

# マルチ・プラットフォームアプリケーション統合開発環境 LINC 2000

Integrated Application Development Environment LINC 2000

山 科 順 一

**要 約** LINC はメインフレーム上の基幹トランザクション処理システム構築ツールとして実績を重ねてきたが、メインフレームばかりでなく、Windows 上でのシステム構築も可能となり、一つの LINC システム仕様（従来のソース・コード・プログラムにあたる）から、異なるプラットフォームへのアプリケーション構築が可能となった。

また、実行環境として IIS (Internet Information Server) と ASP (Active Server Pages) を使った Web アプリケーションの開発も可能であり、イントラネットからインターネットまで幅広いアプリケーション環境で利用が可能となった。

本稿では、マルチ・プラットフォーム環境 (CS シリーズ) の統合開発環境として進化した LINC 2000 の特徴から実装技術まで、Web サービスへの対応も含めて紹介する。

**Abstract** LINC has proven itself as an enterprise application development tool on mainframe systems, and allowed the system development on a Windows platform in addition to mainframe. As for LINC, it has enabled the same application system to be developed on different platforms, providing one LINC system specification (equivalent to previous source code)

Development of Web application using IIS (Internet Information Server) and ASP (Active Server Pages) has been made possible, and LINC system can be used in the wide range of application environment covering from intranet to Internet. This paper introduces the features of LINC 2000 evolved as Integrated Development Environment of multi platform environment, as well as implementation technology.

## 1. はじめに

これまで、メインフレーム上で開発されてきた基幹トランザクション処理システムが、UNIX や Windows 上で開発されるようになり、開発者は様々な技術の習得が求められるようになった。また、Windows クライアントの増加に伴う相互連携やインターネットを利用した企業間連携の必要性も高まっている。

一方で、メインフレーム上の基幹トランザクション処理システム構築ツールとして実績を重ねてきた LINC は、オープンな世界に進化した統合開発環境 LINC 2000 へ進化している。従来のメインフレームに加え、Windows 環境の ES シリーズ上でのシステム構築も可能となり、一つの LINC システム仕様（画面定義、データベース定義、アプリケーション・プログラムを含む従来のソース・コード・プログラムにあたる）から、異なるプラットフォームへのアプリケーション構築が可能となった。

また、コンポーネント技術の導入により IIS (Internet Information Server) と ASP (Active Server Pages) を使った Web アプリケーションの開発も可能となり、イントラネットからインターネットまで幅広い環境で利用することが可能となった。

本稿では、異なるプラットフォーム環境を用いてシステム開発を行う場合の開発ツールの要件を整理し、CS シリーズで LINC 2000 を利用することのメリットについて

説明する．

## 2. マルチ・プラットフォームシステム開発の課題と開発環境に求められる要件

従来メインフレームで開発されてきた基幹トランザクション処理システムが，UNIX，Windows といったプラットフォーム上で実現されてきている．本章では，異なるプラットフォーム環境を利用した開発の課題と求める開発環境の要件を整理する．

### 2.1 マルチ・プラットフォームシステム開発の課題

異なるプラットフォーム上でシステム開発を行う場合，それぞれのプラットフォーム毎に開発アプローチや開発のツールが様々であり，同じ開発者が複数のプラットフォーム上の開発に携わるのが困難である．メインフレームの開発担当者が Windows のアプリケーション開発に携わる場合や，その逆の場合も，基盤技術の違いから要員の再教育が必要となり，要員のローテーションや再配置を困難にしている．また，プラットフォームに依存した開発ツールを使用しているため，既存資産の再利用や移植にコストが掛かる．

### 2.2 開発環境に求められる要件

マルチ・プラットフォーム環境における開発環境の要件を，次の様に定義する．

- 1) プラットフォームに依存しない開発アプローチ  
分析，設計，開発，導入までの手順や考え方が，採用するプラットフォームに依存せず，共通のアプローチがとれること．
- 2) プラットフォームに依存しない開発環境  
使用する言語や開発ツールが，採用するプラットフォームに依存せず，共通な開発環境で開発可能なこと．また，開発生産性の点で GUI 環境であること．
- 3) 他のアプリケーションとの相互連携の容易性  
開発したシステム間の相互連携が容易であること．特に異なるプラットフォーム間のトランザクション連携が容易に実現できること．また，連携手段が標準的なインタフェースであること．
- 4) Windows クライアント環境との連携の容易性  
市販ツールを利用したクライアント・サーバ型や Web 環境など，様々な Windows クライアント環境への展開が容易であること．
- 5) 既存資産の継承と再利用  
既存資産の Windows クライアントとの連携や Web 連携といったモダナイズが容易に行えることや，既存システムの異なるプラットフォームへの移植や既存システムを再利用したシステム開発が容易に行なえること．

