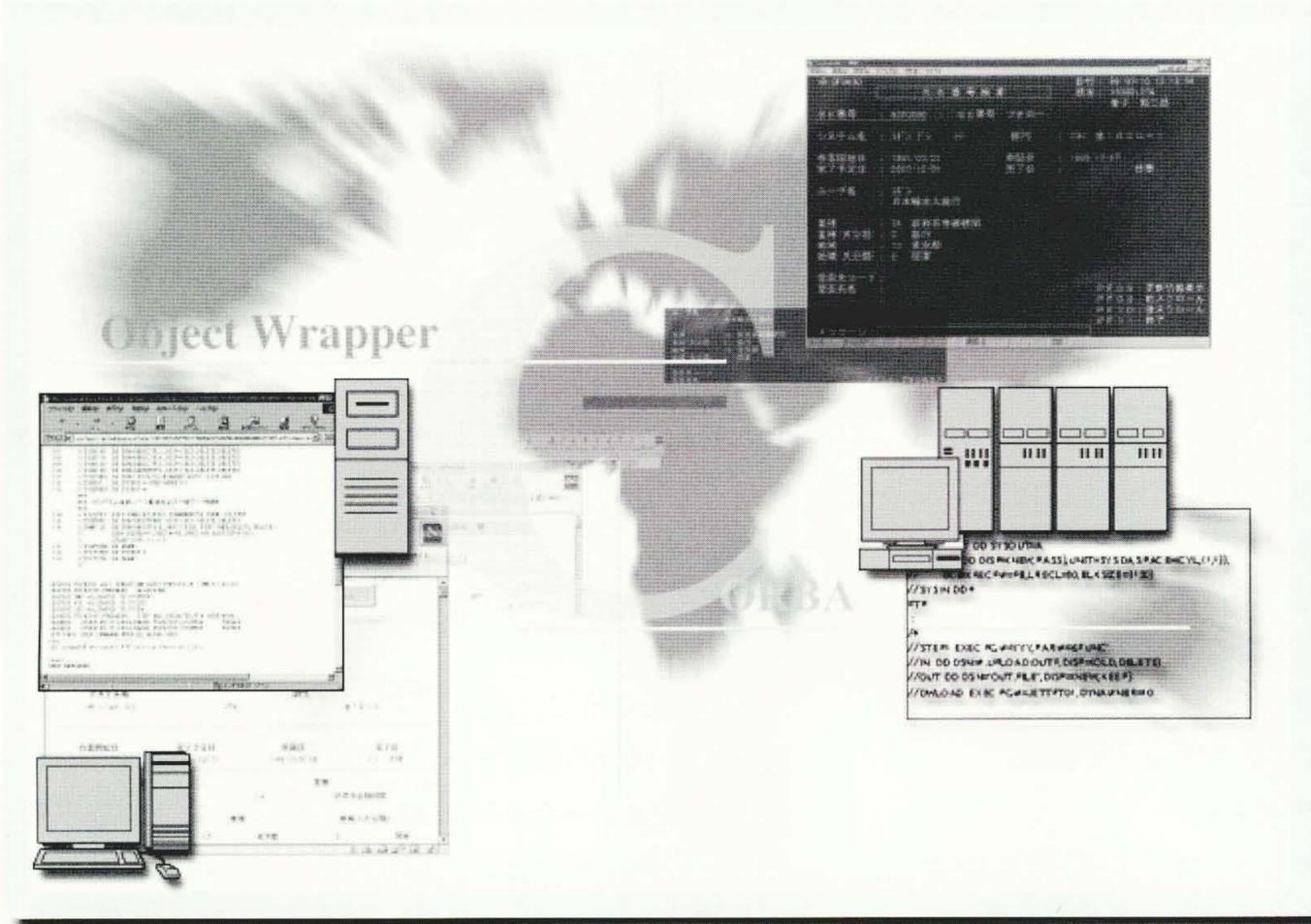


分散オブジェクト技術による アプリケーションインテグレーションの実現とその評価

Realization of Application Integration by Using Distributed
Object Technology and Its Evaluation

