

## 開発支援ツール MIDMOST/DE for COBOL の採用

### Introduction of Development Supporting Tool “MIDMOST/DE for COBOL”

山 崎 明 彦

**要 約** BankVision の設計・製造・テストを支援するために、MIDMOST/DE for COBOL（以下、MIDMOST/DE）を採用した。MIDMOST/DE は「MIDMOST を基盤とする COBOL による AP 開発のための開発支援ツール」という位置付けのクライアントサーバ形式のツールであり、設計工程から保守工程までをカバーする各種ツール群で構成されている。BankVision の基本 AP 構造を堅持するための各種機能を有し、大規模開発でのプログラム品質向上に寄与している。

また BankVision 自体が Windows 基盤上で稼働するという特徴を生かし、単体テスト工程、および統合テスト工程において、プログラム開発用クライアント PC 上でのテスト実行を可能とすることで、特に大規模開発において課題となる、限られたテスト環境資源の有効利用を実現した。

**Abstract** In order to support the design, development, and testing of BankVision, we adopted MIDMOST/DE for COBOL (hereafter, MIDMOST/DE). MIDMOST/DE, which is a client-server style tool positioned as “development supporting tool for application development in COBOL based on MIDMOST”, contains various tool suites covering from the design process to the maintenance process, and various functions to hold fast to a fundamental application structure of BankVision, as well as contributes to the program quality improvement in a large-scale development project.

Making full use of the characteristics that BankVision itself runs on the Windows infrastructure, and enabling the testing on the client PC for program development in both the unit test and integration test processes, achieve the effective use of the limited resources of testing environment which will be a challenge especially in the large-scale development project.

#### 1. は じ め に

BankVision<sup>\*1</sup> の開発に際して、設計・製造・テストを支援するために、MIDMOST/DE for COBOL<sup>\*2</sup>（以下、MIDMOST/DE）を採用した。MIDMOST/DE は「MIDMOST を基盤とする COBOL による AP 開発を支援する開発ツール」という位置付けのクライアントサーバ形式のツールであり、これまでに BANKSTAR（セブン銀行で稼働中のオープン勘定系システム）、BankForce（国際勘定系パッケージ）等にて採用実績がある。

BankVision では、その特徴の一つであるコンポーネント指向をベースとする AP 構造を実現するために、基本となるプログラム構造規約を規定しているが、MIDMOST/DE は、それらの規約を遵守するための自動規約チェック、設計情報登録管理、各種自動生成機能等を有している。また BankVision 自体が Windows 基盤上で稼働するという特徴を最大限に生かし、単体テスト工程、および統合テスト工程において、プログラム開発用クライアント PC 上での

テスト実行を可能とすることで、限られたテスト環境資源を有効利用した。特に統合テスト工程の支援機能は、今回の BankVision への適用に際して新たに開発した機能である。

本稿では、MIDMOST/DE について、2 章で概要を説明し、3 章で設計・プログラム支援機能の、4 章でテスト支援機能の特徴を述べる。

## 2. MIDMOST/DE の概要

### 2.1 BankVision への適用までの経緯

MIDMOST/DE は、地域金融機関向け勘定系システム SBI21 用の開発環境であった SBI21/DE がベースとなっている。SBI21 は実行環境こそ、当時の主流であるメインフレームであるが、そのアプリケーション構造にオブジェクト指向のカプセル化の概念を取り入れ、かつ開発環境に Windows 搭載の PC を採用したことが大きな特徴であり、これらを反映・実現する目的で開発されたのが SBI21/DE である。

SBI21 の考え方は、PC 上でプログラム作成から単体テスト工程までを実施し、実行確認がとれたプログラムをメインフレーム上の実行環境に転送して稼働させるというものである。そのために、SBI21/DE では、主に COBOL ベンダ間で相違のある日本語機能部分を対象とするソースコード変換機能やメインフレーム実行環境のデータベースエミュレート機能等により、PC と実際の実行環境との相違を吸収したが、その実装は将来的に実行環境のオープン化を想定したものになっていた。

その後 SBI21 は、セブン銀行向け勘定系システム BANKSTAR として、実行環境に Windows & MIDMOST を採用しオープン化されたが、SBI21/DE は、その開発環境として改造され、適用された。それをベースとして汎用化のためのツール群整備がほどこされたものが、MIDMOST の標準開発環境、MIDMOST/DE である。MIDMOST/DE は、同時期に開発を開始した国際勘定系システム BankForce 等への採用を経て更に機能が整備されたのち、BankVision の開発環境として適用されるに至った。BankVision への適用に際して、基本の方針・考え方はそのままであったが、特筆すべき機能拡充として、これまで単体テストまでしかサポートしていなかったテスト支援機能に、統合テスト支援機能を追加している。

### 2.2 全体構成

MIDMOST/DE は、図 1 に示すとおり、設計工程から保守工程に至るプログラム管理の範囲をカバーするツール群として提供されている。

BankVision は、そのベースとなった SBI21 の特徴である「オブジェクト指向のカプセル化を採用したオンライン AP 構造」を踏襲している。したがって、設計工程からプログラミング工程は、SBI21 開発において規定されたオンライン AP 構造・ネーミング規約等に準拠することによって、既に標準化された手順に従い、既存のツール群でのサポートが可能となっている。この部分は、COBOL で記述された勘定系 AP に特化したツールであるため、汎用性には乏しい面がある。

それに対して単体テスト工程以降～保守工程をサポートする部分は、比較的汎用的に使用可能なツール群で構成されている。次章以降では、MIDMOST/DE の、設計・プログラミング支援、およびテスト支援機能の概要について述べる。

