報事業本部 ** 日立製作所 ソフトウェア開発本部
**** 日立製作所 情報システム事業部

*** 日立製作所 ビジネスシステム開発センタ

1 はじめに

オブジェクト指向は、プログラム開発段階だけでなくシステムの解析・設計段階でも大きな効果が得られる新しいソフトウェア技術である。日立製作所はオブジェクト指向製品として、システム解析・設計工程で使う分析・設計ツールからプログラム開発工程で使うプログラミング言語まで、一貫した製品群を提供している。ここでは、オブジェクト指向の基本的な考え方と関連製品の概要と効果について述べる。

2 オブジェクト指向技術

2.1 オブジェクト指向の考え方

オブジェクト指向は日常の生活空間で認知できる「もの」(オブジェクト)をそのままソフトウェアの世界に置き換えるモデル化の考え方である。

例えば、販売のシステム(15ページの図参照)で、顧客と受付係、販売員、商品カタログ、販売促進マニュアル、発注システムはそれぞれ「オブジェクト」である。オブジェクト間にどのような関係があるかを分析し記述すると、業務の流れ図が出来上がる。「オブジェクト」をそれぞれプログラムモジュールとし、それらを「メッセージ交信」で結び付けると情報システムが出来上がる。現実の世界と一対一に対応したソフトウェア構造を作成する点に特徴があり、ソフトウェアは理解しやすく、業務の変更の反映も容易である。

2.2 設計プロセス

まず、対象世界を解析し、システム化の目的にとって重要な概念をとらえる。ここでは、「いかに処理するか」でなく「何を処理するか」を考え、対象世界のビジネス用語で「オブジェクト」や「クラス」などを記述する(図1参照)。

次に、問題解決戦略や性能確保の方法などシステム構造の全般を考慮しながら、解析で見いだした対象システムの構成要素である「オブジェクト」や「クラス」を整理し、グループ化、階層化する。これらの解析・設計の手法としてRumbaughが提唱したOMT(Object Modeling Technique)がある。OMTはオブジェクト図などを用いて、システム構成を与えられた記述方法でビジュアルに表現しながら設計を進める設計手法で、適用可能範囲の広さと記法の理解しやすさから現在最も広く使われている。