渡部史朗 Shirô Watanabe 只野完二 Kanji Tadano
神園幸三 Kôzô Kamizono 田中哲雄 Tetsuo Tanaka



分散オブジェクト技術によって実現するアプリケーションインテグレーション CORBA(Common Object Request Broker Architecture)環境を実現するレガシーラッピングをフィルタとすることにより、パソコン上でメインフレームの環境を容易に利用することができる。

ビジネスの急速な変化に伴って、情報システムも迅速かつ柔軟に構築しなければならない。スピード化の時代にあっては、既存のシステムや流通パッケージなどを部品と見立てて、それらを組み合わせてシステムを構築すること、さらに、将来も、既存のアプリケーションを連携させてシステムを進化させていく仕組みにすることが求められる。これを実現するのが「アプリケーションインテグレーション」と呼ぶシステム構築方法である。

日立製作所は、アプリケーションインテグレーションを実現するために、分散オブジェクトの技術を利用して異質の環境で作られた既存のプログラムを連携する「ラッピング」という技術を開発した。対象となる既存システムの稼動環境に応じてラッピング技法を分類し、おのこの適用を進めている。その一環として、社内の情報管理システムに対してオンラインシステムのラッピングを適用した。また、開発環境の再構築にバッチシステムのラッピングを適用した。これらにより、アプリケーションインテグレーションが、新規システムの分散開発と遠隔地からの既存システムの共同利用の実現に有効であることを確認した。

1 はじめに

近年、既存のシステムや流通パッケージなどを有効に再利用し、それらを組み合わせてシステムを構築することが注目されるようになった。さらに、将来的にも、ア

プリケーションを柔軟に組み合わせてシステムを進化させていく仕組みが強く求められている。これを、「アプリケーションインテグレーションによるシステム構築」と呼ぶ。

しかし、メインフレーム、UNIX^{*1)}やWindows^{*2)}といった異質のプラットフォーム上に、異なる時期に構築した

※1) UNIXは、X/Open Company Limitedがライセンスしている米国ならびに他の国における登録商標である。

※2) Windowsは、米国およびその他の国における米国Microsoft Corp.の登録商標である。

プログラムどうしをメインフレーム上で連携させることは非常に困難である。これを解決させるために開発したのが、分散オブジェクトの技術を使って既存のプログラムを連携させる「レガシーラッピング」という技術である。

ここでは、対象となる既存システムの稼動環境に応じたレガシーラッピング技法、日立製作所の情報管理システムに適用した事例と開発環境の再構築に適用した事例、およびその評価結果について述べる。

2

アプリケーションインテグレーションでの既存システム資産活用

インテグレーション(統合)の対象となる実際のシステム環境には、多くの既存システム(主としてメインフレーム上に構築されたシステム)が存在している。新規に代替システムを構築する手段を取らずに、既存システムと新規システムの連携を図る方法には、多くの有利な点があるが、技術的な課題も多い。既存システム資産を活用するうえで、これらを解決することが、インテグレーションを成功に導くために不可欠な条件となっている。

2.1 既存システム資産活用の有用性

アプリケーションインテグレーションでの既存システム資産活用には、代替システムを新規に構築する場合に比べて、次のような利点がある。

(1) 既存システムの永続性

長年にわたって使用されてきたシステムは、新規に開発するシステムに比べて、実稼動の実績があり、機能面や信頼性が保証されている。

(2) 保有データの保全性

既存システム資産として、蓄積されたデータがある。特にメインフレーム系では、ばく大なデータがサブシステム間に整合性を維持したまま保有されている。新規システムへ移行することは困難であり、できれば避けたいところである。

