

COBOL 利用技術のご紹介

COBOL コンソーシアム利用技術分科会

柳川 哲也 (日本アイ・ビー・エム株式会社 ソフトウェア開発研究所)

第 5 回 COBOL 開発環境

1. COBOL 開発環境

開発生産性を向上させたり、アプリケーションの品質を高めるためには、優れた開発環境を利用することが重要です。COBOL アプリケーションの設計からプログラミング、実行、テスト、デバッグ、保守のあらゆる開発局面で作業効率を向上させる多くのツールが各ベンダーから提供されています。

開発支援ツールの中核となるエディター、デバッガー、プロジェクト管理といった機能は、コンパイラを含めた統合開発環境として統合され、連携して機能することによって、より開発生産性を向上させるものとなっています。また、最近のオープン系 COBOL 開発環境では、GUI 画面構築ツールを標準的に組み込むようになってきました。操作性の良い Windows 上の統合開発環境は、Windows 以外にもメインフレームや UNIX をターゲットにした開発までも可能にしています。

COBOL アプリケーションの開発のためには、要求定義やアプリケーション設計を行う上流設計ツールから影響分析ツールまで幅広い種類のツールが提供されていますが、ここでは、COBOL 言語に依存したサポートを行う統合開発環境に含まれるツールを中心に紹介します。

2. COBOL 用開発ツール

各ベンダーから提供されている開発ツールで多少の機能の差異はありますが、多くのベンダーでは、COBOL に特化した同様の機能を提供しています。以下に、各ベンダーの COBOL 開発環境の中から IBM VisualAge COBOL を例として用い、開発ツールの中核となる設計ツール、COBOL エディター、テスト支援ツール、デバッガー、プロジェクト管理機能について説明します。

2.1 設計ツール

COBOL アプリケーションで使用するファイル/レコード定義、画面定義、帳票定義を簡単に構築できる多数のツールが提供されています。これらのツールは定義を直感的に判りやすい GUI で作成できるようになっています。作成した定義から COBOL コードに自動変換したり、COBOL から呼び出し可能なサブ・プログラムを作成したりすることによってアプリ

ケーションから使用可能となります。

開発者は操作性の良い GUI で対話的に、アプリケーション利用者の実イメージのまま設計・開発が行えるので、開発効率が非常に良くなります。

「COBOL 利用技術のご紹介 第 2 回- GUI」に画面作成の例がありますので、詳しくはそちらをご覧ください。

2.2 COBOL エディター

統合開発環境では、COBOL 言語依存の機能として、COBOL 予約語や定数、標識領域 / A 領域 / B 領域のカラムによる色分け表示機能、COBOL 言語に依存したタブ / インデント機能を実現した COBOL エディターが提供されており、ソース編集を効率良くできるようになっています。エディターの COBOL 解析機能を利用することにより、コンパイルする前に不注意なエラーをある程度無くすることができますので、効率的なプログラミングが可能となります。

図 1 は、COBOL エディターの例です。COBOL 文法上に問題がある部分は、色分けやエラー・メッセージによって開発者がエラーだと分かるようになっています。例えば、リテラルの終わりの区切り文字が見つからない場合には、編集途中にエラー・メッセージが出力されます。また、DISPLAY 命令を記述する場合に、"DISLPAY" とスペル間違いをしても、COBOL 予約語の色付けがされないので、開発者は直ぐにスペル間違いがあることに気付いて修正することができます。