オブジェクト指向で設計されたソフトウェア構成は、

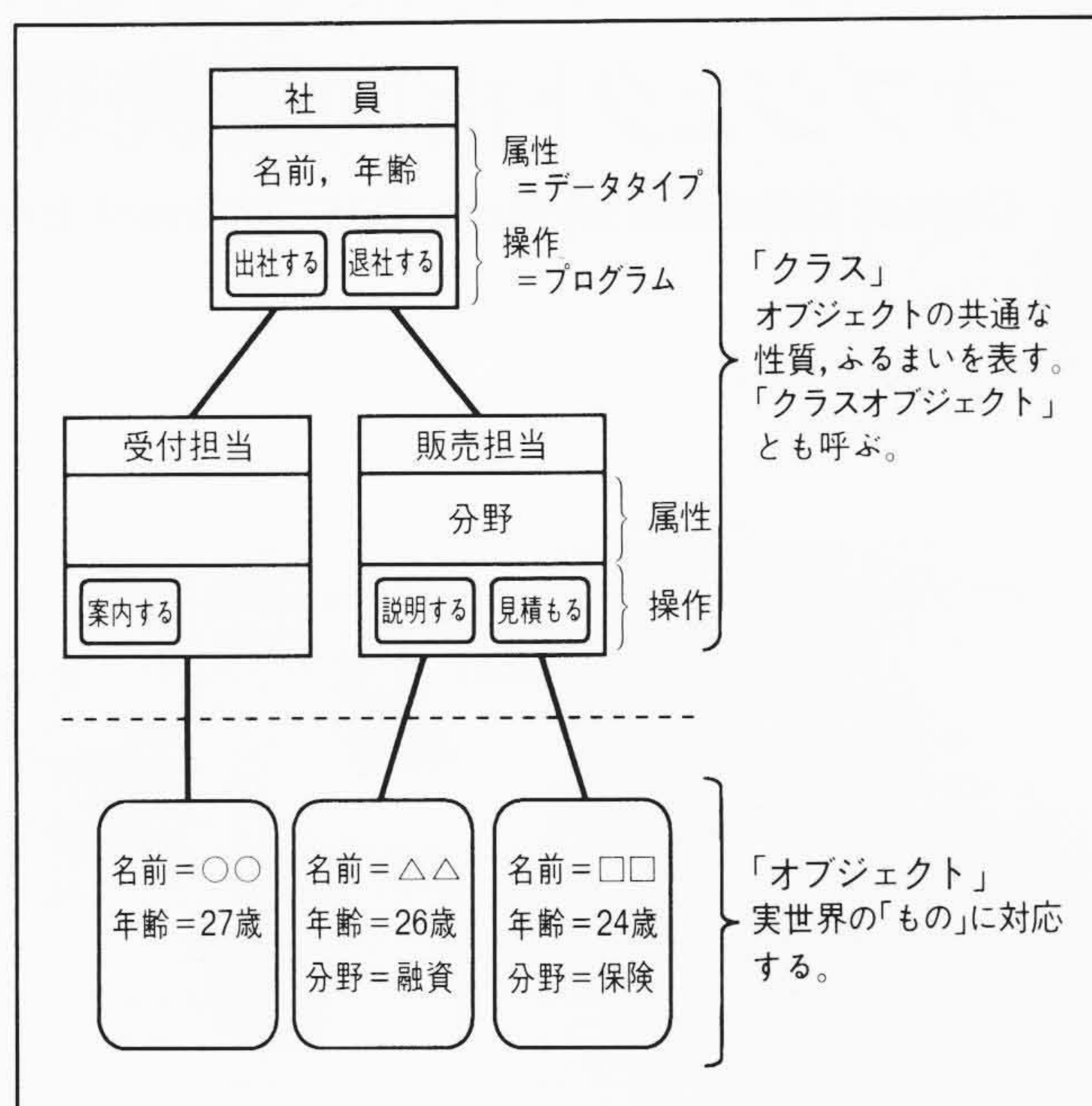


図1 「オブジェクト」と「クラス」

オブジェクト指向のプログラムは、従来のデータとプログラムを一体とした「オブジェクト」、および「クラス」の階層で構成する。下位のオブジェクトは上位のオブジェクトの性質を継承する。

従来型の手続き型言語でも記述できるが、C++やOO COBOLなどのオブジェクト指向言語を用いて記述することが望ましい。これらの言語ではオブジェクト指向の考え方を実現するために、カプセル化、クラス階層化やメッセージ交信などの仕掛けを提供しているからである。

2.3 オブジェクト指向技術の効果

オブジェクト指向の適用で得られる基本的な効果は、(1) 理解のしやすさ、(2) 設計品質の向上、(3) 再利用性の確保である。

「理解のしやすさ」は解析、設計、開発という三つの開発フェーズで一貫して「オブジェクト指向」を用いることから得られる効果である。仕様とそれを実現するプログラムの対応が明快で、システムの変更に対応しやすいシステムが構築できる。

「設計品質の向上」は、実世界に対応させたシステム構成を設計することから導かれる。システム設計者は、いかにシステムを作成するかという以前に、目標とするシステムは何をするシステムかを熱心に検討せざるをえなくなり、これが洗練されたソフト構造を導く。

三番目の「再利用性」は上述の効果の結果として得られる。特に、オブジェクト指向言語のカプセル化機能は規定されたアクセス以外を許さないため、プログラムの独立性を高めて再利用しやすくする。