(3) 開発工数の削減

新規システムを構築するにはばく大な開発工数(経費)を投資する必要があるが、既存システムを活用することにより、開発工数を削減することができる。

2.2 既存システム資産活用での技術的な課題

既存システムをできるだけ修正することなく新規システムと連携させ、既存のシステム資産を活用することができれば、システムインテグレーションで有利である。

しかし、以下にあげるような、既存プラットフォーム

と新規プラットフォーム間の技術的な相違が、インテグレーションの際の障壁となる。

(1) 既存システム資産の理解

既存システムでは、構築されてから時間がたっていて設計者が職場を離れていたたり、ドキュメントが失われていることがある。このため、システムの仕様が把握されておらず、代替システムへの移行が難しい。

(2) インタフェース

メインフレーム上の既存プログラムは、通常、ネットワークインタフェースやAPI(Application Programming Interface)などの、オープンシステムと連携するためのインタフェースを持たない。

これらの技術的な違いに加えて、異なる時期に開発された複数の既存システムが対象になる場合、相手システムごとの作り込み作業が発生し、多くの工数を要する。

3

既存システム資産を有効に活用する技術「レガシーラッピング」

上述したように、既存システム資産へのアクセスには、統一的な手段が必要である。このために日立製作所が開発した「レガシーラッピング」技術について以下に述べる。

3.1 レガシーラッピングの仕組み

レガシーラッピングは、既存システムの機能を、CORBA(Common Object Request Broker Architecture)のオブジェクトの形態で提供するものである。ベンダ非依存の分散オブジェクト技術の標準であるCORBAを用いることにより、CORBA準拠の広範囲のアプリケーションから既存システムへのアクセスが共通の方法で可能になる。

レガシーラッピングは、CORBAオブジェクトの実装に際して、CORBAインタフェースのオペレーションを既存システムのインタフェース(プロトコル)に対応させることによって実現する。

3.2 レガシーラッピング方式の分類

既存システムの資産の中で、ラッピング対象と考えられるリソースは、(1) オンラインプログラム、(2) バッチプログラム、および(3) データベースの三つである。これらのそれぞれに対してラッピング方式を定義することができる(表1参照)。

3.3 レガシーラッピングの実現技術

日立製作所は、上にあげたそれぞれのラッピング方式に対応する“Object Wrapper”製品を用意して、レガシーラッピングをサポートしている(図1参照)。

表1 レガシーラッピング方式の分類

レガシーラッピングの適用では、対象リソースごとに、大別して3種類の方式がある。

方式	リソース	機能
オンラインラッピング	オンラインプログラム	端末画面と入出力項目への抽象的なアクセス手段を提供
バッチラッピング	バッチプログラム	以下の二つの技術要素から成る。 (1) スクリプトラッピング バッチプログラムの非同期的な実行とジョブの状態の監視の抽象化 (2) ファイルラッピング バッチジョブの入出力ファイルを位置透過的、可搬的なオブジェクトとして提供
データベースラッピング	データベース	DBMSのベンダ、プラットフォーム、アーキテクチャ(RDB・階層型)によらない共通インタフェースを提供

注：略語説明 DBMS(Database Management System)
RDB(Relational Database)

(1) オンラインラッピング

Object Wrapper定義ファイルを記述することにより、メインフレーム上のオンラインシステム資産を、クライアント側で利用することができる。定義ファイルは、既存システムの画面・データ項目(トランザクションデータ)とObject Wrapperが提供するメソッド・変数を対応づけている。Object Wrapper製品は、対象のオンライン

システム種別ごとに用意している。

(2) バッチラッピング

“Object Wrapper for VOS3 Batch”により、クライアント側からメインフレーム上のJCL(Job Control Language)の起動、データの取得、ジョブ実行結果などを行うことができる。Object Wrapperを構成するスクリプト、ファイル、レコードオブジェクトに対して、それぞれの定義ファイルを作成する。

(3) データベースラッピング

DBMS非依存のDBアクセスを提供する“DABroker”とそのCORBAインタフェース“DABroker for ORB”により、メインフレーム系のDBにもオープン環境のDBと同じ手続きでアクセスすることができる。