## 3. マルチ・プラットフォームシステム開発における LINC 2000 の特徴

LINC 2000 は，システム開発の生産性向上のために，利用する基盤技術とアプリケーション開発を分離し，開発者にアプリケーションの設計と開発に専念できる環境を提供している．

本章では，2 章で定義した開発環境の要件をどのように実現しているのか，LINC

2000 の特徴を紹介する。

### 3.1 開発環境の要件と LINC 2000 での実現

LINC 2000 では、1 章で定義した開発環境の要件を以下の考え方で実現している。

#### 1) プラットフォームに依存しない開発アプローチ

LINC 2000 では、ビジネス指向のモデリング技術を採用することで、利用する基盤技術とアプリケーション開発を分離している。そのため、メインフレーム上で稼働するアプリケーションも Windows 上で稼働するアプリケーションも共通のアプローチがとれる。開発者は、利用する OS やデータベースに関わるインタフェースをプログラミングする必要は無く、ビジネス・ルールのみプログラミングするため、アプリケーション開発に専念できる。そのためプロトタイピング・アプローチが容易に実現できる。

#### 2) プラットフォームに依存しない開発環境

Windows 上で稼働するプラットフォーム共通の開発環境を利用するため、開発者は採用するプラットフォームに関わらず共通の開発環境を利用できる。利用する OS やデータベースを意識することなく、プラットフォーム共通の LINC 定義言語を利用してプログラミングするため、新たな技術習得が最小限ですむ。

#### 3) 他のアプリケーションとの相互連携の容易性

LINC 2000 は LINC システム間同士で利用可能なトランザクション連携と、コンポーネント技術を利用した相互連携を提供している。これらのインタフェースを利用することで、システム間の相互連携が容易に実現できる。

#### 4) Windows クライアント環境との連携の容易性

3) と同様にコンポーネント技術を利用することで、Windows クライアントとの相互連携を実現している。クライアント・サーバ型や Web 環境など、様々な Windows クライアント環境への展開が容易である。

#### 5) 既存資産の継承と再利用

1), 2), 3), 4) で説明したとおり、Windows 環境と連携した既存資産のモダナイズ、既存資産の異なるプラットフォームへの移植が容易であり、既存資産の再利用を実現している。

### 3.2 LINC 2000 の特徴

本節では、3.1 節で説明した考え方に基づき、LINC 2000 の特徴を説明する。

#### 1) ビジネス指向の開発

#### 2) プロトタイピング開発

#### 3) オールインワン開発環境

#### 4) プラットフォームに依存しないアプリケーション開発

#### 5) 容易な他システムとの相互連携

#### 3.2.1 ビジネス指向の開発

LINC 2000 はビジネスの世界を LINC オブジェクト・モデルを用いて表現する。LINC オブジェクト・モデルは、次の三つの要素で構成される。

#### 1) コンポーネント

顧客情報や商品情報といった永続性のあるデータを表す。例えば、商品情報は

商品コンポーネントに対応づける。

コンポーネントは永続データであり、永続データの保守機能(追加, 変更, 削除, 照会の機能), データベース定義が LINC により自動生成される。

## 2) イベント

受注情報, 仕入情報といった業務活動から発生する取引データを表す。受注伝票を作成するという取引活動は, 受注イベントに対応づける。

LINC オブジェクト・モデルでは, イベントは取引活動の記録であり一定期間保持されるデータと考える。イベントには, 取引データの記録機能(データの追加機能)とデータベース定義が LINC により自動生成される。

## 3) プロファイル

コンポーネント, イベントに対する視点であり, 情報活用のキーとなる情報。例えば, 商品コンポーネントを商品番号をキーとして情報活用するためには, 商品番号プロファイルの商品コンポーネントに対応づける。

プロファイルはコンポーネント, イベントの検索キーとなるもので, データベースの INDEX にあたる。

図 1 に, 受注/出荷業務を例に対応付けのイメージを示す。図 1 では, 受注受付係が, 注文を受けた商品情報を参照しながら, 受注記録として受注伝票を作成する受注業務を LINC オブジェクトモデルにどのように対応づけているかを表わしている。

商品情報は商品名, 単価など永続的な情報を持っており, コンポーネントに対応づける。同様に, 得意先の永続的な情報として, 得意先名, 住所があり, 商品情報と同様にコンポーネントに対応づける。

受注記録としての受注伝票は業務活動から発生する取引データであり, イベントに対応づける。

商品情報を特定するために, 商品名あるいは商品コードを使用するが, これは情報利用の視点であり, プロファイルに対応づける。同様に得意先情報の視点として, 得意先名, 得意先コードをプロファイルに対応づける。

