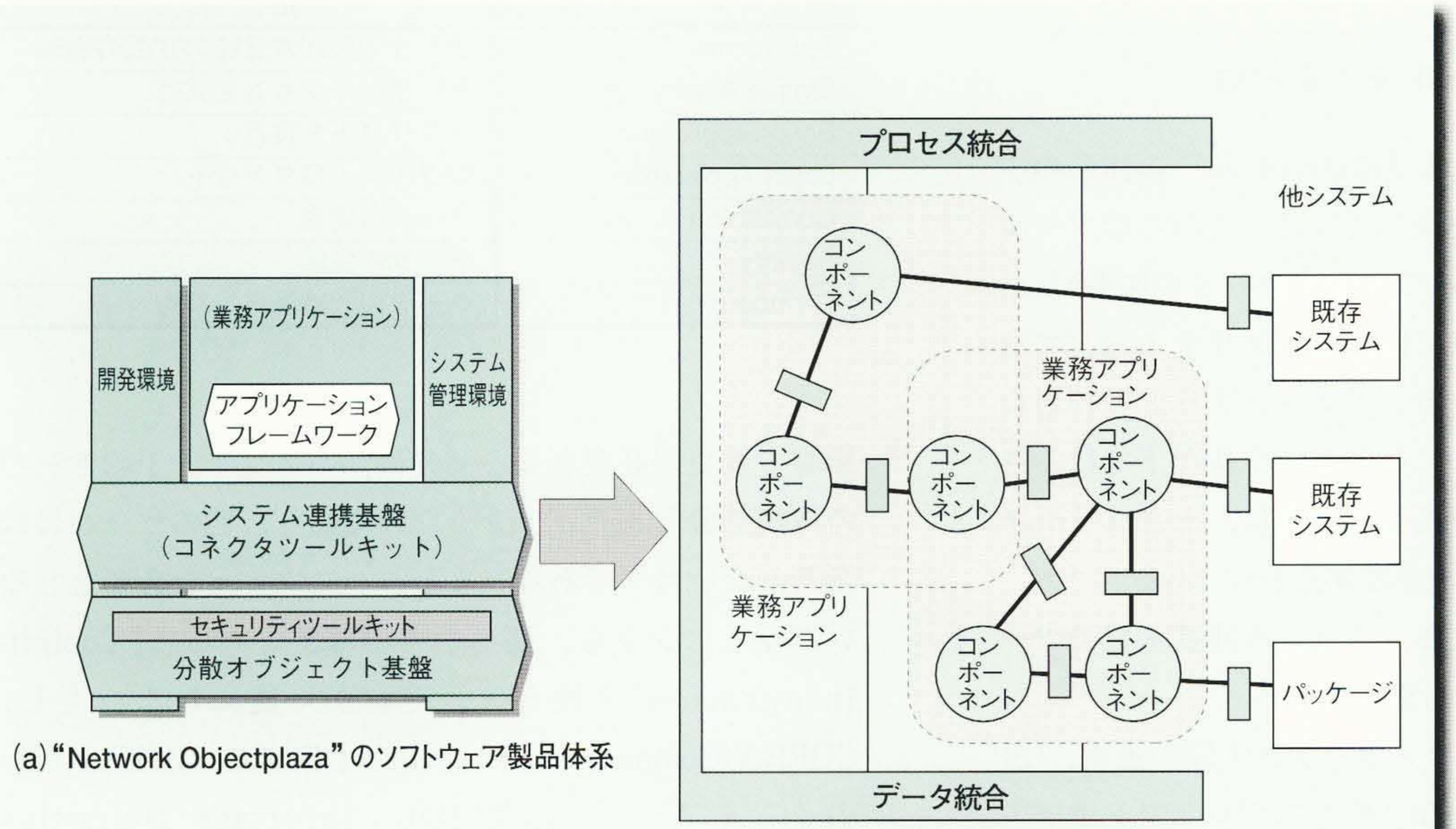
特集　ネットワーク時代の新情報システムソリューション：エンタープライズ アプリケーション インテグレーション

