

XMLを用いたアプリケーションの構築法 —Java Beansによるフレームワーク—

八木下 和代¹ 浦本 直彦¹ 丸山 宏²

¹日本アイ・ビー・エム(株) 東京基礎研究所

²日本アイ・ビー・エム(株) 東京基礎研究所 および 東京工業大学情報理工学研究所

1 はじめに

インターネット上で文書やデータをやり取りする際に利用される文書記述言語として、XML(Extensible Markup Language)がある[1][2]。XMLではタグを定義しデータ構造を記述することができる。またXMLプロセッサを使ってXML文書をアプリケーションが自動的に解析/生成することも可能である[3]。その結果、ネットワーク上に散在するWebサイトやデータベースなどから得られる結果を組み合わせたWebアプリケーションを、容易に作成することも可能となる。

またソフトウェアの生産性を向上させるためには、ソフトウェアを部品化し、再利用性を高めることが効果的である[4]。よってWebアプリケーションを作成する際にも、ソフトウェアコンポーネントを利用することは極めて有用である。

そこで我々は、Java言語によるソフトウェアコンポーネントのためのフレームワークであるJava Beansを用いてXML文書を扱うためのいくつかのコンポーネントを作成し、それらを組み合わせた簡単なWebアプリケーションを実装した。

本報告では、Java Beansによるコンポーネントの設計方針を提案し、実際に必要となってくると考えられるBeanに関する検討を行う。

2 コンポーネントの設計方針

2.1 何故Java Beansなのか

Java Beansを含めて、一般にソフトウェアコンポーネントの仕組みは、GUIを簡単に作るための仕組みとして捉えられがちである。一方、ユーザーインターフェースとして対応するグラフィックスを持たないBeanは不可視Beanと呼ばれ、通常の画面に見えるBeanと区別される。このような不可視Beanの利用は、APIドキュメントを参照せずにアプリケーションプログラムの開発が可能となることから、GUIを持たないサーバーアプリケーションの作成に大きな威力を発揮する。また不可視Beanをビジュアル

の作成に大きな威力を発揮する。また不可視Beanをビジュアルビルダー上で用いることによって、プログラム中のデータの流れを自然に表現することも可能となる。すなわち、入力として受け取ったデータに何らかの処理を加えて出力するようなコードを、Java Beansのビジュアルビルダーを用いて視覚的に容易に生成することが可能となるのである。XML文書を扱うアプリケーションでは、XML-XML間、あるいはXML-他のフォーマット間の変換を繰り返し行うことによって最終的な結果を得ることが多い。よって、XML文書を扱うアプリケーションの開発に不可視Beanを用いることは非常に有効である。

2.2 フレームワーク

XML文書のためのBeanを開発する際のフレームワークとして、IBMのXML Productivity Kit for Java(以下XPK4J)がある[5]。XPK4Jはパイプライン方式を採用し、各Beanは他のBeanからXML文書を受け取り、処理後のXML文書を他のBeanに引き渡す。最終的な結果はXML文書がいくつかのBeanを経ることによって得られる。ここで、Bean間のXML文書の受け渡し方法をどうするか、言い換えるとBeanにどのようなフィーチャー(プロパティ、メソッド、イベント)を定義するかを決定することが重要となる。本報告では以下の2つの手法を提案する。

(1) event-listener 方式

XPK4Jで採用されている方式である。各BeanはListenerインターフェースを実装し、自分自身を他のBeanで起こるイベントのリスナーとして登録することによって接続される。このイベントは、結果として出力されるXML文書が生成された時点で起こるものとして定義され、生成されたXML文書はこのイベント中に埋め込まれて他のBeanに引き渡される。

(2) event-to-method 方式

本方式では、あるBeanのイベントと別のBeanのプロパティあるいはメソッドとがアダプタコードを介して接続される。イベント中に含まれているXML文書は、接続されたプロパティ自身あるいはメソッドの引数に対して引き渡される。本方式を用いれば、リスナーを実装していないBeanにもイベントを到達することができるため、より柔軟なプログラミングが可能と

なる。

2.3 Beanの提案

以上のような考えに基づいてXML文書処理するWebアプリケーションを作成する場合、以下のような機能を備えたBeanが必要となってくる。

XML文書変換機能

あるDTDに従って生成されたXML文書を、別のDTDに従うXML文書、あるいはHTMLなどXML以外のフォーマットに変換する。入力されたXML文書が、変換規則用のXML文書に基づいて変換され出力される。