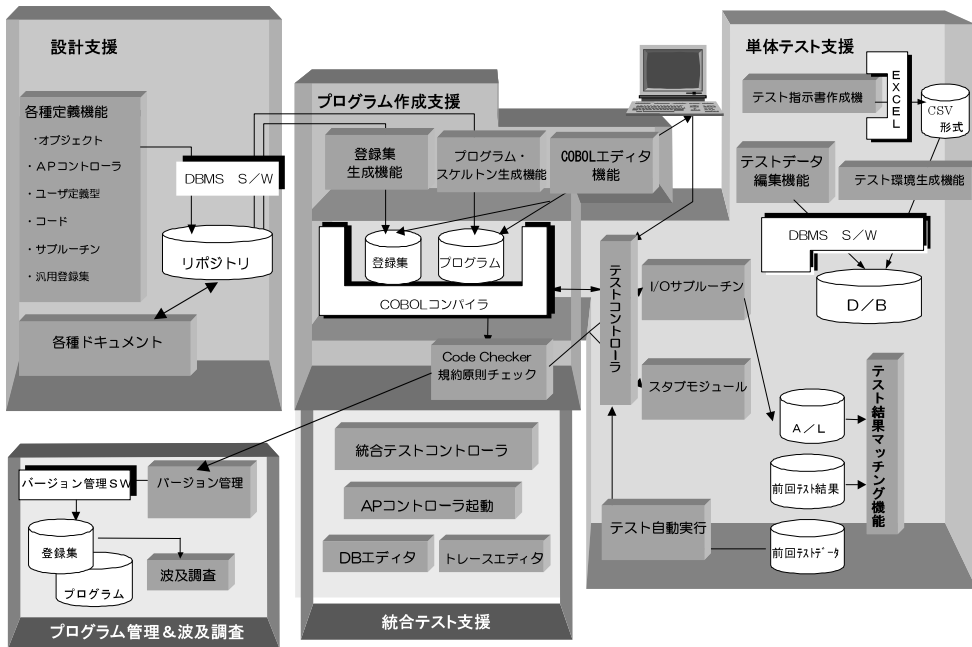


図1 MIDMOST/DE 全体イメージ

### 3. 設計・プログラミング支援機能

#### 3.1 BankVision の AP 構造

設計・プログラミング支援機能を説明するにあたり、BankVision のオンライン AP 構造に関して、簡単に概要を説明する。図2は、普通預金の出金処理に関する、AP 構造の構成要素とその関連を簡略化して示したものである。図2に示すとおり、コンポーネント指向に基づきサブシステム内にデータをカプセル化し、他のサブシステムからデータを参照できない構造をとっている。

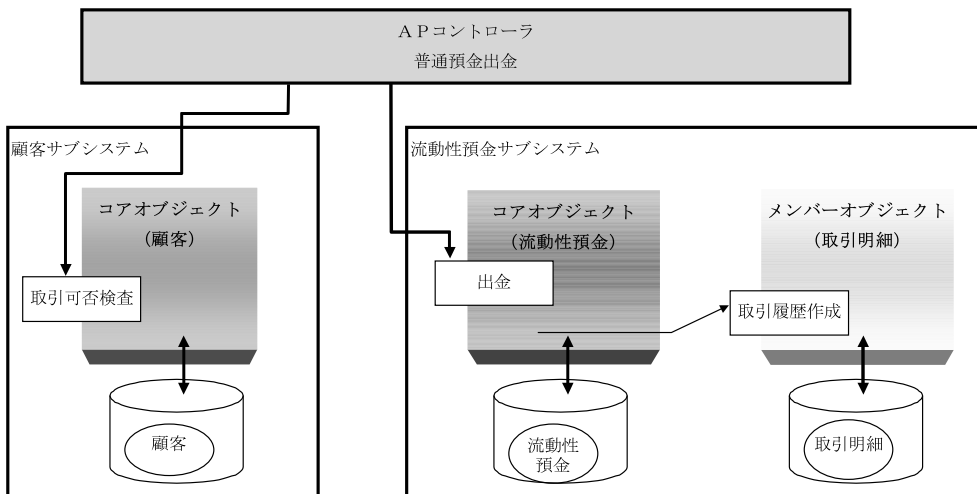


図2 AP 構造概念図

BankVision のオンライン処理構造は全て、図2に示す AP 構造に基づく仕組みとなっており、取引ごとに用意された AP コントローラから、役割単位に分割したサブシステムのコアオブジェクトを、取引に必要な順番で呼び出すことで、処理を遂行する。それぞれの要素の説明を表1に示す。

表1 AP 構造の主な構成要素

構成要素	図中の要素	実装形態	説明
APコントローラ	普通預金出金APコントローラ	COBOL プログラム	取引単位の制御プログラムとして主に、その取引の遂行に必要な役割をもつサブシステム (=コアオブジェクト) の呼び出し処理を記述している。
サブシステム	顧客サブシステム 流動性預金サブシステム	同一ネーミング規約 で結ばれる論理単位	BankVisionのカプセル化の単位であり、勘定系システムを役割単位に分割したコンポーネントである。実際には200余りのサブシステムが存在する。
コアオブジェクト	顧客コアオブジェクト 流動性預金コアオブジェクト	COBOL プログラム	サブシステムを構成する代表モジュールでありサブシステムの役割の実装。サブシステムに一つだけ存在する。
メンバオブジェクト	取引明細メンバオブジェクト	COBOL プログラム	コアオブジェクトと結びつきの強いオブジェクト。サブシステムに0個以上が存在する。
サービス	取引可否検査 出金 取引履歴作成	COBOL プログラム内の SECTION	オブジェクトが持つ機能の実装。コアオブジェクトのサービスはサブシステム外部に公開し、メンバオブジェクトのサービスはサブシステム内部に公開する。
テーブル	顧客 流動性預金 取引明細	RDBの表	オブジェクトと1対1で対応する。オブジェクトの属性を定義したデータ構造体と同じレイアウトである。

これらの要素は、名称、および用途は違っても、基本的に COBOL プログラムである。よって開発にあたっては、この用途の違いを正確に把握し、設計・実装する必要がある。これを支援するのが設計支援機能である。