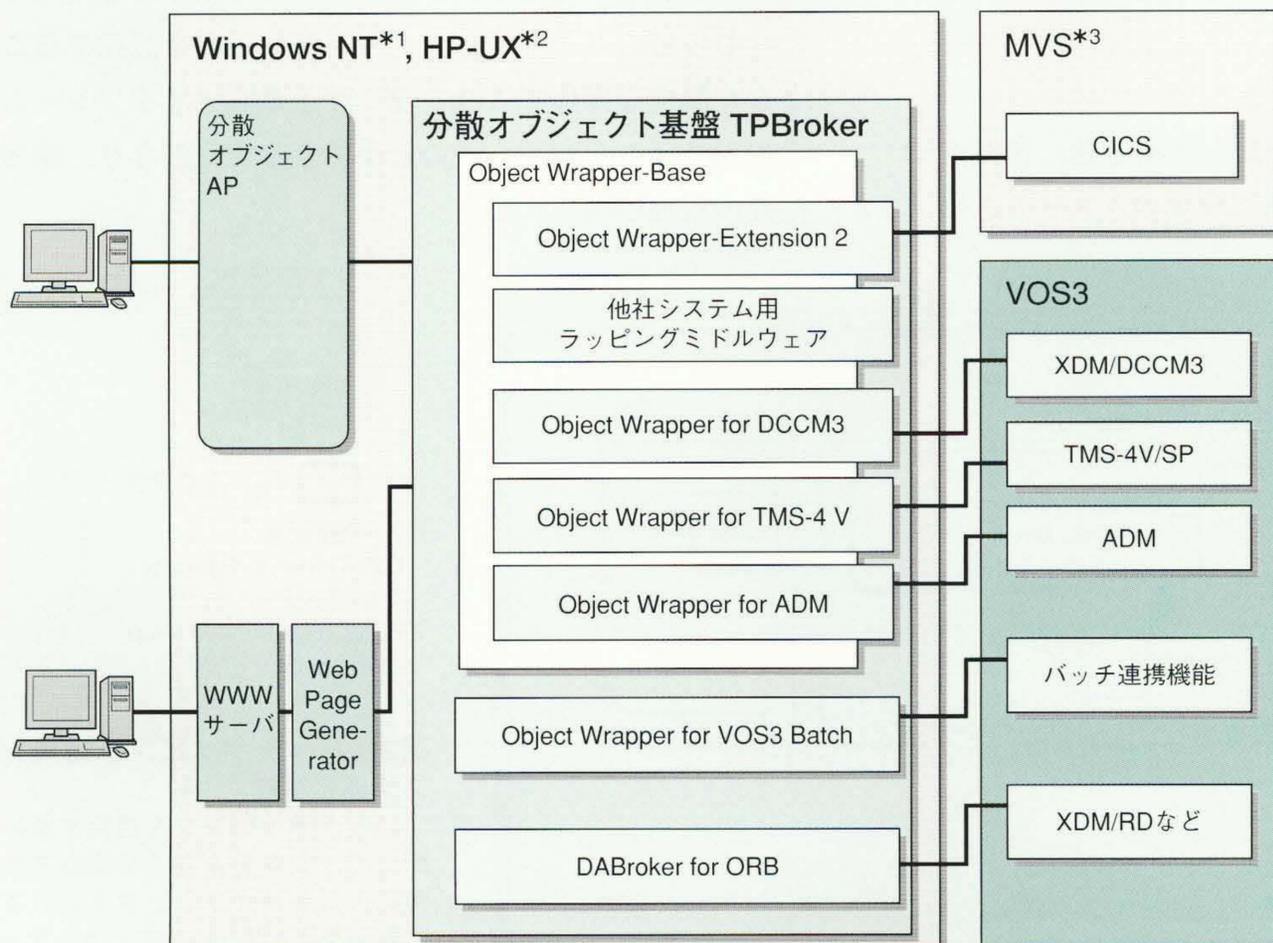
4 バッチシステムのラッピング事例

4.1 ねらい

メインフレームのCASE(Computer-Aided Software Engineering)ツール(バッチプログラム)をパソコン中心の環境からも容易に使えるようにするため、バッチラッピングを用いた新システムの開発を行った。

