



Message from V Java Vendors

コンポーネントウェアで実現するリッチプロトタイピングツール「SMART PP」 第2回：MZ Platformのコンポーネントウェアとしての特長

テキスト=鈴木 聡平
text by Souhei SUZUKI

第1回では、SMART PPや、そのコア技術であるMZ Platformの成り立ち・概要について説明しました。今回は、MZ Platformのコンポーネントウェアとしての特長を解説します。なお、今回以降は、読者の皆さんがMZ Platformを利用可能だという前提で、実際の操作方法を説明します。プログラムは、<http://www.gihyo.co.jp/ad/download/>から入手できます。無料で利用できますので、ぜひインストールした上で、続きをご覧ください。



完全なコンポーネントウェアとは

コンポーネントベースの開発の必要性はしばしば話題に上ります。しかし、Java開発支援ツールはいろいろあるものの、完全なコンポーネントベースの開発ツールは少なく、「コンポーネントを利用する」という場合には、「プログラム上からライブラリとして呼び出す」という使い方がほとんどではないでしょうか。そうした使用方法には良い面もありますが、実際には、コンポーネントウェアとしての利点を最大限に活用できていないのが実情です。

そもそもコンポーネントウェアとは、再利用可能なオブジェクトの集合体である「ソフトウェア部品(コンポーネント)」を組み合わせることで開発を進める手法を取り入れたツールを指します。実行基盤や開発環境の共通化により、

- ①開発効率の向上
- ②アプリケーション開発の汎用性・適応性向上
- ③実績のあるコンポーネントを利用することによるア

プリケーション開発の信頼性向上

などを享受できることが利点です。

「コンポーネントベース」「コンポーネントウェア」という売り文句で販売されているツール類は、単なる開発ツールの延長であることが多く、「プログラミングをサポートする」という考え方からくる制約によって、オブジェクトとコンポーネントを混同しがちです。その結果コンポーネントウェアの本来の利点が得られないことも多いようです。

今後、コンポーネントウェアはそういった問題点を解決し、利点・長所を生かすツールとして、より追求されて行くと思います。MZ Platformも、そうしたコンポーネントウェアとしての利点・長所を追求したツールの1つであると言えるでしょう。

では、実際にMZ Platformを利用したアプリケーションを開発してみましょう。容易に開発・カスタマイズができることを経験していただき、MZ Platformがコンポーネントウェアとしての利点を生かしたツールであることを実感していただけたらと思います。

Webサイトから入手し、インストールしたMZ Platformを利用して、簡単なアプリケーションをカスタマイズしてみましょう。今回は、チュートリアルにある、ガントチャートをカスタマイズします。実際の操作方法は図1に示しました。このとおりに操作してみてください。



独自の特長とコンポーネントの再利用性

テーブルを操作する機能を持つボタンも、コーディ

Message from Java Vendors

ングすることなく、非常に簡単なGUIの操作のみで追加することができます。MZ Platformのアプリケーション開発は、コンポーネントの接続関係（呼び出し関係）を定義します。互いに独立したコンポーネント間を動的に接続する仕組みで、各コンポーネント単位での保守性・再利用性を確保しています。

その接続はJavaBeansの規定に従い、転送イベントモデルを使用し、すべての連携はイベント発生をトリガにした処理起動によって行われています。そのイベントを管理、処理する基幹となる部分が「コンポーネントバス」と呼ばれるコンポーネントです。コンポーネントバスはあるコンポーネントからイベントを受け、他のコンポーネントの処理を起動するという処理フローで、コンポーネントの接続関係を動的に処理しています。定義した接続関係をXMLとして保存できるので、可搬性も高いと言えます。

作成したアプリケーションの実行についても、一般的な開発ツールのようにビルドする必要はなく、ボタンを押下することですぐに実行できます。コンポーネント間の接続関係をプログラムソース内に埋め込まず、実行時のデータとして管理することによって、動的に処理制御を変更することにより可能となっています。

さらに、作成したアプリケーションはシリアライズされたデータとして保存され、他のアプリケーションから以前開発したアプリケーションを呼び出すことも可能なため、再利用性も非常に高いと言えるでしょう。

業務の現場では、必要とする機能が随時変更になることはよくあります。従来であれば、一度開発が始まると対応が困難な仕様変更作業も、MZ Platformでは設計と開発作業を同時に行うことができるため、何度でも繰り返して仕様の確認ができます。また、そのようなアプリケーションをスパイラル的な開発作業で容

易に構築できるため、実際のアプリケーション開発のプロトタイプとして利用することも可能です。

今までのコンポーネントウェアは、「開発者サイド」から「開発（プログラミング・試験）をサポート」するツールでした。しかし、MZ Platformを基盤としたSMART PPは「ユーザサイド」から見て「必要なアプリケーションを作るサポート」をするツールであり、コンポーネントウェアとしての要件を、本質から捉えたツールであると言えます。

今回は、MZ Platformが実現する高い開発生産性について取り上げます。

図1 カスタマイズの操作手順

手順1 アプリケーションのロード

操作

PFBuilde.exeを起動する
AP_DATA¥Tutorial¥Lesson11（ガントチャート）.apiをロードする

結果

アプリケーションビルダー画面上に、Lesson11（ガントチャート）.apiとして保存されていたアプリケーションの接続関係が表示される

手順2 カスタマイズ前アプリケーションの実行

操作

実行ボタンを押下する

結果

ガントチャートとそのデータが入ったテーブルが表示される

手順3 テーブル行追加ボタン機能の追加

操作

新規にボタンコンポーネントを追加する
アクションイベントを定義する
テーブルコンポーネントを接続コンポーネントに定義する
起動メソッドに「行を追加する」を選択する

手順4 画面へ手順3で追加した行追加ボタンを配備

操作

画面編集ダイアログを起動する
手順3で追加した行追加ボタンを貼り付ける

手順5 実行

操作

実行ボタン押下する

結果

テーブルへ行追加ボタンが表示される
で追加したボタンを押下すると、テーブルに行が追加される