

5. JavaBeansとコンポーネントウェア

山口 浩

日本サン・マイクロシステムズ(株)

はじめに

ソフトウェアを通常の工業製品と同様、いくつかの部品を組み合わせることで構築することはソフトウェア工学における大きな課題であった。これに対する試みは、古くはFortranで使用する科学技術計算ライブラリのようなサブルーチンの集合として現れたが、その後オブジェクト指向プログラミングが登場した1980年代にはさまざまなクラスライブラリという形になり、さらにウィンドウ環境が一般的になった1990年頃からは、GUIビルダーと呼ばれる開発ツールで使用する部品^{*}として現れるようになった。これらの部品は、当初は個々のGUIビルダーに依存した設計であったが、しだいに部品間のインタフェースが標準化されるようになり、いくつかの業界標準も登場するようになってきた。コンポーネントウェア^{**}はこのようなソフトウェア部品を中心とした部品の組み立てによって、ソフトウェアを構築する技術を指す言葉として用いられている¹⁾。一方、Java言語は1995年に発表されて以来、その最大の特徴であるプラットフォーム非依存性によりさまざまな分野で注目を浴びてきたが、特にソフトウェア部品をJavaによって作成するというアイデアは、JavaBeansという形で急速に実現されることとなった。本稿では、JavaBeansを用いたアプリケーション開発とその背景にある技術を紹介するとともに、現在開発作業が行われている新しい機能についても概説する。

JavaBeansを用いたアプリケーション開発

JavaBeansの仕様書²⁾では、JavaBeans (Beans)を「再利用可能なソフトウェア・コンポーネントでありビルダーツールによって視覚的に操作できるもの」と定義している。これはJavaBeansがGUIビルダーのようなビルダーツールを利用してアプリケーションの作成を行うユーザを対象としたソフトウェア部品であり、利用にあたってはJavaのプログラミング言語としての知識は仮定しないことを前提としている。また、JavaBeansはビルダーツールによって視覚的に操作できる必要はあるが、それ自体は必ずしも目に見える

部品でなくてもかまわない点にも注意しておく。上述の仕様書では、JavaのプログラムがJavaBeansであるための条件を、実装すべきクラス、メソッド、あるいは最終的な部品形態としてのパッケージング形式等について定義しており、これに従って作成されたJavaBeansはJavaをサポートするすべてのプラットフォーム上でビルダーツールを用いて自由に組み合わせることが可能となる。

すでに、各社からさまざまなJavaBeans対応のビルダーツールが登場しているが³⁾、図-1に米国サン・マイクロシステムズが開発したビルダーツールJavaStudio⁸⁾を例として示す。

JavaStudioのメインウィンドウには個々の部品(Beans)に対応したアイコンがいくつかのカテゴリに分類されて並んでおり、ユーザは必要な部品をここからGUIウィンドウにマウスを使ったドラッグアンドドロップで配置することによりアプリケーションを作成する。この時、デザインウィンドウには、これらの部品に対応するアイコンが表示されており、マウス操作でそれらの間にリンクを張ることで部品間に処理の流れ等の論理的な関係を定義することができる。

あらかじめ用意されている部品としては、テキストを表示するラベル、入力フィールド、ボタンといったGUIの基本部品に加え、マルチメディア関連やデータベース操作を行うもの、あるいはループや条件判断、データの分岐を行うような「制御」のための部品など、約50種類がある。さらに、ユーザが作成したり購入したりしたBeansを新たな部品として追加登録することも行えるようになっている。これらの操作は、マウスによる簡単な操作や適当なテキストの入力を行うだけで実行でき、プログラミング言語に関する知識はほとんど必要としない。実際にJavaStudioを用いてアプリケーションの作成を行ってみると、それが電子機器の回路図を書き上げる作業に類似していることに気付くであろう。しかもこうして作成したプログラムは、JavaアプレットとしてさまざまなOS上のWWWブラウザで動かしたり、あるいはスタンドアロン型のアプリケーションとして、さまざまなJava環境で動作させたりすることができる。さらにこれを新しいBeanとして部品化することも可能である。このような特徴は従来のGUIビルダーには見られないものであり、JavaBeansの持つ大きな特徴でもある。

^{*} ボタンやラベル等の部品でウィジェットあるいはガジェットと呼ばれることもある。

^{**} コンポーネント・ソフトウェアとも呼ばれる。ほかにIBM社、Apple Computer社によるOpenDocやMicrosoft社によるOLE/COMなどが知られている。

³⁾ JavaBeans対応の開発ツールについては、<http://java.sun.com/beans/tools.html>に一覧がある。

JavaBeansで用いられるコンポーネントウェアとしての基盤技術

JavaBeans を特徴づける重要な属性は「プロパティ」、「メソッド」、「イベント」である。「プロパティ」はBeansの公開された変数であり、適当なメソッドを呼び出すことで、参照や変更が可能な属性である。「メソッド」は通常のJavaのパブリックなメソッドであるが、Beansとして使用するものをこれらのメソッドの一部に限定することも可能である。最後に「イベント」はBeans間の通信に用いられる仕掛けであり、1つのコンポーネントで、ある事象が発生したことを、他のコンポーネントに知らせるための通信手段として提供されている。事象が発生するイベントソースに、その事象に関心を持つコンポーネントがリスナーオブジェクトを登録しておき、事象の発生時にそのオブジェクトの適当なメソッドが呼び出される仕組み（デリゲーションイベントモデル）によって実現されている。