アプリケーションインテグレーションのためのツールキットの活用技術

Methods for Utilizing Toolkits to Provide Business Solutions Based on EAI

奥川淳一　*Jun'ichi Okukawa*　天野雅志　*Masashi Amano*
首藤卓馬　*Takuma Sudô*　新家博文　*Hirofumi Shinke*



(a)“Network Objectplaza”のソフトウェア製品体系

(b) 情報システムの構成

注1：コンポーネントとは，業務機能を実現する構成要素である。
注2：▯（コネクタツールキット）

アプリケーション インテグレーションのためのツールキットの利用例

(a) に表すコネクタツールキットを使ってコンポーネントを連携することにより，(b) のような情報システムが構築できる。

アプリケーションのインテグレーション(統合)によって実現するソリューションで，実際の作業で高効率を実現するのは「ツールキット」である。ツールキットは複数のカテゴリーに分類できるが，アプリケーションインテグレーションという観点では，種々のシステムを連携させる，システム連携基盤のツール(コネクタツールキット)が特に重要である。

日立製作所は，コネクタツールキットをソフトウェア製品として提案している。その製品の一つ“Toolkit-Integration”は，業務機能を実現する要素(コンポーネント)間の連携を助けるツールである。この製品により，多種多様な異種システム間の連携を伴うアプリケーションインテグレーションで，複雑さを軽減し，システムの相互運用性と変更の容易性を高めることができる。また，もう一つの製品“Object Wrapper”は，既存システムとコンポーネントとの間を連携するツールキットである。既存システムをオブジェクト化することにより，カプセル化のメリットなどを享受できるようにするとともに，複数アプリケーションの機能の統合や画面のGUI(Graphical User Interface)化のためにも有用である。ツールキットを使う際の手順やノウハウは「技法」という形でまとめている。

1　はじめに

使い慣れないツールを使うのは難しく，むしろ，慣れるまでのオーバヘッド(本来の目的外の負荷)のほうが大きいこともある。そこで，ツールを使う際の手順やノウハウを「技法」という形で開発方法論の一部としてまとめ，よりよいソリューションの提供が可能なようにすることが求められている。

日立製作所は，EAI(Enterprise Application Integration)を支援する新しい開発方法論を提案している(詳細はこの特集の「インテグレーション指向のシステム開発方法論」を参照)。この新開発方法論では，個々の製品(ツール)や開発言語などに依存しないように，汎用的な記述を採用している。一方，「技法」は，むしろ，個々の製品や技術によって異なる点を明確に示していこうとするものである。技法には，(1) 市販のパッケージソフトウェアごとのインテグレーション技法，(2) 製品ごとではなく，個別の技術観点ごとに知識を集約した技法(プロジェクト管理技法など)，および(3) 日立製作所が提案するインテグレーションツールごとの技法がある。

技法では，適用事例で得たノウハウなどを基にして，(1) 適用例，(2) メリット・デメリットを判断するガイド，(3) ツール適用の手順，(4) ノウハウ，(5) 作業工数，性能，信頼性などの評価，(6) 前提知識としての技術解説，

機能解説などについて記述する。

ここでは，EAIを支えるツールキットの活用技術および「技法」の一端について述べる。

2 EAIのためのツールキットとは

日立製作所は，“Network Objectplaza”の体系の中で，さまざまなツールを提案している。それらのツールは，五つのカテゴリーに分類できる(49ページの図参照)。このうち，EAIという観点では，セキュリティ(安全性)のツールキットが重要である。インターネットの利用など情報システムがオープンになっていく状況下では，さまざまな異種システムが連携したときに，セキュリティの確保には今まで以上に注意を要するからである。一方，種々のシステムを連携させる，システム連携基盤のツール(コネクタツールキット)も重要である。