4.2 既存システム

既存システムは、(1) 各種プラットフォーム上に分散した定義系ツール、(2) メインフレーム上の生成系ツール、



注：略語説明ほか

- AP (Application Program)
- WWW (World Wide Web)
- DCCM3 (Data Communication and Control Manager 3)
- TMS-4V (Transaction Management System 4V)
- ADM (Add-Drop Multiplexer)
- VOS3 (Virtual-Storage Operating System 3)
- ORB (Object Request Broker)
- CICS (Customer Information Control Service)
- SP (System Product)
- RD (Relational Database)

*1 Windows NTは、米国およびその他の国における米国Microsoft Corp.の登録商標である。

*2 HP-UXは、米国Hewlett-Packard Companyのオペレーティングシステムの名称である。

*3 MVSは、米国における米国International Business Machines Corp.の登録商標である。

図1 レガシーラッピングミドルウェア製品群

日立製作所が提案しているObject Wrapper製品群と対象システムを示す。

および(3) プログラム資産で構成する。定義系ツールは、全国に展開する複数の開発拠点上で、ファイル仕様書、レコード仕様書、書式定義、プログラム仕様書の各仕様書定義作成に使用される。生成系ツールは、定義系ツールで作成された後に、メインフレームに転送された仕様書群を基に、プログラムソースの生成や、ドキュメントの生成などを行う。

新システムを開発するにあたっては、以下の要望があげられた。

(1) 既存システム資産の利用

十数年にわたるシステム開発のノウハウが蓄積されているメインフレームシステム資産(パターン、部品、各種ツール類)を今後も利用する。

(2) 操作性の改善

(a) 処理に必要な時間は1, 2分であり、待ち時間は対話処理としては長い。繁忙期ではジョブがすぐにスケジュールされず、レスポンスがさらに悪くなる。

(b) ユーザーがログインして、実行状況を明示的に確認しなければ、生成系ツールが終了したかどうかかわからない。

(c) ユーザーは、実行一確認の間はログインしていなければならず、回線が単一ユーザーに占有される傾向にある。

(3) パソコン環境への移行

ワークステーションで使用していた定義系ツールを、安価なパソコン環境に移行する。

(4) 共通開発環境、管理が容易

インターネットをベースに拠点間を連携させ、各拠点での開発環境を共通化する。Webベースのシステムを用

いることにより、インストールの手間を省く。

以上の要件を満たすため、Webブラウザによるクライアントとバッチラッピングを用いて、メインフレームを利用する新システムの開発を行った。

4.3 新システム構成

新システムでは、“Object Wrapper for VOS3 Batch”と、Webベースのクライアントの機能を考慮するとともに、新システムの構成を検討し、システム統合のためのミドルオブジェクトを新たに開発した。