図 1 COBOL エディター

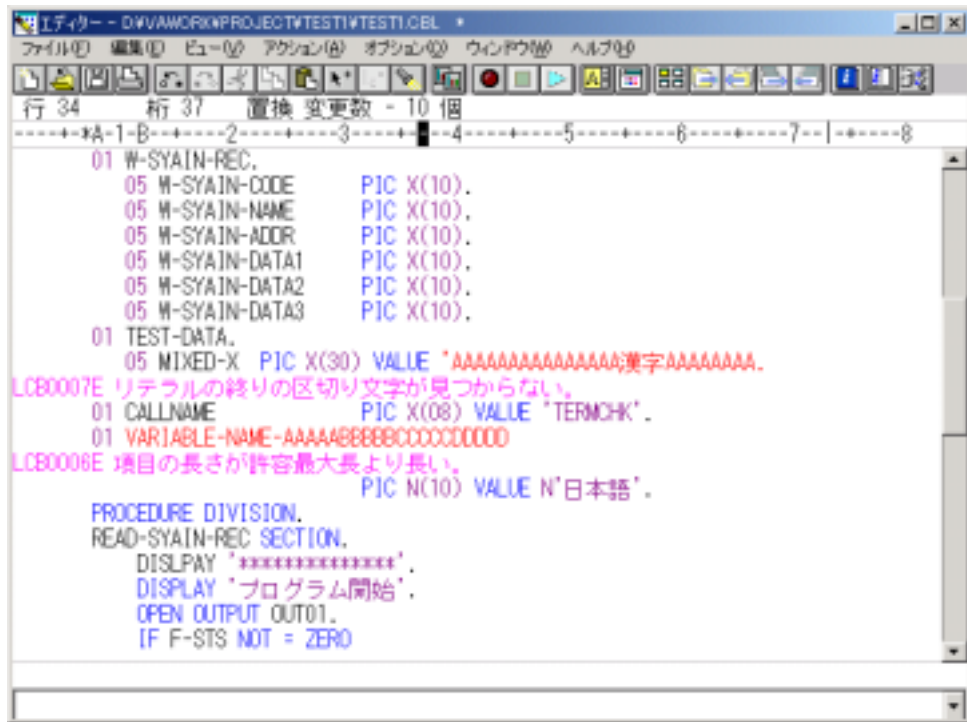
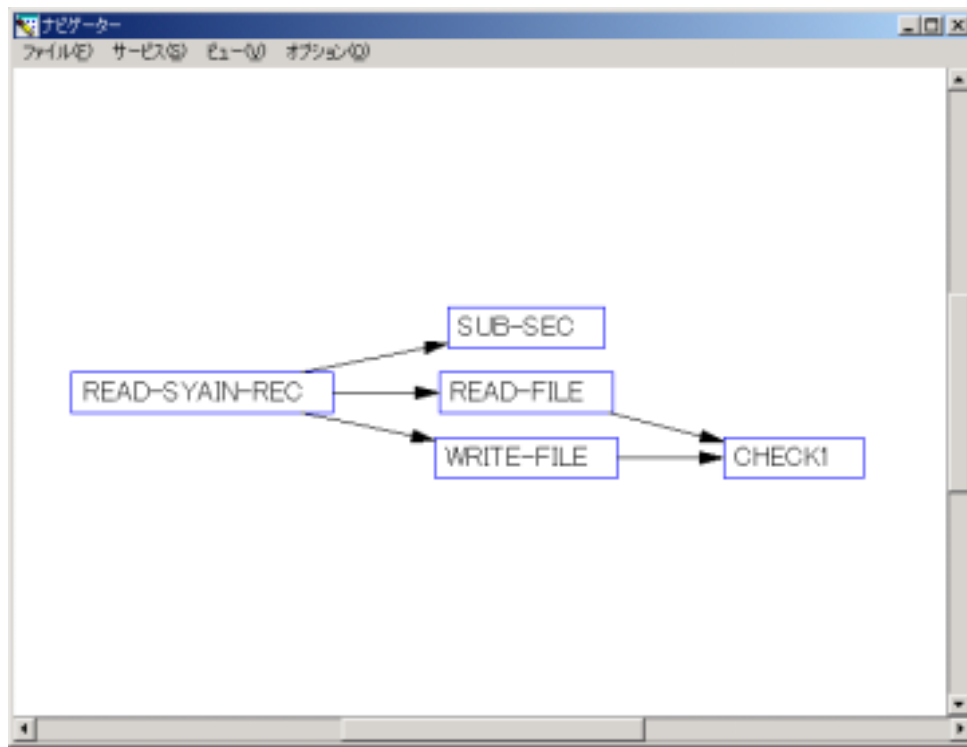


図 2 は、エディターで表示されるモジュール構造図の例です。プログラミングを行う場合、プログラム全体の概要構造を把握する必要性がでてきます。例えば、他のプログラマーが作成したプログラムの保守を行う場合に、プログラム全体の構造を知りたい時があります。視覚的に全体構造を把握できれば、プログラム構造の理解がし易くなります。

図 2 モジュール構造図



また、その他にも COBOL に特有な機能として、COPY 句ファイルの展開表示機能、行番号の自動割り振り機能、COBOL 構造を意識した畳み込み機能などが殆どの COBOL エディターで提供されています。

2.3 テスト支援ツール

テストには単体テストやシステムテストなどがありますが、COBOL 用開発ツールとしてはビジネスロジックの単体テスト支援ツールを提供しているものがあります。

単体テスト支援ツールは、メイン/サブ・プログラム、ファイル、データベースなどの呼び出しシミュレーション機能、テストデータの作成支援、実行パス解析などの機能をサポートし、COBOL 単体プログラムの品質を高めることができます。