システム連携には，(1) コンポーネント間の連携，(2) 既存システムやパッケージなど他システムとコンポーネントとの連携がある。コネクタツールキットも，それに対応して幾つか存在する。

以下で，“Network Objectplaza”のツールのうち，コネクタツールキットである“Toolkit-Integration”と“Object Wrapper”を取り上げ，活用技術の一部を紹介する。

3 “Toolkit-Integration”

“Toolkit-Integration”は，コンポーネント間の連携を助けるツールキットである。多種多様な異種システム間の連携を伴うアプリケーションインテグレーションで，複雑さを軽減し，システムの相互運用性と変更の容易性を高めることができる。これは，**表1**に示すように，インタフェースを限定して共通化するためである。

3.1 適用例

一般的な旅行代理店の窓口システムを考える。外部の各リソース(交通機関，宿泊施設など)のシステムと情報の交換を行い，目的に合った旅行計画を作成しているものとする。

ここで，外部システム接続などに変更が生じた場合でも，“Toolkit-Integration”は，他の外部システムにこの接続変更の影響を生じさせない。それが「変更容易性」である。

例えば，現在数社の外部システムを連携しているとする。ここに，バスツアー会社の予約サーバが機能追加され，インタフェースが変更になった場合を考える。この影響で変更が必要となる部分は，バスツアー予約サーバの接続部分とクライアントだけであり，他のサーバには影響がないはずである。また，バスツアーを必要としないクライアントも，変更不要のはずである。“Toolkit-Integration”を使えば，それが可能になる。もし，CORBA(Common Object Request Broker Architecture)のインタフェース言語(IDL：Interface Definition Language)だけで実現しようとすれば，すべての構成要素のインタフェースの定義をやり直さなければならない。

表1 “Toolkit-Integration”で提供するインタフェース
アプリケーションどうしのインタフェースは7種類に限定されている。

インタフェース名	機　能
Application	データの形式や意味の定義の通知
Simple Application	アプリケーションを実行
Script Application	スクリプトを実行
Login Application	ログイン・ログアウト
Conversion	データの変換
Factory	データの生成
Clipboard	データの一時保存・共用

メリットはこれだけではなく，インタフェースが共通化されるため，開発作業量が削減できることなどが考えられる。デメリットはメリットの裏返しとなるが，例えばある一つの接続だけ見ると，共通化はコスト増となることもある。したがって，メリットを最大に享受するためには，システムのどこに“Toolkit-Integration”を適用するかを考える必要がある。

3.2 適用方法

まず，どこのインタフェースに適用するかを検討する。変更が頻繁に発生する個所など，ツールの特徴を発揮できる観点で見ていくようにする。このとき，コンポーネント間で同じ意味のデータの受け渡しが可能かどうかなど，他のアプリケーションとの連携についての観点を忘れないようにすることも大切である。単純な変換では，連携ができない場合があるからである。例えば，毎月20日締め切りの給与データと毎月25日締め切りの給与データでは，単純な変換だけではやり取りができないはずである。

次に，インタフェースの種類(連携方法)を決定し，コンポーネント間でやり取りするデータの種類(データアイテム)を決定する。

最後に，コンポーネントのオペレーションを設計し，各インタフェースごとの規約に従ったコードを実装する。

4 “Object Wrapper”

“Object Wrapper”は，既存システムとコンポーネントとの間を連携するツールキットである。これにより，分散オブジェクト技術で開発したプログラムと既存システムとの間で通信が可能となる。これを，「既存システムをラッピングする」と言う。

連携する既存システムの形態や環境に合わせて，適切なラッピング方法とラッピングミドルウェアを選択する必要がある。

4.1 適 用 例

統合メニューからさまざまな既存システムを利用することができるシステム例を**図1**に示す。この例のように，ユーザーは，既存システムがどの環境(ホスト)で動作しているのかは意識しなくてもよい。また，ユーザーインタフェースをGUI(Graphical User Interface)化することができる。