3 オブジェクト指向開発環境

3.1 ソフトウェア開発形態の多様化

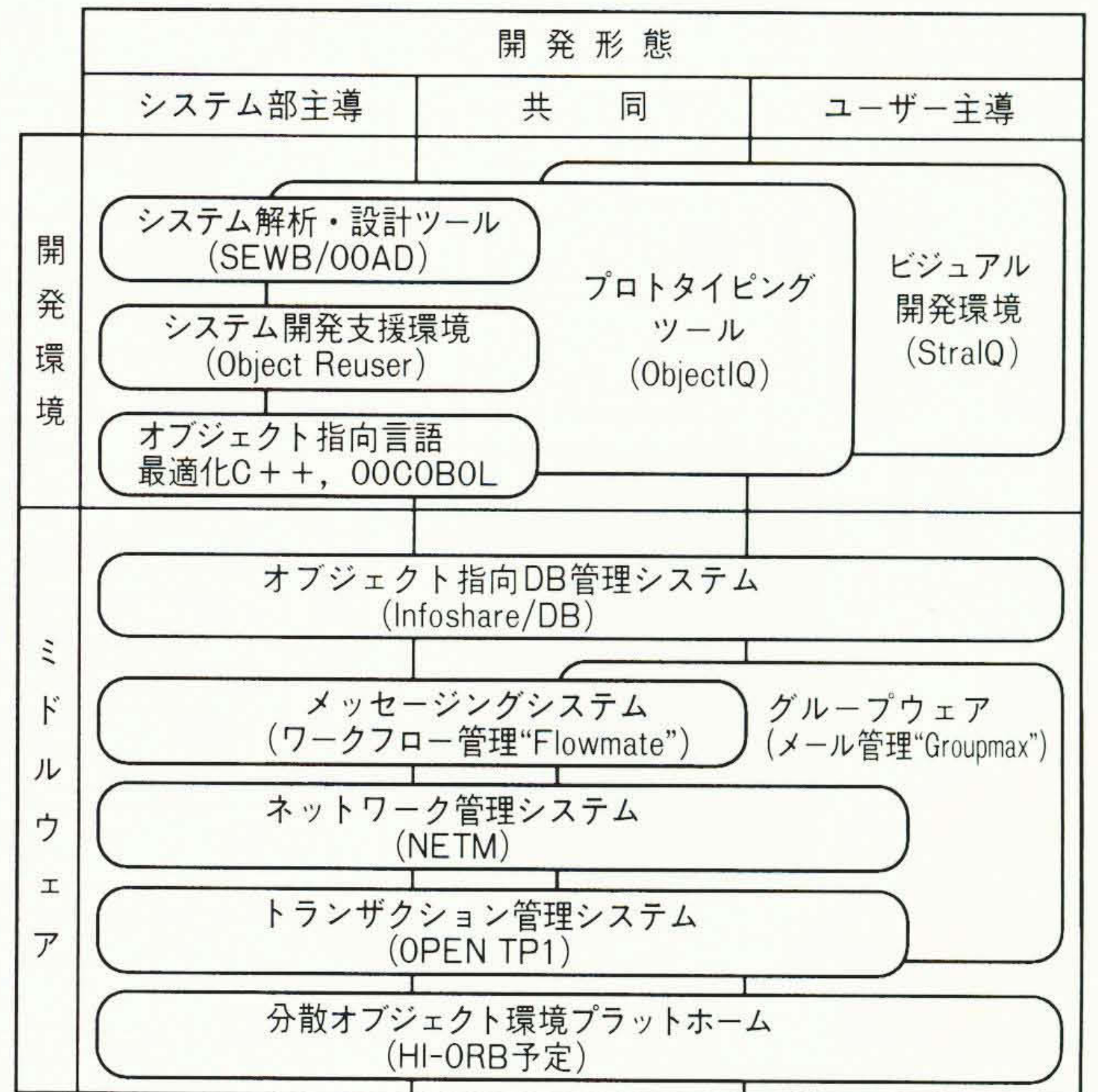
計算機環境が整うにしたがって、ソフトウェア開発形態も多様化している。要求される信頼性の高い基幹システムは、従来どおりシステム部主導の開発を行わなければならない。一方、処理内容の変更頻度が高く、エンドユーザーにある程度のシステム構築スキルの習得を期待できる場合は、エンドユーザーとの共同開発を行う(図2参照)。