このように, LINC ではビジネスの世界をコンポーネント, イベント, プロファイルの三つの要素で表現する。

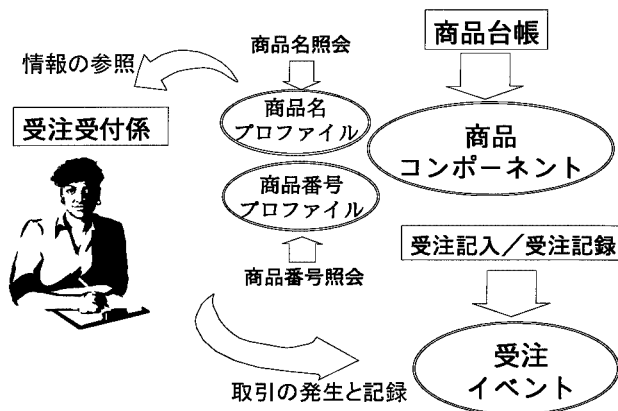


図 1 ビジネス指向の開発

### 3.2.2 プロトタイピング開発

LINC 2000 はプラットフォームに依存したプログラミングが不要であり、プロトタイピングを利用した繰り返し型開発が可能である。開発中のシステムは、リポジトリに格納されることにより常に整合性の取れた状態となり、その時の完成レベルで実際に動作可能なシステムである。

Windows 上で稼働する LINC 2000 の開発環境は、テストのための実行機能と対話型デバッガを提供しており、実行サーバへの生成を行うこと無くテストが可能である。開発環境のローカルデータベース（特別なデータベースソフトウェアは不要）を利用して、直ちに動作の確認が可能であり、短いサイクルで開発とテストが繰り返されるため、手戻りの無い開発が可能となる。

例えば、オンラインデータ入力考えた場合に、

第 1 ステップ：画面のユーザインタフェース設計と画面遷移の確認

第 2 ステップ：画面毎の更新処理と画面間のデータ連携の確認

第 3 ステップ：入力チェック、エラー処理の確認

というように、段階的に精度を上げていく進化型のプロトタイピングが可能となる。

また、データベース定義を含め、LINC 2000 が整合性の取れたシステムを自動的に生成、再編成するため、図 2 に示すような段階的に機能を拡張していく、開発サイクル全体を繰り返す進化型のアプローチも可能である。

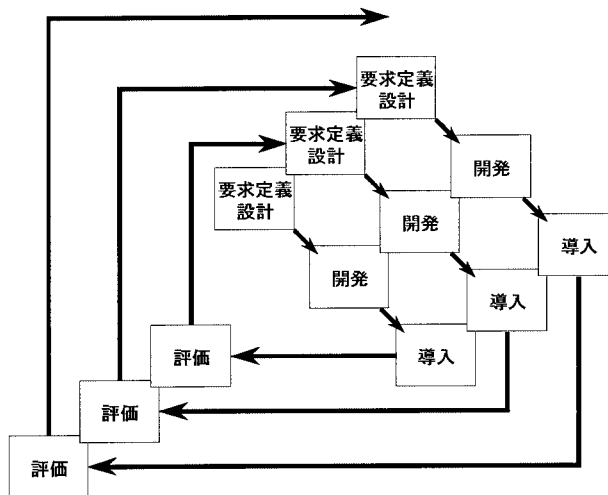


図 2 開発サイクル全体をプロトタイピングする例

### 3.2.3 オールインワン開発環境

LINC 2000 は、リポジトリを中心としたサーバ側アプリケーション、クライアント側アプリケーションの双方を開発可能な統合開発環境である。

LINC 2000 で定義したクライアントのプレゼンテーション部分（GUI の画面定義）は、プログラミング無しに Windows の GUI で利用したり、市販ソフトウェア（Visual Basic など）との連携や、Web 環境での利用が可能である。