さて、ビルダーツールは個々のBeansが持つさまざまな属性を知るために「イントロスペクション」と呼ばれる機能を用いる。これはJavaのCore Reflectionを用いた機能で、Javaのクラスファイルに定義されている変数やメソッドのシグネチャを調べたり、あるいは指定したメソッドを適当な引数とともに実行したりする一種のメタプログラミングを行う強力な仕掛けである。さらに、ユーザがこれらのBeansの属性を変更するためのインタフェースはCustomizer APIとして用意されているが、Beansの提供者はビルダーツールのためにカスタマイザを書くことで、カスタマイズのための優れたユーザインタフェースを実現することができる。一般にカスタマイズを行った結果は、あるクラスのインスタンスとなるが、これをそのまま保存する仕組みも提供されている。これにはオブジェクトのシリアライゼーションと呼ばれる機能が使われている。これらは、いずれもJDK1.1で導入された一般的な機能であるが、JavaBeansの誕生にはこれらのさまざまな機能をうまく組み合わせることが必須であったことは容易に想像されるであろう。

JavaBeansの新しい機能

現在のJavaBeansの仕様に基づくBeans開発ツールやビルダーツールは、ほとんどが単体で動くJavaの

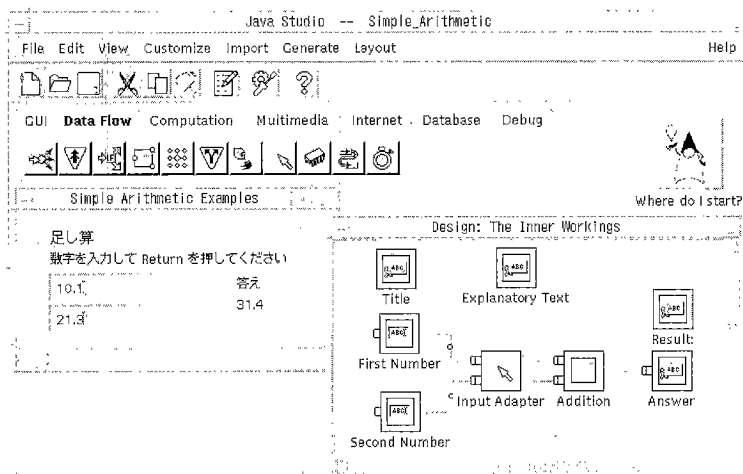


図-1 JavaStudioによるアプリケーション開発

表-1 JavaBeans 関連の拡張機能

新しい仕様	内容
Enterprise JavaBeans ³⁾	C/S システムにおいてサーバ側で動作するJava プログラムを作成するために拡張された JavaBeans 仕様
Drag and Drop ⁴⁾	Java アプリケーション間および通常のアプリケーションとのドラッグアンドドロップ操作
Activation Framework ⁵⁾	データソースの型や処理方式の決定方法の一般化
Runtime Containment ⁶⁾	JavaBeans が自分自身を含むコンテナに関する情報を実行時に得るためのインタフェース
InfoBus ⁷⁾	コンテナ内の複数のJavaBeans が相互にデータ交換を行うためバスアーキテクチャ

世界に閉じたアプリケーションのためのものであるが、他のコンポーネントウェアとの接続やBeans間のより高いレベルの結合、あるいは分散環境への対応といった多くの課題が残されており、現在もさまざまな機能拡張のための作業が進行中である。これらのうちから主要なものを表-1に紹介する。今年の夏頃にリリースが予定されているJDK 1.2では、GUI機能の拡張やサーバ環境での利用等を踏まえた機能拡張が行われるが、表-1のうちのいくつかは基本機能として盛り込まれる予定である。JavaBeansの大きな目的の1つは、企業内システムで用いられるような複数のプラットフォームで動作する大規模なアプリケーション開発をBeansベースで行うことであるが、これらの機能を盛り込むことで、それが現実のものとなる日もそう遠くはないと考えている。

参考文献

- 1) 青山幹雄: コンポーネントウェアの挑戦, bit, Vol.28 (Mar., Apr. 1996).
- 2) JavaSoft: JavaBeans 1.01 (July 1997).
- 3) JavaSoft: Enterprise JavaBeans Version 0.9 (Feb. 1998).
- 4) JavaSoft: Proposal for a Drag and Drop subsystem for the Java Foundation Classes (Dec. 1997).
- 5) JavaSoft: JavaBeans Activation Framework Specification (Dec. 1997).
- 6) JavaSoft: A Draft Proposal to define an Extensible Runtime Containment and Services Protocol for JavaBeans (Version 0.98) (Jan. 1998).
- 7) JavaSoft: InfoBus 1.1 Specification, Draft 6 (Dec. 1997).
- 8) SunSoft: JavaStudio 1.0, <http://www.sun.co.jp/workshop/js/> 文献2) ~7) は米国サン・マイクロシステムズのJavaホームページ (<http://java.sun.com>) からダウンロード可能である。

(平成10年2月27日受付)