既存システム間の機能差のためにうまくつながっていない部分をラッピングプログラムで統合する例を**図2**に示す。工程フローを定義するシステムと生産管理(計画・実績)システムが別になっているためにシステム間のつなぎが悪かったのを，ラッピングプログラム内の処理により，検索結果を使って他システムへアクセスするようにした例である。

4.2 適用手順

適用手順を**図3**に示す。この手順は，既存オンラインシステムのラッピングの事例である。

作業手順上の第一のポイントは，ラッピングをどこに，どのように適用するかを検討することである。

構築作業に入ってからのポイントは，まず，既存システムの解析である。既存システムの画面(オンラインプログラムの場合)や入出力，動作について解析する。正常時の動作だけではなく，エラー時の動作についても漏れなく把握することが必要となる。ラッピングには幾つかの方式があるが，方式ごとに必要な情報が異なる点に注意する必要がある。

「内部処理設計」では，解析結果(入出力フロー図など)を利用してラッピングプログラムの動作を設計する。ラッピングプログラムは一つのオブジェクトであるから，メソッド単位に動作することになる。したがって，メソッドをどのように設計するかがポイントとなる。

次に，ラッピングプログラムへの各入出力について，入出力データを設計していく。“Object Wrapper”を使ったラッピングプログラムへの入出力はツリー形式のデータ構造である。データ構造についての実装者の思い違いによるミスコーディングは，起こりやすく見つけにくいので，注意が必要である。

ラッピングでのテストは，既存システムと新規作成オブジェクトおよびラッピングプログラムを組み合わせて行う「組合せテスト」が主となる。

ラッピング技術を使ったシステム構築の豊富な経験者は，現時点では少ない。しかし，初めてラッピングを行うという場合でも，実装のサンプルを手本にすることにより，比較的簡単に適用できるようになる。