LINC 2000 による開発環境と実行環境の概要を図 3 に示す。クライアント・アプリ

ケーションは、LINC 2000 で定義したユーザ・インタフェースをもとに再開発すること無く、様々なクライアント環境に短期間に適用可能である。

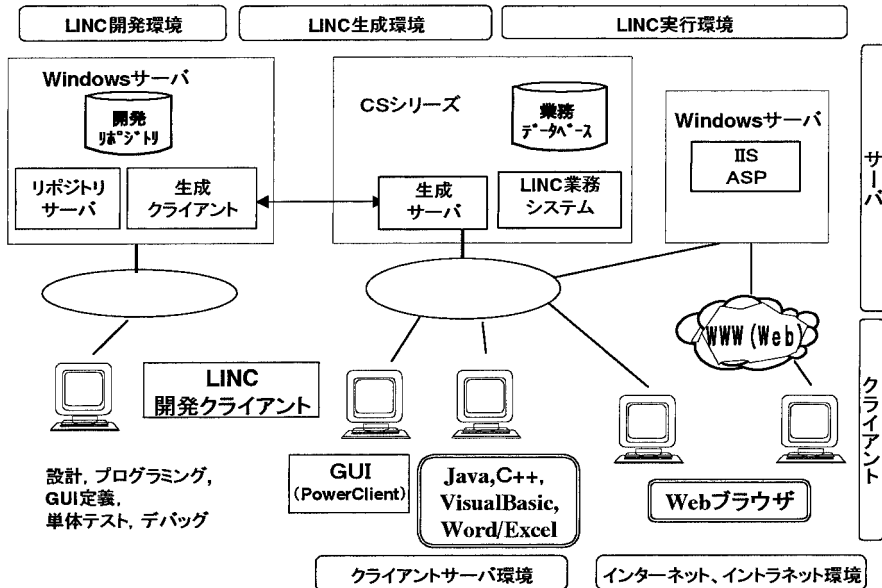


図 3 オールインワン開発環境

### 3.2.4 プラットフォームに依存しないアプリケーション開発

LINC 2000 は、強力な第四世代言語を用意し、リポジトリ内の LINC システム仕様から、業務プログラム、メッセージ制御、データベースまで、業務システム全体を自動的に生成する。OS やデータベースなど基盤となる技術とのインタフェース部分は LINC が自動生成するため、開発者は OS やデータベースソフトウェアに依存する部分を意識する必要はない。

言換えると、同一のシステム仕様から異なる実行プラットフォーム環境に業務システムを構築することが可能となる。また、既存システムの一部機能を別プラットフォームへ移植したり、新たな業務を開発し、メインフレーム環境と Windows 環境をトランザクション連携したアプリケーション構築も可能である。

現在 MCP, OS 2200 と Windows 環境に構築が可能である。

### 3.2.5 容易な他システムとの相互連携

LINC 2000 は、LINC システム間でのトランザクション連携を実現する HUB 機能と、Windows 環境との連携や LINC 以外の他システム間との相互連携を容易にするためのコンポーネント技術を提供している。

LINC 2000 のコンポーネント技術は、画面インタフェースを COM (Component Object Model) オブジェクト化する事により、クライアント環境から利用できるようにした。これにより、クライアントのアプリケーションは、COM オブジェクトを利用して LINC 2000 アプリケーションとインタフェースするため、Windows 上のオブジェクトインタフェースと同様のプログラミングが可能となり、LINC 固有のインタフ

エースを意識する必要が無い。また、最大 65,000 バイト長のメッセージを取り扱うことのできる拡張バッファ機能（リリース 3.2 から利用可能）を利用することで、従来のメッセージ長の制約が実質無くなり、インタフェースの柔軟性が拡大する。

コンポーネント技術の機能については 4 章で、他システムとの相互連携については、5 章でさらに説明する。

#### 4. LINC 2000 の開発環境と実行環境

LINC 2000 は以下の製品で構成されている。

- 1) 開発の中核であり、開発、テストを行う：LINC 2000 Developer
- 2) LINC 2000 Developer と連携しサーバ上にシステムを構築する：LINC 2000 Builder
- 3) サーバ上の LINC 業務システムの実行時ライブラリ群：LINC 2000 Runtime
- 4) クライアント上の GUI 実行支援ソフトウェア：PowerClient
- 5) LINC 2000 コンポーネント開発キット：ActiveLINC

本章では、LINC 2000 の開発環境と実行環境に分けて、LINC 2000 の構成を説明する。

##### 4.1 LINC 2000 の開発環境

###### 4.1.1 サーバ・アプリケーションの開発

LINC 2000 の開発環境は、リポジトリ・サーバを中核にしたグループ開発環境であり、アプリケーションの仕様は、LINC 2000 Developer のリポジトリに一元管理されている。

開発者は、LINC 2000 Developer の操作だけでシステムの開発、テスト、生成が可能であり、LINC 2000 Developer から LINC 2000 Builder へ生成指示を行うことで、実行サーバ上にシステムを生成することが可能である。