ミドルオブジェクト(ローカルサーバ、コネクションサーバ)は、次のような機能を持つ。

(1) リポジトリ内の各種仕様書をファイル変換するためのAPI操作を行う。

(2) 投入したジョブの識別子の保持、状態チェック、結果の取得と管理を行う。

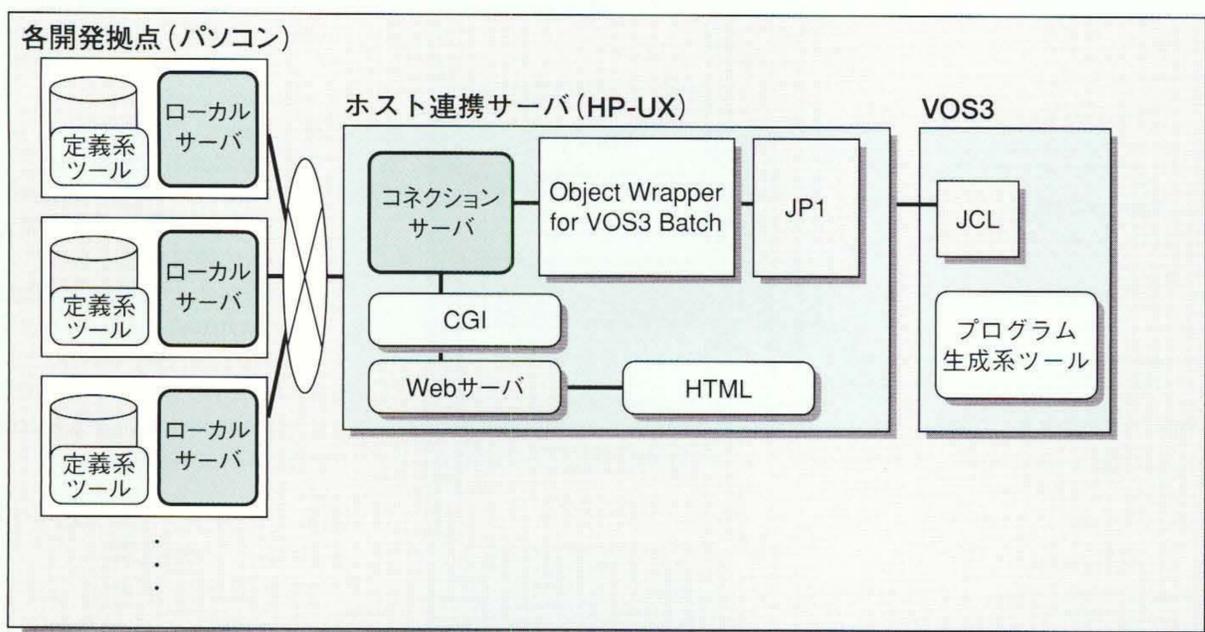
(3) リポジトリ上のファイル抽出・格納処理と、メインフレーム上の処理を連携させる。

新システムの構成を図2に示す。

4.4 評価結果

Webブラウザがインストールされたパソコンから、メインフレーム上の生成系ツールを利用することが可能となった。

また、既存のメインフレーム上のシステムを利用することから、高価なシステムをパソコンごとにインストールする必要がなくなり、クライアントの保守性も向上した。さらに、ユーザーは、メインフレームを意識することなく簡便な操作でWebブラウザ上からメインフレーム上のシステムを利用することができるようになり、開発期間の短縮が図られた。



注1: (ミドルオブジェクト)

注2: 略語説明
 CGI(Common Gateway Interface)
 HTML(Hypertext Markup Language)

図2 バッチラッピング適用システムの構成

バッチラッピングを適用するに際して、システム要件を満たすために、ミドルオブジェクトとしてコネクションサーバとローカルサーバを開発した。

5 オンラインシステムのラッピング事例

レガシーラッピング技術の早期適用と先行事例構築を目的として、社内システムの一部についてオンラインラッピングを適用した。その適用評価結果を以下に述べる。

5.1 ねらい

社内の情報管理システムにはメインフレーム系やUNIX、パソコン系などの異なるプラットフォームが混在しており、システムの再構築に際しては、以下のような要望があった。

- (1) Webインタフェースによってシームレスな画面に統合し、検索機能を充実させることにより、ペーパーレス化を実現し、調査工数を削減したい。
- (2) できるだけクライアント側にソフトウェアを配布しないで管理システムを稼働させたい。
- (3) ユーザーには、各業務システムの動作環境を意識させない、Webブラウザだけを活用させたい。