3.2 オブジェクト指向解析・設計支援ツール (SEWB/OOAD)

SEWB/OOADはOMTに準拠したオブジェクト図の作成を支援するオブジェクトエディタと、オブジェクト図を解析してC++言語のクラス定義部分を自動生成するコードジェネレータから構成される(図3参照)。オブジェクトエディタは強力な構文チェック機能を内蔵している。これにより、OMTの初心者でもその記述法を学びながら、しかも効率よくシステム設計ができる。

3.3 C++言語処理系(最適化C++)

最適化C++言語処理系群は、強力な最適化機能によって実行性能の良いプログラムコードを生成するだけでなく、プログラミングおよびデバッグの作業を支援する周辺ツールを持つ点に特徴がある。その一つがブラウザ(SEWB/C++BROWSER)である。C++プログラムを



注：略語説明 DB (Database)

図2 オブジェクト指向ソフトウェア製品

システムの分散化とともに開発形態が多様化している。開発形態に応じて適切なツール、ミドルウェアを選択する必要がある。

解析して、そのプログラム構造をビジュアルに表示する。これにより、複雑になりがちなクラス間の関係の正しい理解が促される。

3.4 オブジェクト指向COBOL言語処理系(OOCOBOL)

OOCOBOLはCOBOL言語のもとに蓄積された知識や

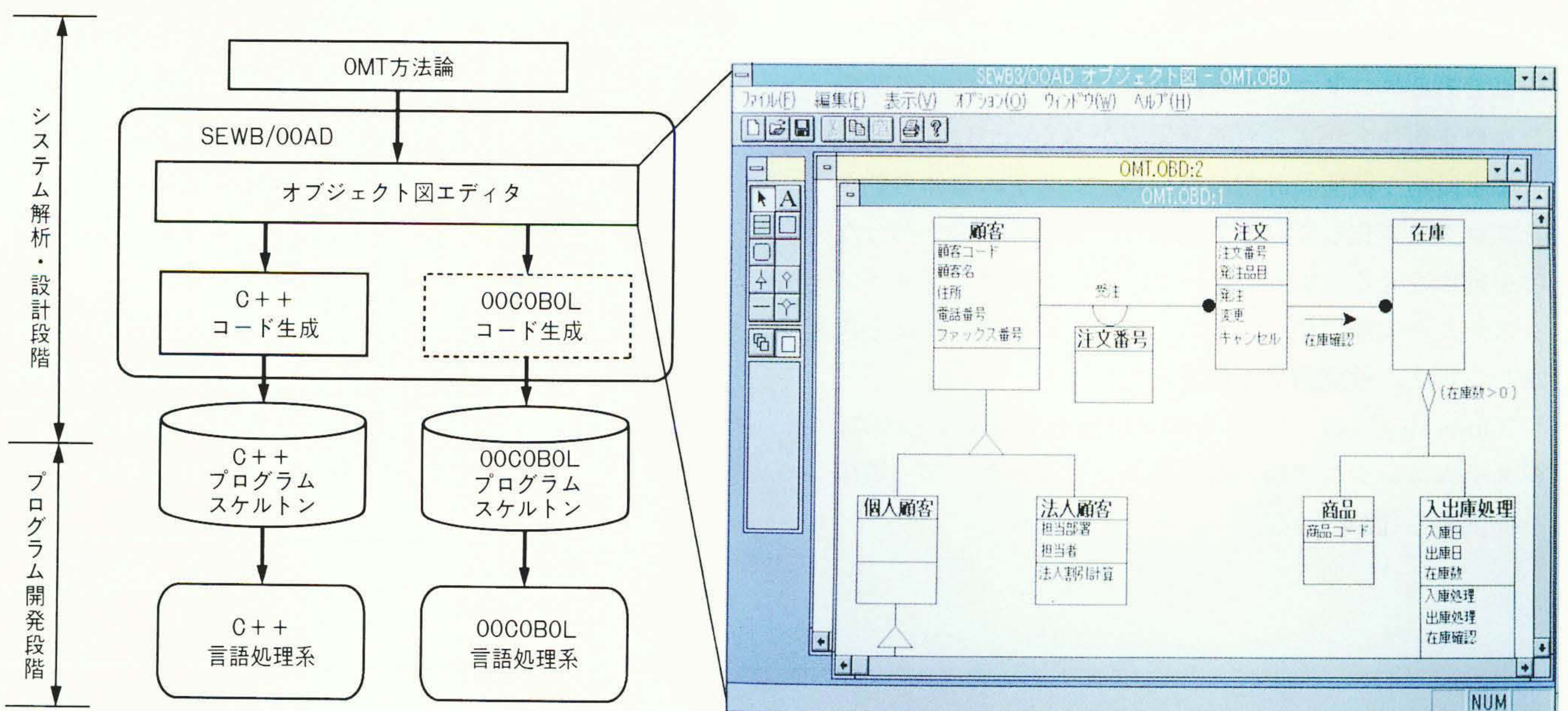
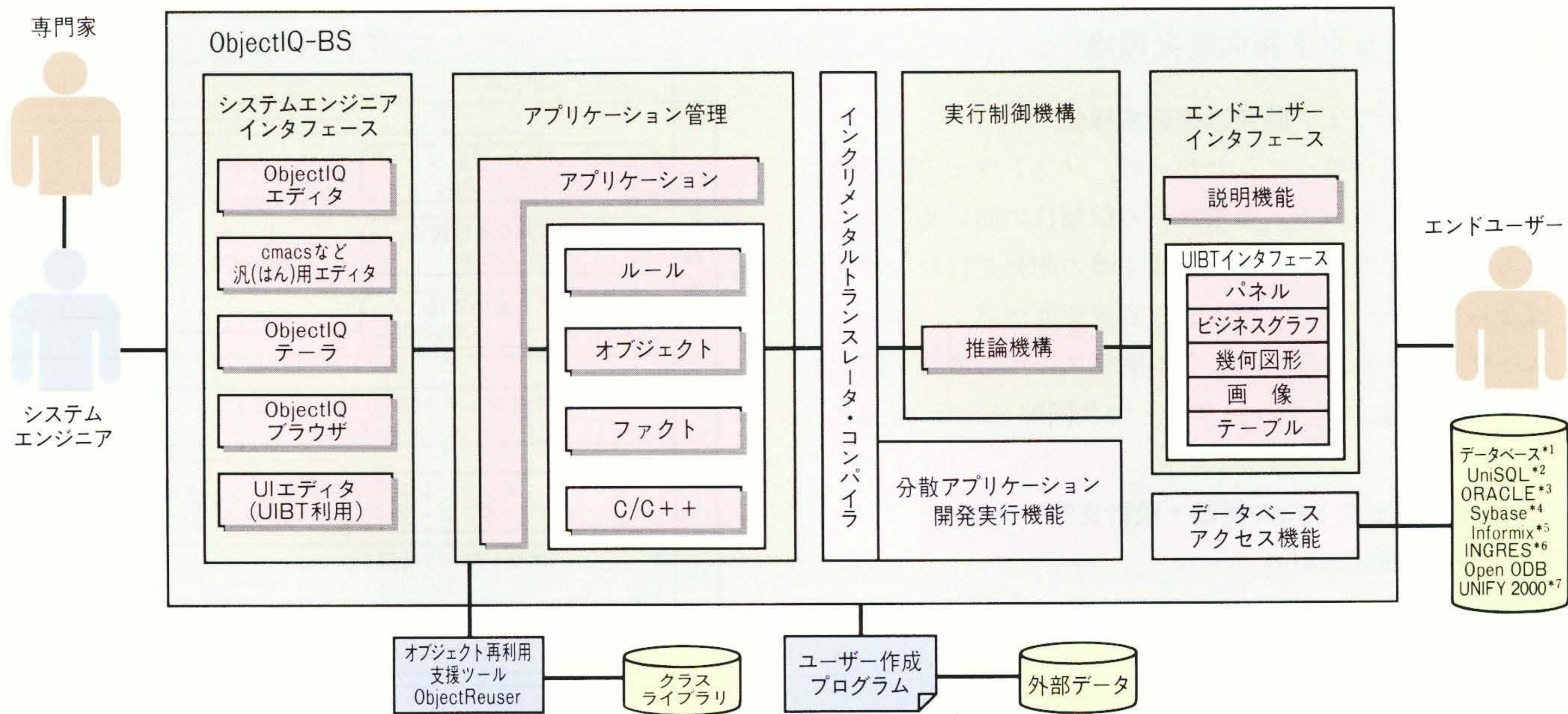