図 4 に示す様に、システム生成に当たっては、LINC 2000 Builder の生成クライア

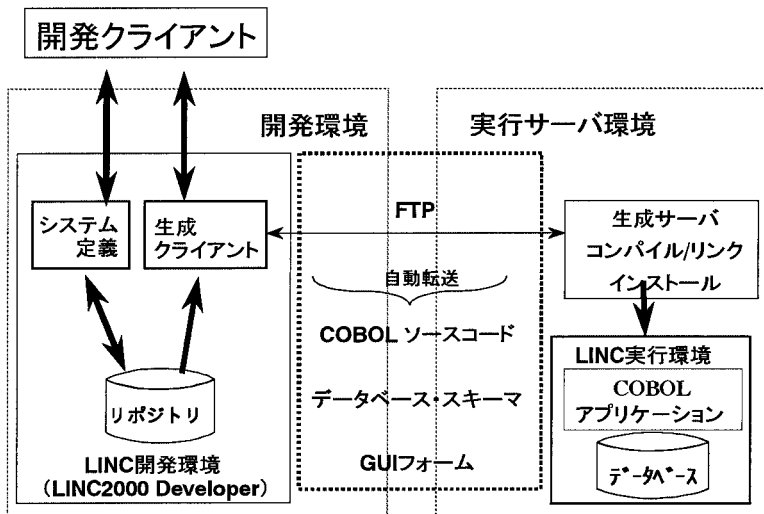


図 4 LINC 2000 のシステム生成

ントと生成サーバが連携し、業務プログラムのコンパイル、データベースの再編成を行う。

生成クライアントは、COBOL ソースコード、データベース・スキーマを生成し、実行サーバに転送、生成サーバによりコンパイルやデータベース再編成が行われる。

#### 4.1.2 クライアント・アプリケーションの開発

クライアント・アプリケーションの開発は図 5 に示すように、リポジトリ上のユーザインタフェース定義をもとに、Windows の GUI (実行支援ソフトウェア PowerClient を使用)、Visual Basic などの市販ツールとの連携(ActiveLINC が生成する COM オブジェクトを使用して Visual Basic アプリケーションを開発)や ASP を使用した Web 化(ASP ジェネレータを利用)など、再開発無しに、使用するクライアント環境に合わせたアプリケーションを生成することができる。

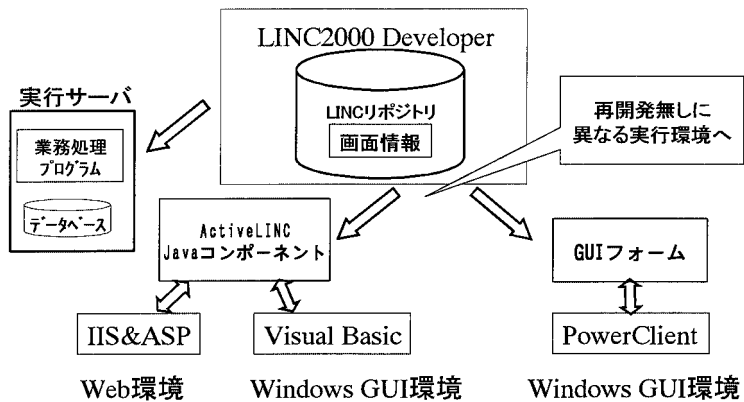


図 5 様々なクライアント環境を生成

Windows の GUI で利用する場合、LINC 2000 Developer で定義した GUI 画面は、クライアントの実行環境である PowerClient を利用することで、すぐに GUI 環境で利用できる。LINC 2000 Developer は PowerClient が利用する画面定義情報を自動生成し、実行サーバにアップロードする。アップロードされた GUI 情報は、クライアント上の PowerClient がアクセス時にクライアント側のバージョン情報とサーバ上のバージョン情報を比較し、常に最新の GUI 情報をクライアントにダウンロードするため、運用管理が容易に行える。

つぎに、3.2.5 項でコンポーネント技術を利用した相互連携ツールとして紹介した ActiveLINC を使ったクライアント・アプリケーションの開発について説明する。

ActiveLINC は、LINC 2000 Developer のサブモジュールとして機能し、図 6 に示すように、LINC 2000 の ISPEC (Interface SPECification, LINC 2000 でのコンポーネント、イベントの総称)に対応する JavaBeans を生成する。生成された Java Beans は米国マイクロソフト社 Java VM (Java Virtual Machine) により Windows 環境で COM オブジェクトとして利用可能となる。LINC 2000 のサーバ・アプリケーションとは LINC 2000 が標準提供する RAS (Remote Access Server) を介してインタフェースする。



