

XML による Java プログラムの API ドキュメント生成*

4M-5

中村 章人†
産業技術総合研究所‡

1 はじめに

Javadoc[2] は、Java™のソースファイルから HTML フォーマットの API ドキュメントを生成するツールである。Javadoc を利用することで、プログラマはコーディングと同時に API ドキュメントを容易に作成できる。また、API ドキュメントの出力を行う Doclet というプラグインプログラムを組み込むことで、出力をカスタマイズできる。

XML は、プラットフォーム、応用、言語等の制約を受けず、プログラム処理に適した、単一ソースで多利用が可能なフォーマットである。本稿では、Java の API ドキュメントを XML フォーマットで出力する XMLDoclet[5] について述べる。

2 XMLDoclet の設計

2.1 Javadoc のアーキテクチャ

Javadoc は、Java ソースファイルの宣言とコメントをパースして API 情報の内部表現を構築し、Doclet と呼ばれるプラグインにその情報を渡す [図 1]。Doclet は、Doclet API を通じて API 情報を取り出し、独自の方法で出力を行う。ユーザが Doclet を指定せずに Javadoc を実行すると、デフォルトの標準 Doclet が使用され、HTML フォーマットの API ドキュメントを生成する。これに対し、XMLDoclet は XML フォーマットで出力を行う。

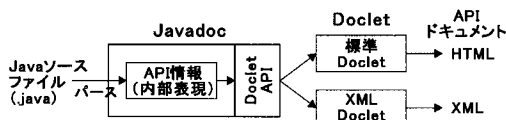


図 1: Javadoc のアーキテクチャ

コメント中に Javadoc 用のタグ [3] を埋め込むことで、コード部分から得られる宣言情報とユーザの説明文とを関連付ける (これをドキュメンテーションコメン

トと呼ぶ)。ドキュメンテーションコメントには HTML タグを使用できるため、説明文の中に表や箇条書き等の構造やリンクを記述できる。また、Javadoc で生成した既存の API ドキュメントへのリンクを張ることができる (これを外部参照と呼ぶ)。

2.2 設計方針

標準 Doclet が生成する API ドキュメントは、表示は容易だが、情報の再利用や加工は困難である。これは、HTML (のタグ) がデータモデルの表現に不向きであることと、標準 Doclet の目的が API 情報の表示のみであることによる。これを XML フォーマットとすることで、様々な応用プログラムから API 情報を再利用でき、加工やフォーマット変換が容易に行えるようになる。

XMLDoclet の設計方針を以下に示す。

- **ユーザ透過性:** 標準 Doclet で処理できるソースファイルは、XMLDoclet でも処理できる。
- **プログラム透過性:** XMLDoclet は、標準 Doclet が生成した API ドキュメントを外部参照できる。逆に、標準 Doclet は、XMLDoclet が生成した API ドキュメントを外部参照できる。
- **データモデルとビューの分離:** XMLDoclet が生成する XML は API 情報のデータモデルであり、そのビューはスタイルシートと応用プログラムによって提供する。

すなわち、プログラマは XMLDoclet 用に特別のドキュメンテーションコメントを記述する必要がなく (ユーザ透過性)、外部参照する API ドキュメントが標準 Doclet または XMLDoclet のいずれで生成されたかを意識する必要が無い (プログラム透過性)。また、データモデルとビューを分離することで、情報の再利用や加工と表示の制御を独立かつ容易に行える。

3 API ドキュメント

XMLDoclet は、API ドキュメントを以下の一連のファイルとして出力する。このファイル構成は、標準 Doclet の出力とほぼ同じである。

*Java API Documentation in XML

†Akihito NAKAMURA (nakamura-akihito@aist.go.jp)

‡National Institute of Advanced Industrial Science and Technology (AIST)