単体プログラムのテストの品質は重要です。なぜなら、単体プログラムの品質が十分でないと、後のシステムテストにも影響がでてきます。システムテストで見つかった単体プログラムの不具合は、生産性を低下させる原因となります。単体テストの必要性は認識されていますが、実行するための手間がかかるために現実には行われていないことが多くあります。支援ツールを使用することにより、単体テストの実施が可能になります。

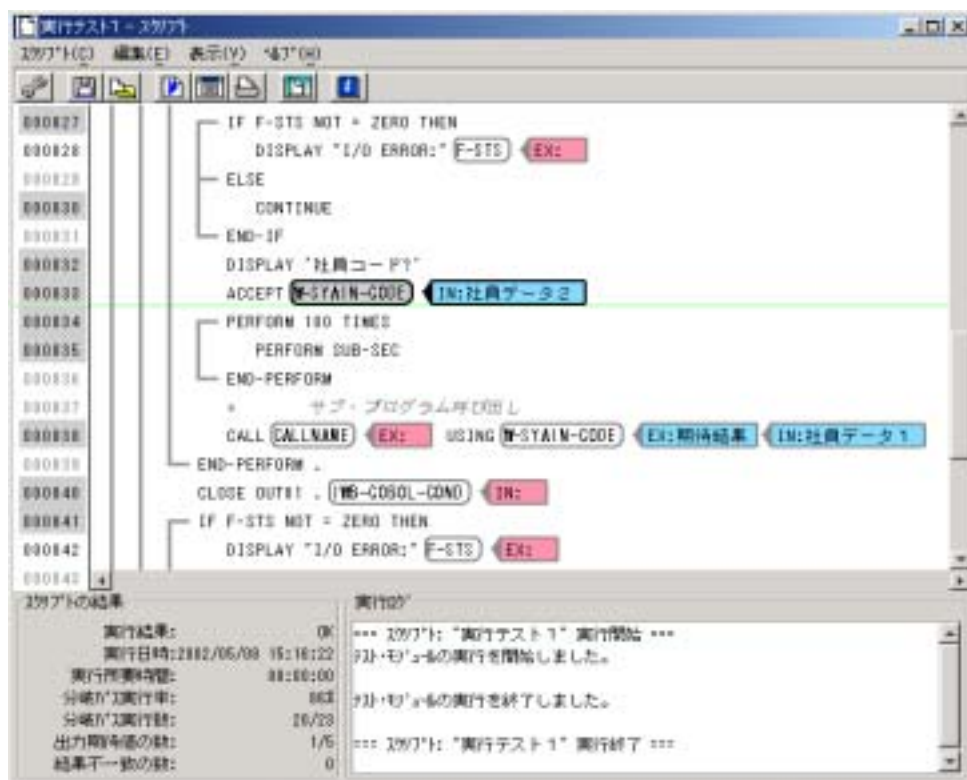
図 3 は、単体テスト支援ツールの例です。サブ・プログラムを CALL しているプログラム

のテストを行う場合、まだサブ・プログラムが開発されていなかったら、ダミーのサブ・プログラムを作成してパラメーターを設定しないと、呼び出し元のプログラムは実行できません。単体テスト支援ツールは、サブ・プログラムをスタブとして自動的に作成し、設定されたパラメーターを使用して、CALL 文をシミュレートします。このシミュレーション機能により、サブ・プログラムがない場合でも呼び出し元のプログラムの単体テストが可能になります。

シミュレーション機能は、呼び出しプログラムの他にファイルやデータベースをアクセスするプログラムの単体テストにも利用可能です。特に大量のテストデータを用意してバッチ実行の単体テストを行う場合に効果を発揮します。テストデータは GUI 画面で簡単に作成することができます。CSV ファイルもサポートされていますので、表計算ツールで大量にテストデータを作成して利用することが可能です。

また、単体テスト支援ツールはシミュレーション機能だけでなく、パス・カバレッジ機能も提供しています。プログラムは通常、多岐にわたる分岐条件文を含んでいます。この場合、様々なテストデータを与え、全てのパスの実行テストを行うことが重要です。支援ツールを使えば、容易にテストデータを作成し、様々なテストデータを与えた場合の実行パスを解析し、実行されていないパスを見つけることができます。