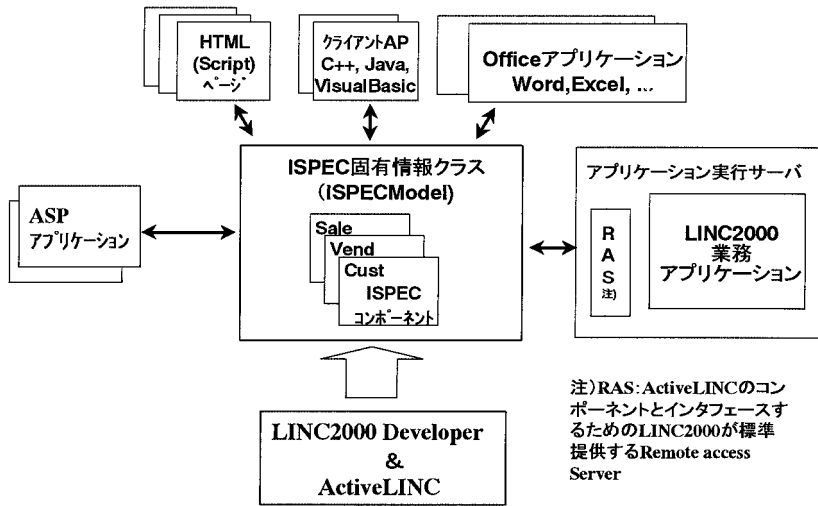


図 6 ActiveLINC の概念図

ActiveLINC が生成する COM オブジェクトは、Visual Basic などの市販ツールや WORD、EXCEL といったオフィス・アプリケーション、ASP など様々な Windows 環境から利用可能であり、LINC 2000 のアプリケーションを業務部品として再利用可能にする。

ActiveLINC が生成する JavaBeans は、ISPEC を COM オブジェクト化するが、ユーザインタフェースは持たない。

Visual Basic などの市販ツール連携では、ActiveLINC が生成する Java Beans を各クライアントに導入して利用する。Visual Basic のアプリケーション開発は、市販のツールで行う。

ASP を使った Web 連携の場合は、市販ツールを使って、ASP アプリケーションを開発することもできるが、図 7 に示すように ActiveLINC では LINC 2000 Developer の画面定義に基づく ASP ファイルの自動生成機能 (ASP ジェネレータ) を Ac-

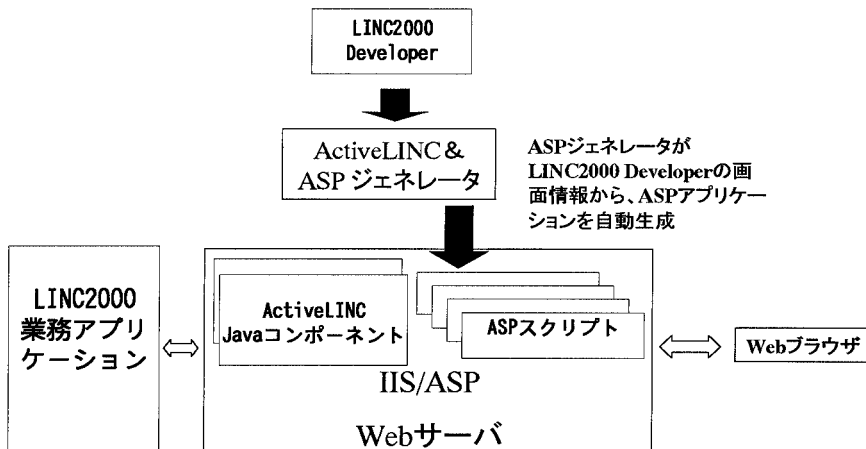


図 7 ASP ジェネレータによる Web 化

tiveLINC のオプション・ソフトウェアとして提供している．この機能を利用することで ASP やスクリプト・プログラミングの知識が無くとも，簡単に Web 化することができる．

#### 4.2 LINC 2000 の実行環境

プラットフォーム毎の基盤となるソフトウェア環境について説明する．

##### 4.2.1 サーバ・アプリケーションの環境

LINC 2000 で開発されたサーバ・アプリケーションは，COBOL とデータベースで構築される．MCP 環境では，COBOL 85 と DMSII，OS 2200 では UCS COBOL と UDS RDMS 1100，Windows ではマイクロフォーカス社 NetExpress (COBOL) とデータベース・エンジンとして ORACLE または SQL Server を使用する．

##### 4.2.2 クライアント・アプリケーションの実行環境

LINC 2000 の実行環境は，端末エミュレータ，クライアントの実行支援環境である PowerClient を利用した Windows の GUI，ActiveLINC を利用した市販ツール (Visual Basic など) 連携，IIS と ASP による Web 環境から選択可能である．

MCP，OS 2200 環境では，端末エミュレータ，PowerClient を利用した Windows の GUI，ActiveLINC を利用した市販ツール連携，ASP を利用した Web 化から選択が可能である．

Windows 環境では，PowerClient を利用した Windows の GUI，ActiveLINC を利用した市販ツール連携，ASP を利用した Web 化から選択可能である．

