

Message from 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3回：MZ Platformが実現する高い開発生産性

テキスト=柴戸 純也
text by Jyunya SHIBATO

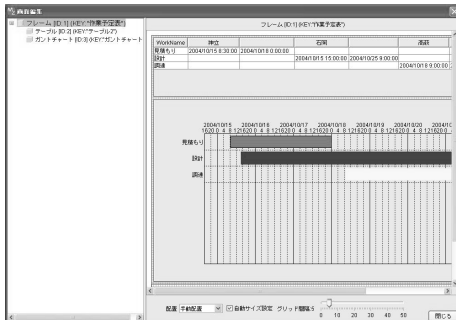
この連載では、スマーテックが提供するJavaアプリケーション開発支援プラットフォーム「SMART PP」と、コアコンポーネントである「MZ Platform」について紹介してきました。最終回となる今回は、実際の開発現場において、MZ Platformが実現する高い開発生産性について取り上げます。MZ Platformの無償トライアル版は、Webサイト (<http://www.gihyo.co.jp/ad/download/>) からダウンロードが可能です。



手間のかかるGUI構築が容易に

Javaにおいて、GUIの構築にはコツが必要であり、経験があってもプログラムは面倒なものです。たとえば、レイアウトマネージャや表示位置の設定などの作業において、レイアウトを修正しては、コンパイル、実行を繰り返すという苦勞をされた方も多いのではないでしょうか？ MZ PlatformのGUI構築機能である「画面編集機能」(図1)を使えば、こうした手間は、ほぼ不要になります。MZ Platform上でのGUI構築であれば、次に示す作業をマウス操作のみで実現できます。

図1 画面編集機能を使用した編集画面。メイン画面の“画面編集”ボタンを押下することにより表示される



- GUI部品の選択
- GUI部品の配置の設定
- GUI部品のレイアウトマネージャの設定 (GridBagLayout, FlowLayoutなど)
- GUI部品の属性の設定

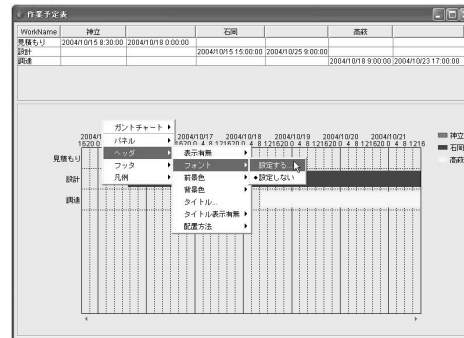
さらに、MZ Platformでは、作成したGUIアプリケーションを実行しながら、色やフォントなどの属性設定が可能です。これにより、本番実行時の形で動作確認を行いつつ、アプリケーションの見た目を編集することが可能となっています(図2)。



150を超える豊富な標準コンポーネント

MZ Platformに用意されている標準コンポーネントは、150を超えています。GUIコンポーネント(パネル、ボタンなど)を中心に、Javaアプリケーションで頻繁に使用されるCSV、DB、XMLなどのリソースに対する入出力コンポーネント、それらのデータを元にして作成可能な図形コンポーネント(ガントチャート、折れ線グラフなど)、さらに制御系コンポーネント

図2 実行時の属性設定画面。メイン画面の“実行(設定可)”ボタンを押下することにより表示される



Message from Java Vendors

(for文, if文など)などが用意されています。ほとんどのJavaアプリケーションを作る上で必要な基本コンポーネントを網羅していると言ってもよいでしょう。もちろん、必要なコンポーネントが存在しない場合には、新規にコンポーネントを作成し、それをMZ Platform上で再利用することも簡単にできます。

また、通常のJavaアプリケーション開発では、各種リソースに対して入出力を行う場合など、面倒な処理(XMLであれば、DTDの指定、ファクトリの生成、DOMツリーの生成など)をプログラミングしなければなりません。MZ Platformが提供するコンポーネントでは、それらの面倒な手続きが隠蔽されており、作業が大幅に簡略化されます。

MZ Platformでは、個々のコンポーネントに対するマニュアルを用意しています。製品に含まれるマニュアルを参照することで、コンポーネントの仕様(公開インタフェースなど)を容易に理解することができます。



コンポーネントの高い再利用性

コンポーネントの再利用は、今日のシステム開発では、何らかの形で、ほぼすべての開発者が行っています。しかし、実際には、インタフェースなど、対象のアプリケーションに依存するコンポーネントを作成する必要があるなどの理由で、コンポーネント自体に手を加えることが多いようです。つまり、再利用するためのプログラミングが必要となってしまいます。

第2回の「MZ Platformの特徴とコンポーネントの再利用性」でも少し触れましたが、MZ Platformにおいて、プログラムによる作りこみが必要なのは、新規コンポーネントのみであり、アプリケーションの構築はコンポーネントの組み合わせだけで行われます。したがって、新規に作成するコンポーネントの仕様(実装)がある特定のアプリケーションの仕様に引きずられて、そのアプリケーション以外には利用できなくなるといった事態を回避しやすくなります。その結果、コンポーネントの再利用性を高めることとなります。

もちろん、一度作成したアプリケーションを1つのコンポーネントとして再利用することもでき、アプリケーション同士の組み合わせも容易に実現できます。

こうした特徴によって、アプリケーションの開発・試験において、完全な差分開発が可能となり、開発生産性を高めることができます。

MZ Platformは、コンポーネントの利用方法などに関するマニュアルを用意しています。製品に含まれるマニュアルを参照することで、コンポーネントの利用、作成方法、アプリケーションのコンポーネント化に関する方法などを容易に理解できます。



アプリケーション品質の向上

一般的な開発では、アプリケーションに対する試験を行う場合、使用する部品(コンポーネント)に対する試験(単体試験)、そして、それらを組み合わせたものに対する試験(結合試験)を行います。MZ Platformでは、十分に品質が保たれた標準コンポーネントの組み合わせでアプリケーションを構築するため、試験としては、コンポーネントの接続部分のみに注力することが可能になります。これにより、試験においても十分な時間の確保ができ、アプリケーションの品質の向上に十分に貢献できると言えます。

もちろん、新規に作成したコンポーネントを利用する場合も、十分に品質を高めて、財産として蓄えておくことで、同様のことが言えます。



まとめ

これまで3回にわたって、MZ Platformを紹介してきました。これらの特徴により「開発生産性」「品質」の向上に大きな貢献ができることをご理解いただけたと思います。また、Javaに対する専門知識のない方でも、容易にJavaアプリケーション開発ができるため、個々の技術力の差を十分に吸収でき、職人技に頼ることもなくなります。

また、MZ Platformは、開発だけでなく、要件定義や設計時のプロトタイプの開発にも最適です。早い段階で完成物のイメージが共有できるだけでなく、事前に、さまざまな部門の意見を収集し、以降のフェーズに活かすことが可能です。さまざまな開発フェーズにMZ Platformを適用することで、各フェーズ同士をシームレスにつなぎ、開発全体としての生産性と品質を大幅に向上することができるでしょう。☞