- **基本内容ファイル群:** API の説明を内容とする XML ファイル群。クラス毎に一つのクラスドキュメント、パッケージ毎に一つのパッケージドキュメント、全体で一つの概要ドキュメントがある。
- **相互参照ファイル群:** 基本内容ファイルの参照を便利にするための XML ファイル群。クラス階層ドキュメントと索引ドキュメントがある。
- **補助ファイル群:** API の説明には直接必要のないファイル群。外部参照機能の実装に必要なパッケージリスト(テキストフォーマット)と表示を制御するスタイルシートがある。

3.1 スキーマ

API 情報を再利用するためには、XML の要素と属性を定義したスキーマを公開する必要がある。API ドキュメントのスキーマ定義には DTD を用いた。ここでは DTD 自体は示さずに、図 2 にクラスドキュメントの例 (java.lang.Object) のみを示す。

```
<javadoc
  xmlns="http://www.aist.go.jp/2001/javadoc">
  <class name="Object" visibility="public">
    <containingPackage name="java.lang"
      href="../../java/lang/package-summary.xml"/>
    ...
    <method name="getClass" signature="()"
      visibility="public" modifier="final native">
      <return>
        <type name="Class" package="java.lang"
          href="../../java/lang/Class.xml"/>
        <description>
          <div xmlns="http://www.w3.org/1999/xhtml">
            the object of type <code>Class</code> ...
          </div></description>
        </return>
      </method>
      <description>
        <div xmlns="http://www.w3.org/1999/xhtml">
          Returns the runtime class of an object.
          That <tt>Class</tt> object is ...
        </div></description>
      </method>
      ...
    <see class="java.lang.Class"
      href="../../java/lang/Class.xml"/>
  </class>
</javadoc>
```

図 2: API ドキュメントの例 (抜粋)

3.2 表示

XMLDoclet 自身に API ドキュメントの表示機能はないが、表示を補助する機能を持つ。XML に XSL と CSS

の 2 種類のスタイルシートをリンクする機能と、デフォルトのスタイルシートを用意した。

デフォルトのスタイルシートは、XML を HTML に変換する XSLT[1] とその HTML にフォントや色等の書式指定を行う CSS である。この HTML は標準 Doclet が生成するものとほぼ同じ構造になるようにした。つまり、標準 Doclet の提供するビューをデフォルトとした。

標準 Doclet では、ユーザ指定の CSS ファイルを HTML にリンクできるが、書式指定をカスタマイズできるだけである。XMLDoclet ではデータモデルとビューを分離し、XSL と CSS を記述することでビューを柔軟にカスタマイズできる。また、XML プロセッサを利用して応用プログラムを記述すれば、データモデルの加工とビューのカスタマイズとを更に柔軟に行える。

4 実装

XMLDoclet の実装には Apache の Xerces 及び Xalan[4] を用いた。

4.1 プログラム透過性

標準 Doclet は、外部参照するファイルの拡張子を .html であると見なす。しかし、XMLDoclet が生成するファイルの拡張子は .xml であるため、そのままではプログラム透過性が実現できない。XMLDoclet は、標準 Doclet が生成するファイルと同じ名前の HTML ファイルも生成する。この HTML に本文はなく、meta タグと content 属性を利用して外部参照のリンクを XML ファイルにリダイレクトする。

5 おわりに

本稿では、Java のソースファイルから XML フォーマットの API ドキュメントを生成する XMLDoclet について述べた。これにより、API 情報の再利用や様々なビューの提供が容易になる。XMLDoclet のプログラム及びドキュメントは [5] で公開している。

参考文献

- [1] Clark, J.: *XSL Transformations (XSLT) Version 1.0*, W3C Recommendation 16 Nov. 1999. <http://www.w3.org/TR/xslt>
- [2] Sun Microsystems: *Javadoc Tool Home Page*. <http://java.sun.com/j2se/javadoc/>
- [3] Sun Microsystems: *How to Write Doc Comments for the Javadoc Tool*. <http://java.sun.com/j2se/javadoc/writingdoccomments/>
- [4] The Apache XML Project. <http://xml.apache.org/>
- [5] XMLDoclet Home Page. <http://staff.aist.go.jp/nakamura-akihito/xmldoclet/>