## 3.2 設計支援機能

業務プログラムを開発するにあたり、BankVision で規定されているオンライン AP 構造の各種規約に則って作業を実施する必要があるが、特に大規模開発において、これらを開発メンバー全員に対して周知徹底するのは難しい。MIDMOST/DE では、設計レベルでこれらの各種規約を遵守させるための仕組みを用意することで、これを解決している。

### 3.2.1 AP 設計情報投入・管理機能 (リポジトリエクスプローラ)

オンライン AP 構造の基本要素であるサブシステム・オブジェクトの定義情報に加え、これ以外の各種サブルーチン、バッチプログラム等の全ての COBOL プログラム定義情報を一元管





### 3.3 プログラミング支援機能

#### 3.3.1 プログラムスケルトン生成、登録集生成

リポジトリエクスプローラから投入された定義情報は、プログラムスケルトン、およびデータ構造体である登録集として自動生成される。プログラムスケルトンとは、AP 構造で規定されたプログラムの骨組み部分と定型的なデータ定義やロジック定義を指し、プログラマは生成されたスケルトンに対し、業務処理の詳細部分のみを追記するスタイルでプログラミングを行う。これにより、AP 構造を特別に意識することなく、規定に則ったプログラムを作成できる。仮にプログラムの定義情報が修正され、既に生成されたスケルトン部分が変更される場合には、追記された業務処理の詳細部分と、変更を反映して再生成されたスケルトン部分をマージして再構成する機能も備えている。BankVision のオンラインプログラムの場合、自動生成さ

表3 リポジトリエクスプローラ生成対象情報

生成情報種類	生成内容	生成形式	説明
プログラムスケルトン	APコントローラ	COBOLソース	取引単位の制御プログラムとして主に、その取引の遂行に必要な役割をもつサブシステム（=コアオブジェクト）の呼び出し処理を前提とした構造で生成される。
	コアオブジェクト	COBOLソース	サブシステムが持つ役割を実装するプログラムとして、公開サービス（サブシステム外部に公開する機能）をSECTIONとして生成する。
	メンバオブジェクト	COBOLソース	コアオブジェクトに従属するプログラムとして、内部サービス（サブシステム内部で呼び出される機能）をSECTIONとして生成する。
	サブルーチン	COBOLソース	様々な処理に共通な、またはある処理に特化したプログラムとして生成される。
	IOサブルーチン	COBOLソース	オブジェクト単位に、対応するテーブルへのアクセスを行うプログラムとして、属性定義のデータ構造体を基に、ストアドプロシージャ基本機能（select, insert等）の呼び出し処理を中心に生成される。
	バッチプログラム	COBOLソース	バッチ処理用プログラムとして基本情報部分のみが生成される。
登録集	属性	COBOLソース	オブジェクトと1対1で対応する属性データ項目定義をデータ構造体として生成する。
	入力パラメタ	COBOLソース	サービスと1対1で対応する受け渡しデータ項目定義をデータ構造体として生成する。
	出力パラメタ	COBOLソース	同上。
	サブルーチン属性	COBOLソース	サブルーチンで専用テーブルを持つものに1対1で対応する属性データ項目定義をデータ構造体として生成する。
	パケット	COBOLソース	サブルーチンと1対1で対応する受け渡しデータ項目定義をデータ構造体として生成する。
	汎用登録集	COBOLソース	様々な用途で使用する各種データレイアウト定義をデータ構造体として生成する。
SQL	テーブル	SQL文	属性データ項目定義から生成される。オブジェクトと1対1で対応するテーブルレイアウト生成用のSQL文を生成する。
	インデックス	SQL文	オブジェクトと1対1で対応するテーブルレイアウトに対してのインデックス設定用のSQL文を生成する。
	ストアドプロシージャ	SQL文	属性データ項目定義から生成される。オブジェクトと1対1で対応するテーブルレイアウトに対してのSP設定用のSQL文を生成する。
その他	コード定義	COBOLソース	プログラムで使用する各種コード情報定義をデータ構造体として生成する。
	ユーザ定義型定義	COBOLソース	類似のデータ項目間で使用する項目の基本属性定義、生成時に各データ構造体に反映される。

れるスケルトン部分の割合は約 40% となっている。登録集は、プログラムが参照する実際のデータ構造体の形式で 100% 自動生成される。さらに、オブジェクトの属性定義から、オブジェクトに対応するテーブル情報（テーブル生成文、インデックス定義、ストアードプロシージャ等）を SQL 形式で生成することで、オブジェクトの属性定義と実際のデータベース環境の整合性を保証している。

その他の定義情報から生成されるものも含めて、リポジトリからの生成物を表3にまとめた。

### 3.3.2 AP 構造各種規約チェック機能

業務処理の記述が終了したプログラムは、PC 上の COBOL コンパイラにてコンパイルし、デバッグするが、MIDMOST/DE は、コンパイル前に BankVision の AP 構造規約に違反していないかを機械的にチェックする機能を提供している。これを規約原則チェック機能と呼び、主に次の 2 点をチェックしている。