さらに，クライアント環境は混在が可能である．例えば，営業活動で利用する在庫検索，納期照会，受注処理は Web で利用し，社内で使用する業務は Windows の GUI で利用する，といった混在環境が構築できる．また，現在は PowerClient を使った Windows の GUI であるが，社内の Web 化に伴う Web 環境からの利用に変更する場合も，最小限の変更で対応が可能である．

### 5. 他アプリケーションとの相互連携

本章では 3 章で説明した他システムとの相互連携について，CS シリーズでの適用例を交えてさらに説明を加える．

LINC 2000 では他アプリケーションとの連携方法として，二つの方法を提供している．一つは LINC システム間同士のトランザクション連携であり，もう一つは Web サービスを利用した相互連携である．

#### 5.1 LINC システム間の相互連携

LINC システム同士の連携方法として，独自の 2 相コミットメカニズムを持った HUB という機能を提供している．

HUB 機能は，LINC システム間をトランザクション連携するもので，端末から入力されたのと同様のプログラミングで実現が可能である．開発者は 2 相コミットのプログラミングや障害回復について意識する必要は無い．

HUB 機能は同一サーバ内の LINC システム間，異なるサーバ間の LINC システム間，さらに異なるプラットフォーム上の LINC システム間で利用できる．ただし，異なるプラットフォーム間で，漢字データをやりとりする場合には，開発者側の責任で

コード変換を行う必要がある。

この機能を利用することで、特別な連携ソフトウェアを導入することなく、メインフレーム上の LINC アプリケーションと Windows 上のアプリケーションを容易に連携することが可能となる。

サブシステム単位に構築したアプリケーションを HUB 機能で連携することで、段階的な開発や個々のサブシステムのシステム規模を小さくすることが可能となり、開発、保守の容易なシステム分割が可能となる。

## 5.2 Web サービス

インターネットを利用したこれからの企業間連携の手段として、Web サービスが注目されている。LINC 2000 では、4 章で紹介した ActiveLINC の次期バージョン（2002 年 4 Q リリース予定）として、ISPEC を Web サービス化する Web サービス・ジェネレータ機能の提供を予定している。この機能を利用することで、Web サービスの構築が容易になる。

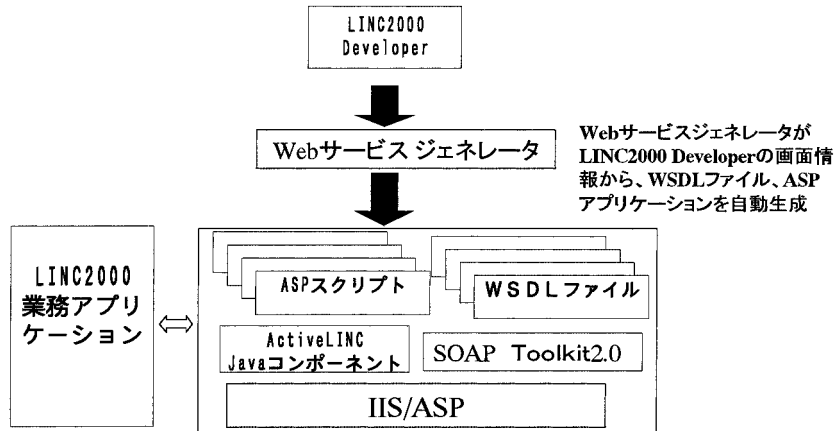


図 8 Web サービスの開発

Web サービスの開発は図 8 に示すように、ActiveLINC の開発環境と同様であり、Web サービス・ジェネレータにより、Web サービスとして ISPEC を公開するための WSDL (Web Services Description Language) ファイルと LINC アプリケーションとインタフェースするための ASP ファイルを生成する。

Web サービスジェネレータが生成する WSDL ファイルには、ISPEC のデータ属性が XML (Extensible markup Language) 形式で記述されている。ASP ファイルには SOAP インタフェースでやり取りするクライアントと LINC アプリケーションを連携するための以下の機能が記述されている。

- ① クライアントからの SOAP メッセージの受け取り
- ② 要求された ISPEC の決定
- ③ XML メッセージを ISPEC のデータ並びに対応づけ
- ④ サーバ上の LINC アプリケーションとの接続
- ⑤ ISPEC へのメッセージ送信

- ⑥ ISPEC からのデータ受信
- ⑦ ISPEC のデータ並びを XML に対応づけ
- ⑧ SOAP メッセージとしてクライアントに送信

Web サービスとして ISPEC を公開する場合、取り扱えるメッセージ長が大きな制約となってくる。そこで、3.2.5 項で紹介した拡張バッファ機能を利用することで、従来のメッセージ長の制限にとらわれないインタフェースが可能となる。