以上の要望を踏まえて、社内のSE (Systems Engineering) 作業を管理する一部の基幹業務システムを対象とし、オンラインラッピングを実現した。

5.2 対象システム

日立製作所のオンラインシステムであるXDM/DCCM3上で稼働する対象モデルシステムは、参照系と更新系を合わせて91の端末画面を定義するXMAP (Extensible Mapping Aids) ファイルで構成している。この中から、参照系を中心に、モデルシステムとして10画面 (7.3 kステップ) を対象にオンラインラッピングを適用した。

5.3 新システム構成

オンラインラッピング方式では、メインフレーム上の

XMAP画面をエミュレートするObject Wrapper製品をWebサーバにインストールすることにより、Webブラウザからの実行を可能とする。WebブラウザとObject Wrapper製品とのインタフェースとしては、CORBAメソッドの呼出しをHTML (Hypertext Markup Language) 上にテンプレートという形式で記述する“Web Page Generator”を利用した(図3参照)。

5.4 開発手順

オンラインラッピングを適用するにあたっては、以下の手順で開発を行った。

(1) 既存システムの解析

オンラインラッピングを適用するには、対象システムの画面遷移とキー操作を把握する必要がある。既存システムの解析方法としては、(a) 画面設計書から画面遷移を把握、(b) 実画面をメインフレーム上で稼働させ、目視による確認、(c) COBOLソースからファンクションキー操作の把握、(d) COPY句 (外部原始プログラム) から入出力インタフェースの把握を行った。

なお、この作業は適用業務システムを把握している担当者が行い、かつ画面設計書が整備されていたため、きわめて少ない工数で行うことができた。

(2) 新画面・画面遷移設計

画面設計書から画面遷移とファンクションキー操作を把握し、入出力フロー図を作成するとともに、画面統合と、Webインタフェース画面の設計を行った。基本的には、PFK (Programmed Function Key) によるメインフレーム上の次画面と前画面移動の操作を、Web上ではボタン操作、スクロール操作、フレーム処理にそれぞれ置き換えることにより、操作性の向上と画面の統合を考慮