#### 1) カプセル化のチェック

（BankVision においては、オブジェクト属性の外部からの隠蔽が主たる目的）

#### 2) BankVision の AP 構造規約チェック

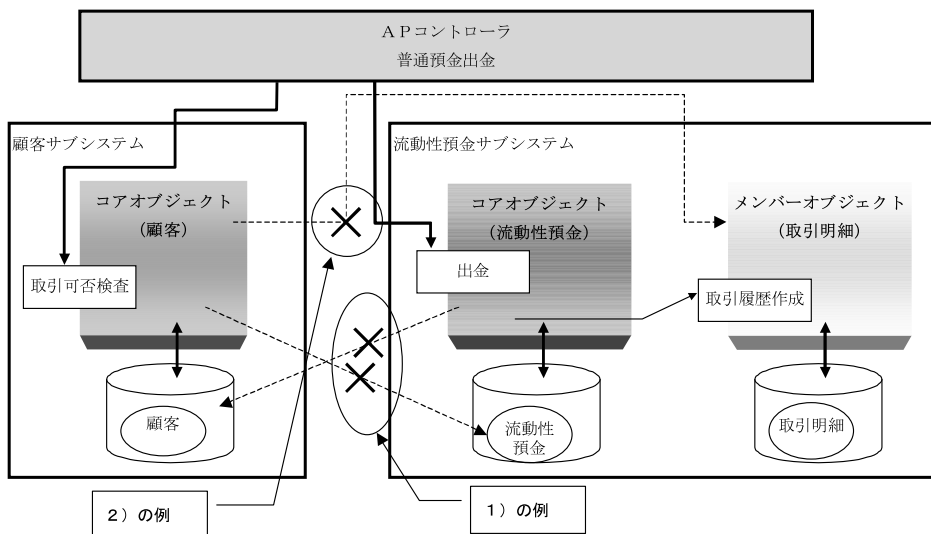


図5 参照制約の例

図5は、チェック対象のうち、参照制約に関する代表的な2例を示したものである。

1) については、サブシステム内にカプセル化されたデータに対する不正な参照が行われていないかどうかのチェック機能である。通常のオブジェクト指向言語であれば、言語レベルでデータの参照制約が可能であるが、BankVision の場合、オブジェクト指向技術を支援する機能を有しない COBOL<sup>\*3</sup>を採用しているため、独自の規約を基にカプセル化のチェックを行っている。

2) は、BankVision が独自に定めた各種規約のチェック機能である。BankVision では AP 構造の歪み防止のために、コンパイラではチェック不可能な不正なプログラム間呼び出しが行われないよう、プログラム同士の参照制約を設けている。



1) 2) とともに、ネーミング規約を定め、このネーミングレベルで不正な参照、呼び出しを行っているかを機械的にチェックしている。

このように開発者が厳守すべき規約を体系的にチェックすることによって、開発時に定めた規約が将来の保守段階においても維持できるように配慮している。

#### 4. テスト支援機能

MIDMOST/DEの単体テスト支援機能では、開発者が机上で使用するクライアントPC上に、実行環境のミドルソフトである MIDMOST を導入することで、実際の実行環境とほぼ同等のテストを行うことが可能となっている。

また、実行環境が Windows となったことを利用して構築した新たな仕組みが、統合テスト支援機能である。

##### 4.1 単体テスト支援機能

単体テスト支援機能は、クライアント PC 上に DB 環境を構築し、実際のトランザクション(以下、TRX) 処理を前提としたテストを実施可能にする。

単体テスト支援機能の目的は主に以下のとおりである。

- 1) 個々のオンラインプログラムが実行する DB 更新内容の正当性検証
- 2) 個々のオンラインプログラムから返される出力パラメタの正当性検証
- 3) 個々のオンラインプログラムに渡される入力パラメタに対するチェック処理の正当性検証、および入力パラメタとその時の DB 状態の相関チェック処理の正当性検証

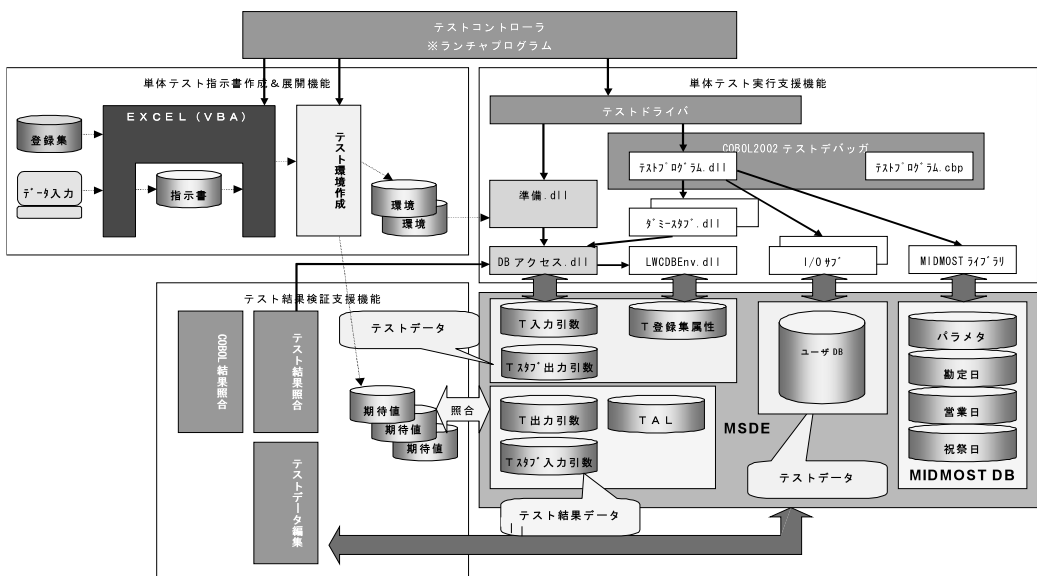


図6 単体テスト支援機能概要図

図6は、単体テスト支援機能の概要図である。大きくは、テストデータ、およびテスト環境の作成を支援する単体テスト指示書作成&展開機能、テスト実行を支援する単体テスト実行支援機能、テスト検証を支援するテスト結果検証支援機能に分かれている。

これらの機能により、別途共有サーバ上に特別な単体テスト用環境を構築する必要がないため、開発拠点内でのテスト作業には環境に起因する制約が生じない。BankVision 開発においては、開発拠点での開発はもとより、オフショア発注先を含む協力会社にこれを貸与することで、協力会社内での単体テストまでを低コストで効率よく実施できた。

#### 4.1.1 単体テスト指示書作成&展開機能

単体テストを行うためにはまず、単体テスト用のデータと環境を作成しなければならない。本機能では、この単体テスト用データ・環境を定義するために「単体テスト指示書」と呼ぶ Excel 形式のブックを用意している。