図3 オブジェクト指向解析・設計支援ツール

オブジェクト図エディタで作成したソフトウェア構成は、そのままC++プログラムの定義部分に自動変換される。これにより、設計からプログラムまでの一貫性が保持できる。



注：*1 機種によってサポートされるデータベースは異なる。 *2 計画中。 *3 ORACLEは、米国Oracle Corporationの登録商標である。
*4 Sybaseは、米国Sybase, Inc.の商品名称である。 *5 Informixは、米国Informix Software, Inc.の商品名称である。
*6 INGRESは、米国ASK Group Inc.の商品名称である。 *7 UNIFY2000は、米国Unify Corp.の商品名称である。

図4 オブジェクト指向統合アプリケーション開発環境の構成
システム部の専門家とエンドユーザーが協力してシステムを構築するためのオブジェクト指向設計・開発環境を示す。

ソフト資産をそのまま有効に活用しながら、生産性の高いオブジェクト指向プログラミングを実現する言語である。C++同様、開発作業を支援する周辺ツールとしてブラウザ、デバッガを装備している。

3.5 オブジェクト指向統合アプリケーション開発環境 “ObjectIQ”

システム開発の専門家であるシステム部と、業務の専門家であるエンドユーザーが協力しながらシステム開発を進める場合を想定した開発環境が“ObjectIQ”である。
まず概略な機能設計だけを行い、システム全体の動きをプログラム化して動作させたり、システムの一部だけを動作させる。エンドユーザーは、このプロトタイプシステムを実際に使いながらシステムに必要な機能を探求し、再度、機能設計を修正・追加する。

“ObjectIQ”はこのような開発形態を支援するために、システムエンジニア用の環境とエンドユーザー用の環境を提供する(図4参照)。

4 おわりに

オブジェクト指向技術の適用はさらに拡大し、開発環境だけでなく実行環境にも改革をもたらそうとしている。オブジェクト指向により、ソフトウェアのインタフェースを単純に接続できる形に整理することができる。これにより、電気器具がコンセントに差し込むだけで動作するように、ソフトウェア部品をORB(Object Request Broker)と呼ぶミドルウェアに単純につなぎこむだけで、所望のシステムが構築できるようになる。ソフトウェア部品はネットワーク結合された範囲であれば、どの計算機上に置いてもよい構造になっている。

日立製作所は、ORBの業界標準仕様CORBA(Common Object Request Broker Architecture)^{※)}仕様に準拠すると同時に、生産性、柔軟性向上を目指したオブジェクト指向開発、実行環境の実現に努力していく。

※) CORBAは、Object Management Groupの商標である。

参考文献

1) J. Rumbaugh : Object-Oriented Modeling and Design, Prentice Hall(1991)