データベースからの結果のXML化機能

データベースから返された結果をXML文書に変換する。

グループ化機能

指定されたKeyに基づきXML文書のDOM[6]を再構築する。入力されたXML文書が指定されたKeyおよびグループ化の適用範囲に基づいて変換され、出力される。

ソート機能

指定されたKeyに基づきXML文書のDOMを再構築する。入力されたXML文書が指定されたKeyおよびソーティングの適用範囲に基づいて変換され、出力される。

XML文書拡張機能

複数のXML文書を1つのXML文書に拡張する。入力された複数のXML文書がスケルトン用のXML文書に基づいて統合され、出力される。

デジタル署名機能

XML文書の全体またはその一部にデジタル署名を付加する。

暗号化機能

XML文書の全体またはその一部を暗号化する。

スイッチ機能

入力されたXML文書を他のBeanに転送するか否かを、指定された条件に基づいて決定する。複数のXML文書の中から条件に応じて必要なXML文書のみを選択する際に有用である。

またXPK4Jでは入力されたXML文書を解析するBeanなどが提供されている。以上のようなBeanのいくつかを組み合わせた簡単なアプリケーションを作成した。その際Java Beansの代表的なビジュアルビルダーの1つであるVisualAge for Javaを使用した。作成したアプリケーションのVisualAge for Javaにおけるビジュアルコンポジションを図1に示す。この図のように、XML文書を扱うためのアプリケーションを、ビジュアルビルダー上でBeanを視覚的に接続することによって容易に作成することが可能となる。

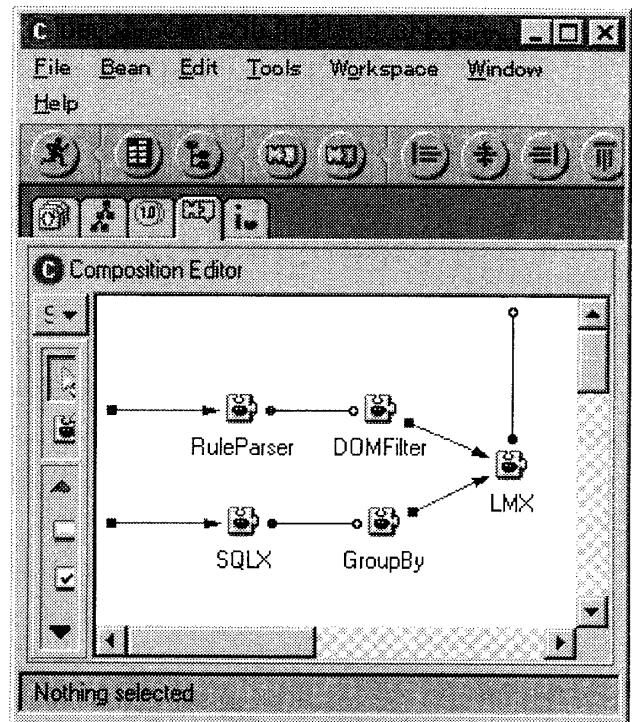


図1: VisualAge for Javaを用いたプログラミングの例

3 おわりに

本報告では、XMLとJava BeansによるWebアプリケーションの作成、およびそのために必要なBeanの設計方針や機能に関する提案を行った。今後は、Bean間の接続方式として先に述べたevent-listener方式とevent-to-method方式を比較して考察を深めるとともに、他の方式の可能性についても探っていきたいと考えている。またより多くのBeanを提案し、実装していく必要があると考えている。

参考文献

- [1] W3C. Extensible Markup Language (XML) 1.0. 1998. <http://www.w3.org/TR/1998/REC-xml-19980210>
- [2] XML/SGMLサロン. XML 完全解説. 記述評論社. 1998.
- [3] 田村健人. XMLプロセッサとその利用方法. 人工知能学会誌, Vol.13, No.4, 1998.
- [4] Erich, G., Richard, H., Ralph, J. and John, V. (著), 本位田真一, 吉田和樹(監訳). オブジェクト指向における再利用のためのデザインパターン. SOFTBANK. 1995.
- [5] IBM alphaWorks. XML Productivity Kit for Java. <http://www.alphaworks.ibm.com/formula/xmlproductivity>
- [6] W3C. Document Object Model (DOM) Level 1 Specification Version 1.0. 1998. <http://www.w3.org/TR/REC-DOM-Level-1/>