単体テスト指示書に定義するデータには次の 6 種類がある。

- ① オブジェクトが使用するデータベースの初期値
- ② ①に対するテスト実行後の期待値
- ③ テストドライバが被テストプログラムに渡す引数の値
- ④ 被テストプログラムがテスト終了後に返す実行結果の期待値
- ⑤ スタブを呼び出す際にセットする引数の期待値
- ⑥ スタブから返される引数の戻り値

ここで記述している「テストドライバ」とは、被テストプログラムを呼び出すダミープログラムであり、「スタブ」とは、被テストプログラムが呼び出す代理応答用ダミープログラムのことである。これらの情報は単体テスト指示書テンプレート内の専用入力シートに、専用の入力補助機能を使用して入力する。

作成後の単体テスト指示書を基に、単体テスト指示書展開機能がクライアント PC に単体テスト環境を自動作成する。また、単体テスト指示書を Excel で作成することで、柔軟な入力・修正も可能となっている。

前述のとおり本機能では、テスト実行後の期待値（上記の②、④、⑤）を定義できる仕組みを提供することにより、後述するテスト結果検証支援機能を利用して、テスト結果の正当性確認を機械的に行うことが可能となる。

#### 4.1.2 単体テスト実行支援機能

単体テスト実行支援機能は、任意のテストパターンを選択し実行指示することで、目的のテストを実行する。

単体テストの実行には、通常実行とデバッグ機能を使用するデバッグ実行の二つの方法があるが、いずれも、実際の BankVision 実行環境と同様、MIDMOST 上でテストを行う。図 6 に示すように、バッチプログラムとして実行されたテストドライバ（exe モジュール）が、オブジェクト・サブルーチン等のオンラインプログラム（dll モジュール）を被テストプログラムとして呼び出し、テストを実行する仕組みである。

単体テストはクライアント PC 上の 32bit 環境で実行されるが、実際の 64bit 実行環境での稼働も、ソースコードの修正を行うことなくリコンパイルのみで対応可能である。

#### 4.1.3 単体テスト結果検証支援機能

単体テスト結果検証支援機能は、テスト結果の検証作業負荷軽減を目的として、以下を対象

とするテスト実行結果の自動比較機能を提供している。

- ① データベースの更新結果とその期待値
- ② テスト実行後の被テストプログラムの引数の戻り値とその期待値
- ③ 被テストプログラムがスタブに渡す引数値とその期待値

比較結果は各構造体の項目単位に表示し、不一致項目は色を変えて識別できるようになっている。また、不一致項目以外は非表示にする機能も提供し、検証範囲の特定も可能になっている。

## 4.2 統合テスト支援機能

通常のオンライン取引は、営業店端末から営業店 I/F サーバを経由して勘定系 AP サーバに電文が送信され、業務処理が実行される。しかしながら、テスト環境に設置可能な営業店端末の台数は限られるため、テスト実施に制約を強いられる。統合テスト支援機能は、図 7 に示すとおり、統合テストサーバに接続されたオンラインテスト用 PC 上で、Excel 形式のオンライン AP テスト指示書にテスト対象の AP コントローラを起動するための入力電文を定義し、実際の端末を経由せずに、AP サーバに入力電文を送信することで業務アプリケーションを呼び出し、統合テストを行うことが可能である。