LINC 2000 による Web サービスの開発では、新しい技術の習得が最小限であり、短期間に実現できる。また、本格的な Web サービスの開発においても ActiveLINC を使って LINC アプリケーションをコンポーネント化することで、Web アプリケーション技術者に LINC アプリケーションとの固有のインタフェースを隠蔽することが可能となり、開発が容易になる。

### 5.3 異なるプラットフォーム上のアプリケーション連携の例

LINC 2000 を利用すれば、メインフレーム環境、Windows 環境どちらの環境にもシステム構築が可能となる。既存の受発注管理システムを利用して、新取引先支援システムを構築し、HUB 機能を用いて受発注管理システムとはリアルタイムなトランザクション連携を行うアプリケーション開発も可能である。このシステムに ActiveLINC を利用することで、インターネットを利用した e ビジネス環境が容易に構築できる。LINC 2000 でシステム開発することで、CS シリーズの異種 OS 環境を有効利用した e ビジネス環境が構築可能となる。

図 9 にメインフレーム環境で稼働している既存受発注システムを利用して、Windows 上に新取引先支援システムを構築したイメージを示す。

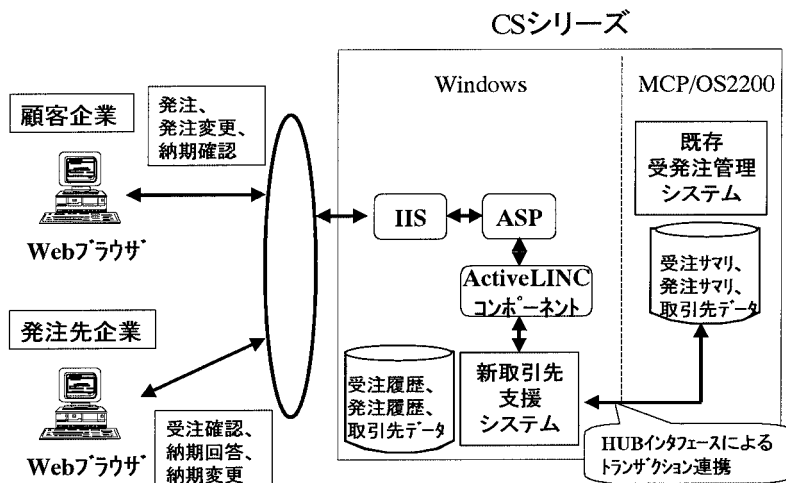


図 9 CS シリーズでの相互連携システム

この例では、既存システムの一部を切り出し、新機能を加えて Windows 環境に生成している。MCP 環境の受発注管理システムとは HUB 機能を用いてリアルタイムなトランザクション連携を行う、クライアント環境には ActiveLINC の Web 連携機能を用いて、従来の画面を Web 化している。インターネット環境では、IIS を利用

した SSL (Secure Socket Layer) による暗号化を行うことで、セキュリティを確保している。

サーバ・アプリケーションは既存資産を最大限に利用し、Web 化については、ActiveLINC を利用することで、再開発を最小限にしている。LINC 2000 は、このようなアプリケーションを容易に構築できる。

## 6. おわりに

LINC 2000 は、従来の LINC 資産を確実に継承しながら、最新 IT を導入していくことができる開発環境である。現在、サーバ・アプリケーションの開発とクライアント・アプリケーションの開発が LINC 2000 Developer に統合され、操作性の高い Windows 環境ですべての開発が行える様になり、従来よりも高い開発生産性が得られるようになった。

開発環境が Windows 環境に統合されたことで、異なるプラットフォームへの移植性の高いアプリケーション開発が容易となった。

相互連携の点でも、コンポーネント技術の導入により、柔軟かつ標準技術に基づいた相互連携が可能となった。

LINC 2000 は従来の LINC のイメージを大きく変える開発環境に進化している。情報システムが対象とする業務は、LINC が登場した頃と比べると、遙かに多くの分野で利用されるようになってきた。しかしながら、BtoB、BtoC を問わず、トランザクション処理が情報処理システムの基本であることは、今後も変わりはないと筆者は考えている。今後も最新 IT を取り込んだ開発環境として、開発者の方々に期待されるソフトウェアにしていきたい。

---

### 執筆者紹介 山科 順一 (Junichi Yamashina)

1955 年生。1978 年専修大学経営学部情報管理学科卒業。  
同年日本ユニシス(株)入社。流通業の客先システム開発、  
LINC の適用支援担当を経て、現在第二ソフトウェアサー  
ビスセンターマイグレーションサービス二室に所属。