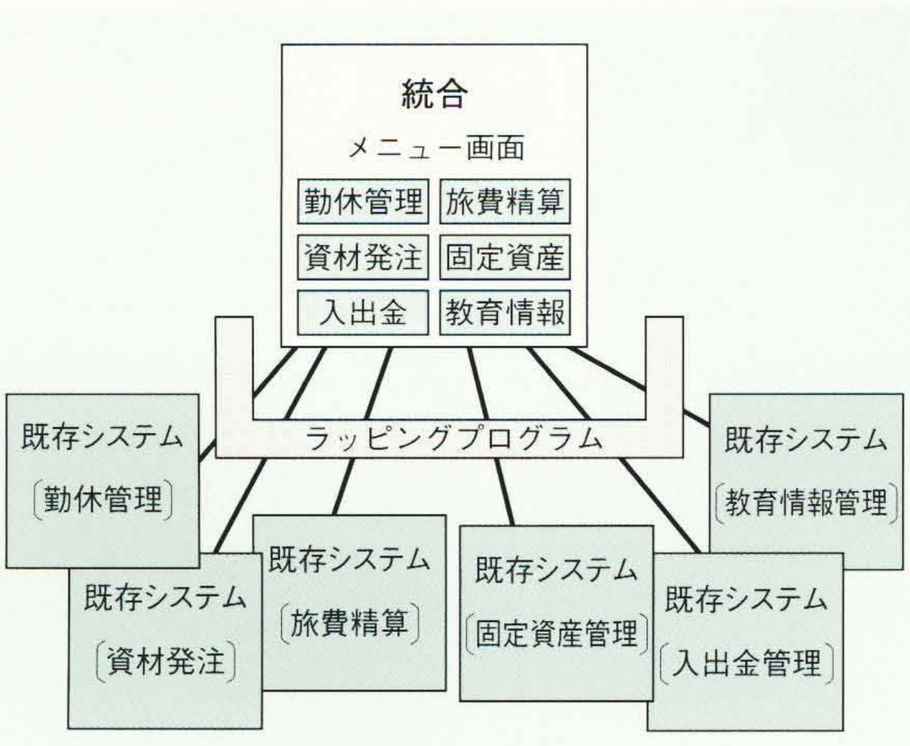


図1 “Object Wrapper”の適用例(1)
別々に開発されたさまざまなシステムを，一つの統合メニューから利用する例である。

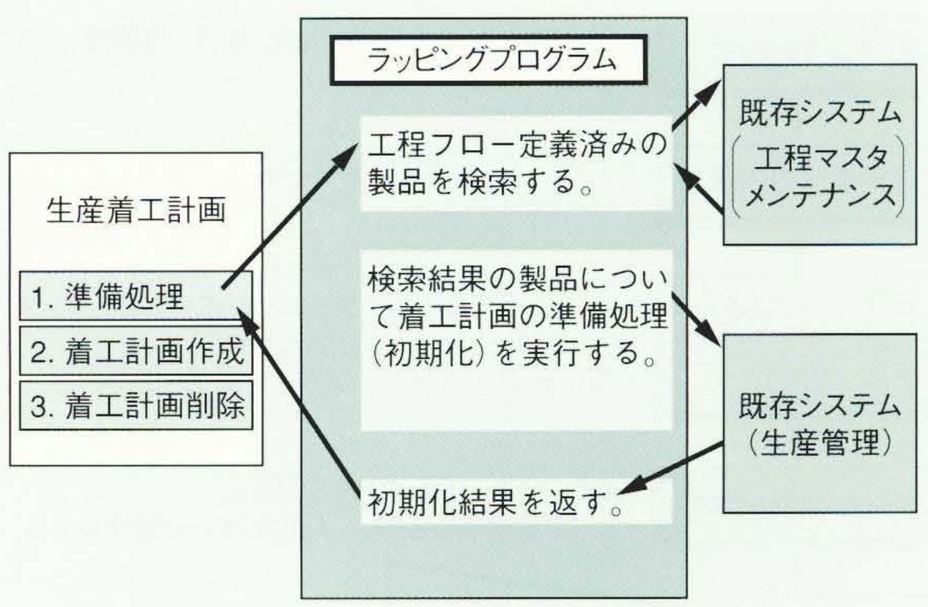
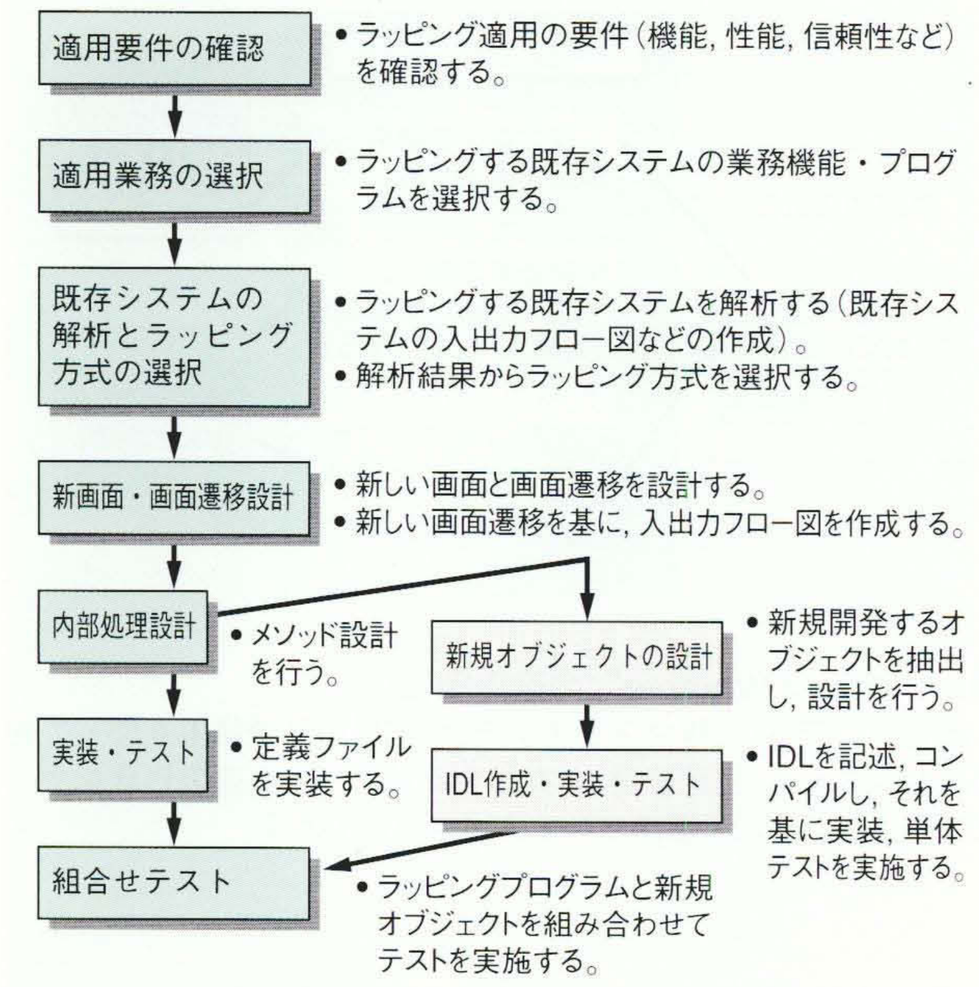