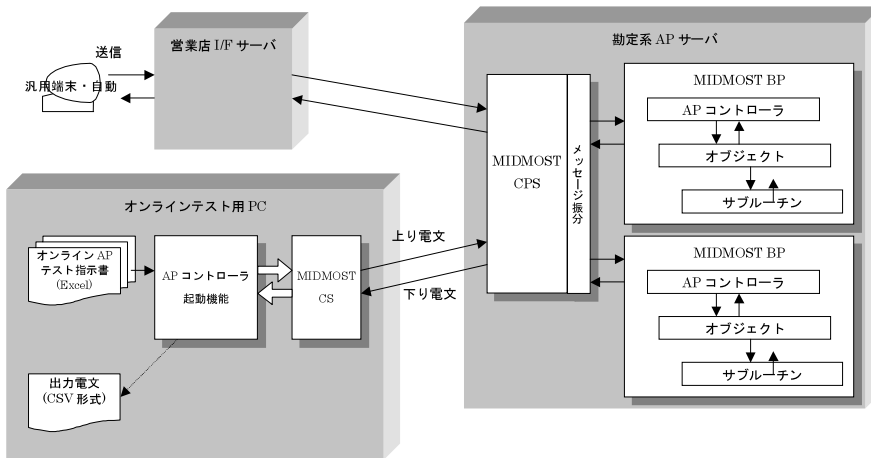


図 7 オンラインテストツールイメージ

本節では、統合テスト支援機能の代表的な機能であるオンラインテストツールに関して概要を述べる。

### 4.2.1 オンラインテストツールの基本機能

オンラインテストツールを使用する目的は主に以下のとおりである。

- 1) 各 AP コントローラの DB 更新内容の正当性検証
- 2) 各 AP コントローラの実送電文内の各データ項目値の正当性検証

なお、出力電文の実送信処理は行われなため、画面/伝票/通帳等への実際の印字内容に関しては、本ツールの対象外である。

オンラインテストツールは、表4に示す機能で構成されている。

表4 オンラインテストツールを構成する機能群

機能	内容
I/O トレースログ編集機能	APコントローラ、オブジェクト、IOサブなどが出力するトレースログ（MIDMOSTのスナップダンブログへのダンブデータ）を、登録集（データ構造体）形式に編集し参照する機能。
トレースログ Excel 出力機能	I/O トレースログ編集機能から呼び出し・実行される機能。 トレースログから必要なデータを抽出し、「オンラインAPテスト指示書」を自動で作成する機能。
オンラインAPテスト指示書機能	Excelを使用して、テストに必要なテストグループ、テストケース、および登録集レイアウトでのデータ定義を行う機能を提供する。 指示書にはAPコントローラを起動する為の上り電文、下り電文期待値、DB初期値、DB更新期待値を定義することが可能。
APコントローラ起動機能	「オンラインAPテスト指示書機能」で指定されたテスト情報に従い、上り電文をAPサーバへ送信しAPコントローラを起動する機能。
テストデータ展開機能	「APコントローラ起動機能」から呼び出され、テストの実行時にオンラインDBへDB初期データを展開する機能。
結果マッチング機能	「オンラインAPテスト指示書」で定義された期待値データと、I/O トレースログに出力された実行結果データのマッチングを行う機能。 受信下り電文、DB更新結果のマッチングを行うことが可能。
業務エラーメッセージ確認機能	APコントローラ起動機能が受信した下り電文をCSV形式で保存し、Excelを使用して、業務エラーの端末宛メッセージレイアウトで表示する機能。
Non-Commit 指示機能 ※BankVision のインフラ側で提供される機能	業務トランザクションの完了時に、応答電文は正常に端末に返すが、DBの更新をコミットしない設定とその解除を行う機能。
トレースログの採取指示機能 ※BankVision のインフラ側で提供される機能	業務トランザクションの実行時に出力されるI/Oトレースログの採取の指示を行う機能。

#### 4.2.2 オンラインテストツールの仕組み

オンラインテストツールは、APコントローラ起動機能を中心に構成されている。APコントローラ起動機能は、Excel形式のオンラインAPテスト指示書経由で与えられたテスト情報

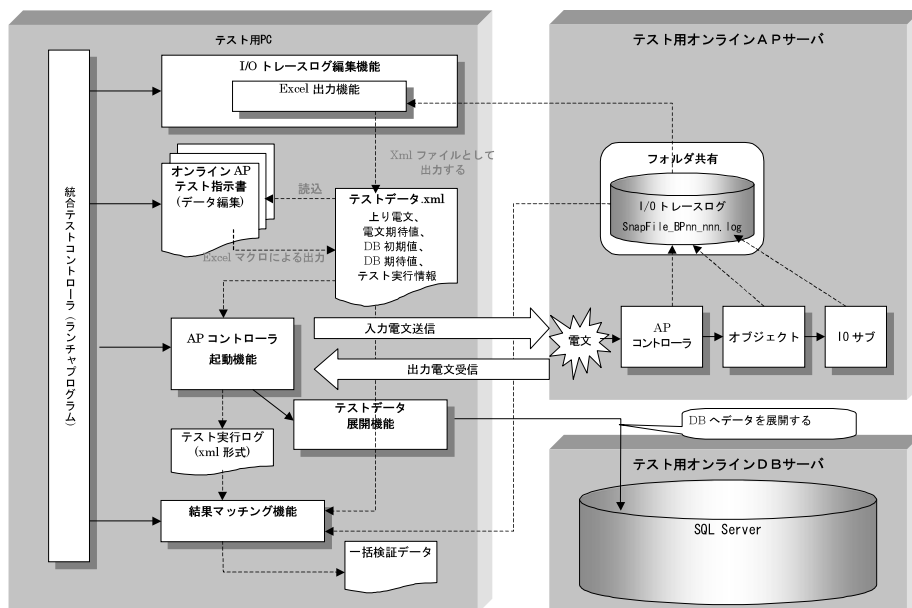


図8 オンラインテストツール構成図

を基に電文を作成し、MIDMOST CS の I/A 型メッセージ送受信機能を使用して AP コントローラを起動し、以下の入力形態のテストを実行する。

- ① 1：端末
- ② 2：自動機
- ③ 3：センタカット<sup>\*4</sup>
- ④ 9：融資稟議支援

オンラインテストツールは、図 8 のような構成となっており、統合テストを実施するサーバに接続可能な PC であればテストを実施できる。

#### 4.2.3 オンラインテスト指示書の作成

オンライン AP テスト指示書機能により、テストグループ、テストケースの作成、各テストパターン毎に業務 AP の起動に必要な上り電文の作成、下り電文期待値の定義を行う。また実際の営業店端末から実行した結果の取引電文トレースログから、オンライン AP テスト指示書を自動作成することも可能である。

オンライン AP テスト指示書は、図 9 に示す構造で、必要なテスト情報を設定し、1 テストパターン 1 トランザクションの単位で定義を行う。1 テスト情報には、複数の AP コントローラに関する情報を含ませることが可能である。

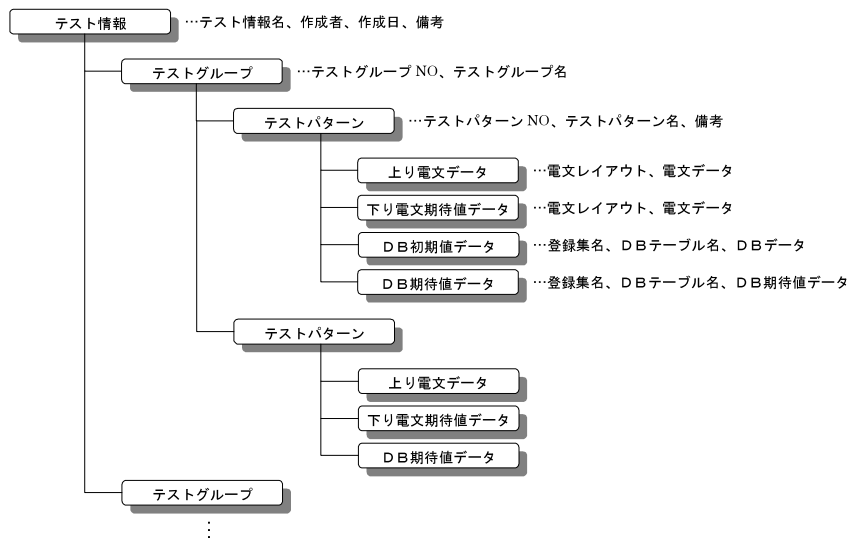


図 9 オンライン AP テスト指示書の構成

#### 4.2.4 オンラインテストの実行イメージ

次に、AP サーバへ電文を送信するため、AP コントローラ起動機能を使用してオンライン AP テスト指示書を読み込む。図 10 にて、AP コントローラ起動機能の使用イメージを業務エラー確認テストを例に説明する。