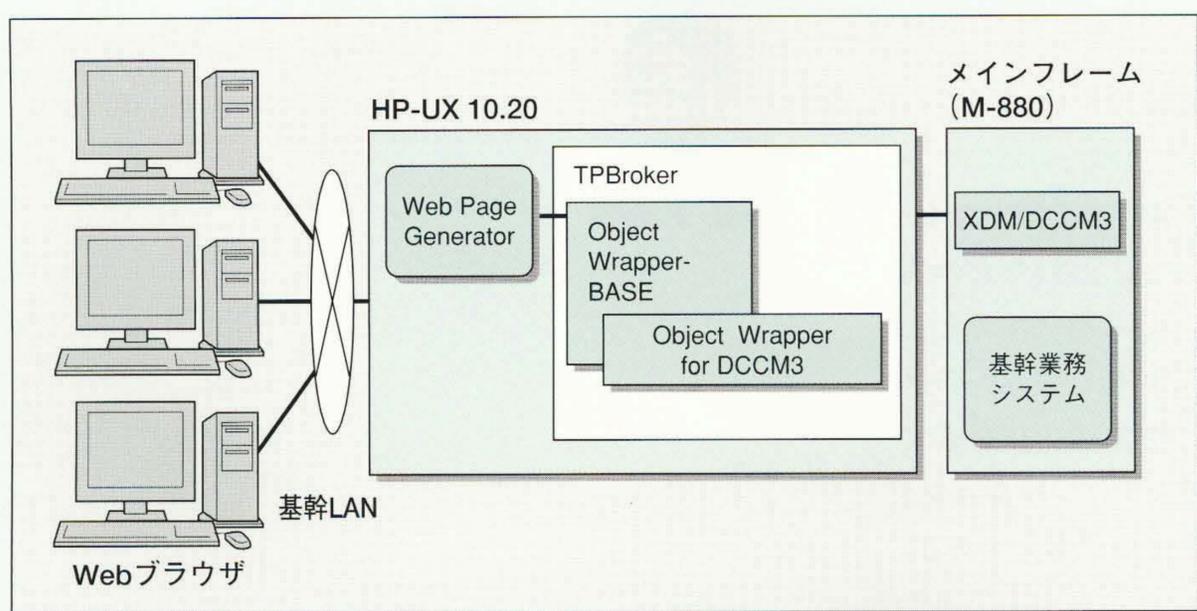


図3 オンラインラッピング適用システムの構成

オンラインラッピングを適用することにより、パソコンのWebブラウザからメインフレーム上の基幹業務システムを利用できるようにした。

した、新システムの画面を設計した。

(3) Object Wrapper定義

Object Wrapper定義ファイルとしては、(a) XMAP定義ファイル、(b) Object Wrapper-BASE定義ファイル、および(c) Object Wrapper for DCCM3定義ファイルがある。

XMAP定義ファイルは、メインフレームのXMAP画面定義情報から、画面上の各項目について位置・けた数・変数名をそれぞれ定義する。

DCCM3定義ファイル(DCCM3.cfg)では、それぞれの画面から実行されるキー操作をメソッドとして定義する。また、Object Wrapper-BASE定義ファイル(BASE.cfg)は、Webブラウザ上から実行するメソッド名を定義する。

(4) Web Page Generatorテンプレート定義

新画面・画面遷移設計フェーズで作成したHTML定義にCORBAメソッド呼出しのためのWeb Page Generatorテンプレートを追加し、Webブラウザ上でメインフレームをエミュレートするインタフェースを作成した。

5.5 評価結果

当該作業での開発規模を表2に示す。

Webブラウザに表示される画面数は6画面と、メインフレーム上のXMAP画面よりも少なくなっている。これは、HTML定義上でフレーム処理による画面統合を行い、CORBAメソッド呼出しについては別のテンプレートとして作成したためである。

また、ユーザーインタフェースでは、プルダウンメニューの付加や、ボタン、スクロールによる操作性の向上、Webブラウザによる共通インタフェースの提供などを実現しており、当初の開発目的を達成することができたと考える。

6 おわりに

ここでは、既存システム資産を有効活用する「レガシーラッピング」技術と、その事例について述べた。

現状では、レガシーラッピングの事例はまだ少なく、また、実装するうえではセキュリティ(安全性)などの問題が残されている。これらの問題を早期に解決するとともに、開発手順の蓄積や支援ツールの開発、製品へのフィードバックなどを行い、今後のビジネスの中核を成す技術としてこの「アプリケーションインテグレーション」によるシステム構築のニーズにこたえていく考えである。

表2 開発規模

メインフレーム上の対象画面(XMAP)10画面に対して、2.9kステップを新規に開発した。開発規模としてはWeb Page Generatorテンプレート定義が最も多い。

項目	本数	規模(kステップ)
対象既存システム	10	7.3
XMAP定義ファイル作成	7	0.4
Object Wrapper for DCCM3定義ファイル作成(メソッド)	19	0.7
Web Page Generator(HTML)定義	21	1.8
合計(開発規模)	47	2.9

参考文献

- 1) 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



渡部史朗

1991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、レガシーインテグレーション技術の開発とビジネス化展開に従事
情報処理学会会員
E-mail: watanabe @ bisd.hitachi.co.jp



神園幸三

1985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 情報システムセンタ 所属
現在、社内情報システムの企画・開発に従事
E-mail: kamizono @ system.hitachi.co.jp



只野完二

1988年日立製作所入社、情報・通信グループ 公共情報事業部 システム開発部 所属
現在、分散オブジェクト技術を中心とした生産技術の開発に従事
E-mail: tadano @ jkk.hitachi.co.jp



田中哲雄

1987年日立製作所入社、システム開発研究所 第二部 所属
現在、システムインテグレーション技術の研究開発に従事
情報処理学会会員、電子情報通信学会会員
E-mail: t-tanaka @ sdl.hitachi.co.jp