図2 “Object Wrapper”の適用例(2)
既存システム(工程マスタメンテナンス)の処理結果を使って他の既存システム(生産管理)を実行するという，一連の処理をラッピングプログラムで連携させて実現する例である。



注：略語説明　IDL(Interface Definition Language)

図3　ラッピング手順

ラッピングを行うための作業手順と作業内容の流れを示す。「技法」では，各ステップがサブステップに分けて詳細に記述されている。

5　おわりに

ここでは，EAIを支えるツールキットの活用技術について述べた。

市場にはさまざまなツールが提案されている。それらをうまく使うことにより，大きな効果を上げることができる。日立製作所は，それを可能にする技術を，手順や技法という形で整備しつつある。適切な個所に，適切なツールを，上手に適用することによって効果が生じ，よいソリューションが実現できるからである。

ここで述べたツールキットの活用技術では，作業工数や性能，信頼性などの評価について，まだ十分なデータを得ていない。今後は，これらのデータの収集に努め，技法の洗練に役立てていく。また，技法が対象とするツールの範囲の拡張を図っていく考えである。

参考文献

1) 齋藤，外：これからの情報システム“Network Object-plaza”，日立評論，**80**，5，391～396(平10-5)
2) 吉岡，外：従来アプリケーションを分散オブジェクト環境から活用する基幹システム連携技術，日立評論，**80**，5，437～442(平10-5)

執筆者紹介

奥川淳一
1989年日立製作所入社，情報・通信グループ 情報システム事業本部 情報システム事業部 生産技術部 所属
現在，分散オブジェクトシステム連携に関する適用技術の開発に従事
情報処理学会会員
E-mail：okukawa @ system. hitachi. co. jp

首藤卓馬
1989年日立製作所入社，情報・通信グループ ソフトウェア開発本部 第4OS設計部 所属
現在，分散オブジェクトシステム連携基盤の開発に従事
情報処理学会会員
E-mail：sudo @ soft. hitachi. co. jp

天野雅志
1991年日立製作所入社，システム開発研究所 第2部 所属
現在，分散オブジェクトシステム連携基盤に関する開発技術の研究に従事
E-mail：m-amano @ sdl. hitachi. co. jp

新家博文
1994年日立製作所入社，情報・通信グループ 情報システム事業本部 システム開発本部 第二部 所属
現在，分散オブジェクトシステム連携基盤に関する技術開発に従事
工学博士
E-mail：shinke @ bisd. hitachi. co. jp