図 10 に示すとおり、画面の左側に展開されたツリー構造から、テストパターン/ケースを選択し電文を送信する。送信電文、及び受信電文はそれぞれ、送信メッセージフィールド、受信メッセージフィールドに表示される。

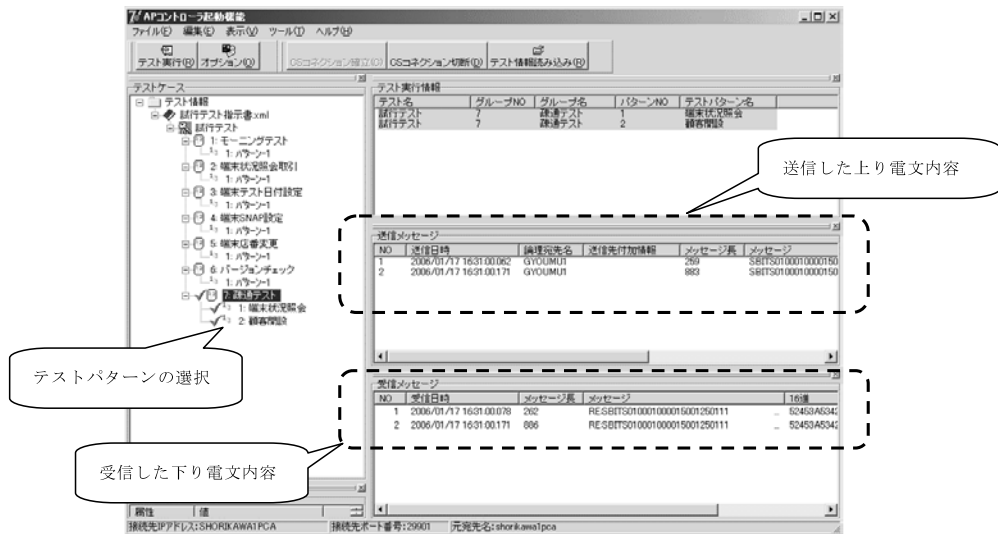


図 10 電文の送信と結果の受信

また図 11 の TRX 間のデータ連携によって、オンラインテストツールを使用して連続する取引を実施する際に、前回実行結果の下り電文を次に実行する上り電文に利用できる。

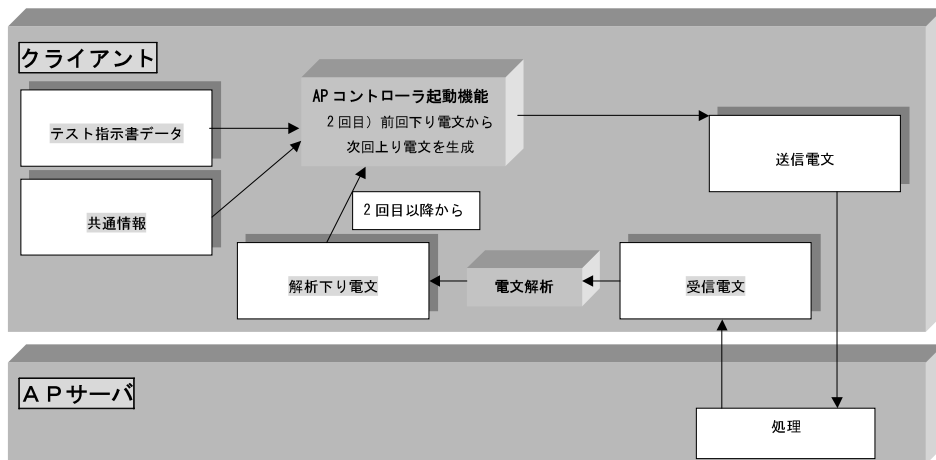


図 11 TRX 間データ連携機能

#### 4.2.5 オンラインテスト実行結果の確認

オンラインテスト実行後は、I/O トレースログ編集機能、結果マッチング機能、業務エラーメッセージ確認機能等を使用して実行結果を確認することができる。

図 12 にて、業務エラーメッセージ確認機能の使用例を示す。受信した下り電文内容を CSV 形式で保存後、エラーメッセージ確認ツールを起動し、CSV ファイルを開くことでエラーメッセージを確認することが可能である。

受信日時	エラー文言	エラー発生情報	任意付加情報
2006/1/16 20:44:58.125	9002オベミス	TRSC05	0004他店指定時は横番省略不可
2006/1/16 20:44:58.281	9002オベミス	UITCF01	0005取引先区分?
2006/1/16 20:44:58.125	9002オベミス	TRSC05	0004他店指定時は横番省略不可
2006/1/16 20:44:58.281	9002オベミス	UITCF01	0005取引先区分?
2006/1/16 20:44:58.125	9002オベミス	TRSC05	0004他店指定時は横番省略不可
2006/1/16 20:44:58.281	9002オベミス	UITCF01	0005取引先区分?
2006/1/16 20:44:58.125	9002オベミス	TRSC05	0004他店指定時は横番省略不可
2006/1/16 20:44:58.281	9002オベミス	UITCF01	0005取引先区分?
2006/1/16 20:44:58.125	9002オベミス	TRSC05	0004他店指定時は横番省略不可
2006/1/16 20:44:58.281	9002オベミス	UITCF01	0005取引先区分?
2006/1/16 20:44:58.125	9002オベミス	TRSC05	0004他店指定時は横番省略不可
2006/1/16 20:44:58.281	9002オベミス	UITCF01	0005取引先区分?
2006/1/16 20:44:58.125	9002オベミス	TRSC05	0004他店指定時は横番省略不可
2006/1/16 20:44:58.281	9002オベミス	UITCF01	0005取引先区分?
2006/1/16 20:44:58.125	9002オベミス	TRSC05	0004他店指定時は横番省略不可
2006/1/16 20:44:58.281	9002オベミス	UITCF01	0005取引先区分?
2006/1/16 20:44:58.125	9002オベミス	TRSC05	0004他店指定時は横番省略不可
2006/1/16 20:44:58.281	9002オベミス	UITCF01	0005取引先区分?
2006/1/16 20:44:58.125	9002オベミス	TRSC05	0004他店指定時は横番省略不可
2006/1/16 20:44:58.281	9002オベミス	UITCF01	0005取引先区分?
2006/1/16 20:44:58.125	9002オベミス	TRSC05	0004他店指定時は横番省略不可
2006/1/16 20:44:58.281	9002オベミス	UITCF01	0005取引先区分?
2006/1/16 20:44:58.125	9002オベミス	TRSC05	0004他店指定時は横番省略不可
2006/1/16 20:44:58.281	9002オベミス	UITCF01	0005取引先区分?
2006/1/16 20:44:58.125	9002オベミス	TRSC05	0004他店指定時は横番省略不可
2006/1/16 20:44:58.281	9002オベミス	UITCF01	0005取引先区分?