図 3 テスト支援ツール



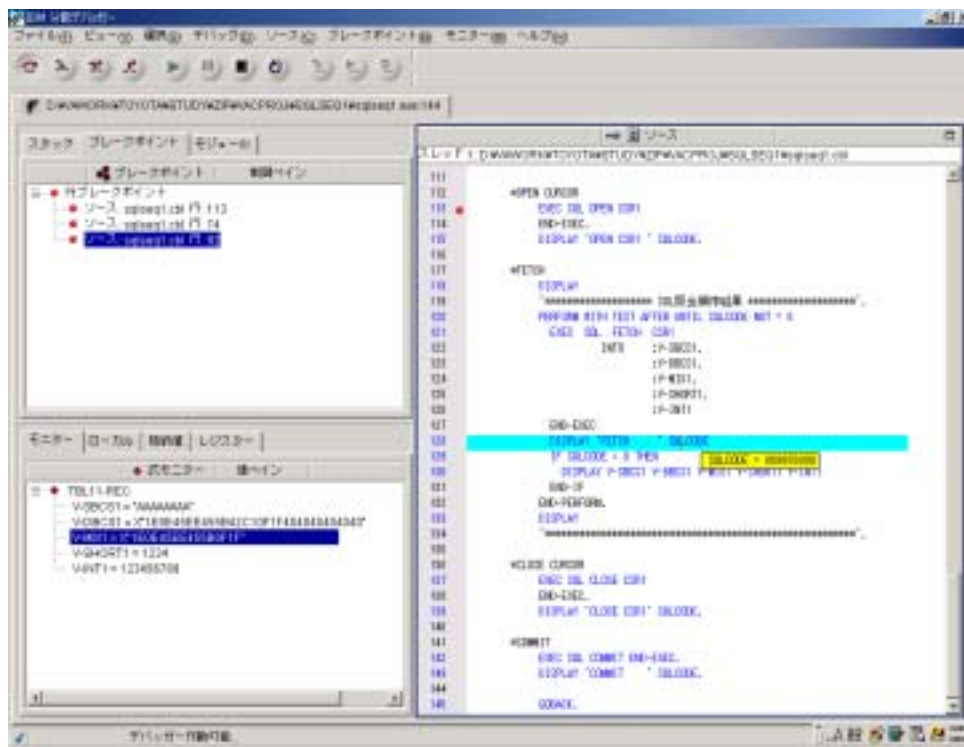
2.4 デバッガー

統合開発環境には、各種条件付きブレークポイントの設定、データ項目の値の参照 / 変更などが可能な対話型ソース・コード・デバッガーが提供されています。

組み込み SQL 文などを含むプログラムは、プリプロセス処理を行ってから COBOL のコンパイルを行う必要があります。以前は、プリプロセス処理後の元のソースとは異なったソースでデバッグしなければいけませんでした。しかし、最近ではコンパイラにプリプロセッサ機能を組み込んで、元の組み込み SQL 文のままデバッグできるようになっています。

図 4 は、デバッガーの例です。1 ステップずつの実行やブレークポイントまでの実行制御やデータ項目のモニターが可能です。メインフレームをターゲットにした開発の場合、COBOL ソースを Windows 環境にダウンロード、あるいは Windows 環境に COBOL ソースを新規作成して、Windows の直感的な GUI 機能を利用して容易にデバッグが可能となります。ツールによっては、ホストで稼動する COBOL アプリケーションをホストと通信しながら Windows の GUI 環境でデバッグできるものもあります。

図 4 デバッガー



2.5 プロジェクト管理機能

COBOL 言語に限らず、COBOL ソース/COPY 句などのソースコードの変更履歴の保持、複数のユーザーが同時に編集をすることを防ぐ排他制御の機能は、資産を管理する上で重要です。これらの管理機能を実現するソース管理ツールは、統合開発環境に組み込まれて利用できるようになっています。

各機能は連携しており、例えば、コンパイルを行ってコンパイル・エラーが発生した場合には、ウィンドウの下部に出力されているコンパイル・エラーメッセージをクリックするだけで、自動的にエディターが起動され、ソースコードのエラー行にジャンプすることができます。

[illegible]

3. まとめ

上記に説明した COBOL 開発環境の機能によって、開発者は高い生産性をあげ、品質の高いアプリケーションを開発することが可能となります。

さらに、最近の傾向としては、Web コンピューティング、オブジェクト指向技術への対応、Java などの他言語のラッピングツールなど最先端の技術を開発環境に取り入れて、益々、充実した開発環境になっていこうとしています。

なお、実装されているツールの種類や機能は各ベンダーの COBOL 開発環境によって異なりますので、利用の際には確認してください。