図 12 端末宛エラーメッセージの確認

#### 4. 2. 6 業務開発グループによるオンラインテストツールの運用イメージ

BankVision の開発において、実際に業務開発グループがオンラインテストツールを適用した際の運用イメージを示す。

まず、想定したテストは以下のような内容である。

テスト名 : ○○入金  
 事前設定取引 : A 開設 B 登録  
 結果検証資料 : X 照会 Y 取明照会  
 事前に準備する指示書 : ○○入金 A 開設 B 登録 X 照会 Y 取明照会  
 端末コミット区分設定

このようなテストを表 5 の流れで実施した。

表 5 オンラインテストツールを適用したテスト実施例 (イメージ)

SEQ	実行場所	取引名	備考
①	P C	A 開設	事前設定データの準備。
②	P C	B 登録	事前設定データの準備。
③	P C	X 照会	事前照会は P C から行い、照会票は⑧にて出力する。
④	P C	Y 取明照会	事前照会は P C から行い、照会票は⑨にて出力する。
⑤	P C	端末コミット区分設定※1	Non-Commit 設定を O N にする。
⑥	P C	○○入金	D B 更新内容・T R X 出力を検証。 検証 N G ならば、原因追及 & 対応を行い、⑥を繰り返す。
⑦	P C	端末コミット区分設定	Non-Commit 設定を O F F にする。
⑧	端末室	出力再検証	③ X 照会の照会票を出力する。
⑨	端末室	出力再検証	④ Y 取明照会の照会票を出力する。
⑩	端末室	○○入金	入力画面のハードコピー・帳票の印書をする。
⑪	端末室	X 照会	
⑫	端末室	Y 取明照会	

事前データの準備からテスト実行までを PC 上で実施することで、設置台数の限られたテスト用営業店端末の有効活用が可能となった。また Non-Commit 設定を有効にすることで繰り返しテストが可能となるため、事前設定データの再設定が不要となった。

## 5. お わ り に

本稿では、MIDMOST/DE for COBOL が用意するツール群のうち、基本 AP 構造をサポートするツール群、および BankVision 開発への適用に際して特に新規開発した統合テストツールの概要・適用イメージについて述べた。

本ツール群が適用できる範囲は、COBOL をプログラミング言語とし、MIDMOST を基盤とする開発にはほぼ限定されているが、その範囲内で、かつ AP 構造・各種規約等への準拠が可能であれば、基本 AP 構造、開発手順、およびテスト実施手順等の横展開が期待できる。既に BankVision への適用に先立って、いくつかの開発現場への適用も進んでいる。当面の拡張計画として、S-BITS<sup>\*5</sup> 共同 OSC (アウトソーシングセンタ) の開発体制をサポートするライブラリ管理の仕組み、複数開発案件のテストサポート等が予定されている。

BankVision 開発への適用においては、今回の開発現場に特化した手順・環境での機能拡充を行わざるを得なかった面もあるため、これらを S-BITS 共同 OSC 体制下における共同開発環境運用に、いかにして適用させていくかが今後の課題である。

- 
- \* 1 日本ユニシスと百五銀行が共同で開発した、世界で初めてとなる Windows ベースのフル・バンキング・システム。
  - \* 2 MIDMOST は、日本ユニシスがメインフレームで培ったノウハウを基にオープン基盤でミッションクリティカル処理を安全・確実に行うために開発したミドルウェア。MIDMOST/DE for COBOL は、MIDMOST を使用した COBOL 言語向け開発環境。
  - \* 3 実際には OO-COBOL 機能を有するが、COBOL 提供ベンダ固有の部分もあるため採用していない。
  - \* 4 MIDMOST CS からの I/A 型メッセージ送信により起動可能な端末センタカットのみが対象であり、全てのセンタカット処理に対応しているわけではない。
  - \* 5 S-BITS (= Succeeding Banking Information Technology for Success consortium, エスビッツ) 新たなバンキングシステムを研究し、その実現のため、対応策の策定と実証モデル(ひな型)開発を行い、参加行に提供することを目的とした、次期バンキングシステム検討・検証コンソーシアム。日本ユニシスと日本ユニシス地方銀行勘定系ユーザー7行で構成されている。2000年11月15日に設立。また、コンソーシアムで策定したグランドデザインを実現するために発足した次期オープン基幹系システムの共同開発プロジェクトをいう。

**参考文献** [1] 奥野信幸, 「SBI21 におけるオープン環境を利用した開発基盤」, Unisys 技報, 日本ユニシス, Vol.19 No.1 通巻 61 号, 1999 年 5 月

**執筆者紹介** 山崎 明彦 (Akihiko Yamasaki)

1991 年日本ユニシス(株)入社。共同センタ加盟信用金庫の補完システム開発を経て、SBI21/DE の開発・適用、及び MIDMOST/DE の開発・適用に従事し、現在、金融アウトソーシングセンター S-BITS 基盤統括 P 第二基盤 P に所属。