DOMインターフェースに基づくSGMLパーザー

4V－9

近藤 豪　広瀬 紳一　北山 文彦　久世 和資
日本アイ・ビー・エム株式会社 東京基礎研究所

1. はじめに

SGML(Standard Generalized Markup Language)は，電子出版や大規模文書処理のための国際標準言語として普及している．

一方，XMLやHTMLのオブジェクト・モデルの標準的な仕様としてDOM (Document Object Model) [1]が定められている．しかし，SGMLに対しては標準的なオブジェクト・モデルが存在しない．

そこで，記述力が高く汎用的なSGMLからDOMインターフェースを実装する構文木を生成するパーザーをJavaで開発した．本稿では，そのパーザーの設計と実装，及びアプリケーション構築を支援するDOMインターフェース実装ツールについて述べる．

2. DOM インターフェース

DOM仕様は，文書の内容や構造などをダイナミックにアクセス・更新するためのプラットホーム及びプログラミング言語に依存しないインターフェースを定義している．昨年，W3Cによって勧告された DOM Level 1 仕様では，HTMLおよびXMLの文書やデータを表現するためのオブジェクトのセット，オブジェクト群をどう組み合わせるかについてのモデル，これらをアクセス・操作するためのインターフェースの標準を提供している．

DOM Level 1 は２つの部分から成り立つ．１つはコア部である． コア部は，文書のオブジェクト・モデルに対するアクセスや操作の基本的なインターフェースの集合である．この中で重要な役割を果たすのが Document インターフェースである．これは文書自体を抽象化しているだけでなく，DOMのインスタンスを生成するファクトリの役目も兼ねている．

もう一つはHTML部である．HTML部の仕様はコア部のインターフェースを継承しており，HTMLのDOMに特化したものである．

DOM Level 1 仕様はHTMLとXMLのために定義されているが，XMLとSGMLの共通部分が多いことを考えれば，コア部を実装するSGMLのオブジェクト・モデルをパーザーが生成することは可能である．

3. パーザーの概要

本パーザーは Java のクラスライブラリとして提供されている．パーザーの利用者は，Parser クラスに入力ストリームを与えてインスタンス化し，そのインスタンスの parse メソッドを呼ぶことによってSGML文書がパーズされ，DOMインスタンスを得ることができる．

SGML文書は文書型宣言の部分とインスタンスの部分から成る．パーザーは文書型宣言部を読み始めると要素・属性定義を貯えていく．次のインスタンス部では文書型定義部で読んだ要素・属性定義をもとに構文解析をしてDOMインスタンスを生成する．

本パーザーのパッケージでは Parser というクラスのみが公開クラスとなっている．本パーザーの利用者はDOMインターフェースとこのクラスのみを知っていればよい．

4. DOMインターフェースを備えたアプリケーションの作成

4.1 ファクトリの置き換え

単純に構文解析をさせただけのパーザーの出力の木の個々のノードは，単にDOMインターフェースを実装しただけである．この木のノードを生成したファクトリが単に Documentインターフェースを実装しているに過ぎないデフォルトのインスタンスだからである．

しかし，本パーザーでは利用者が自由にファクトリを置き換えるられるメソッドを提供して

A SGML Parser based on DOM interface
Go Kondo, Shin-ichi Hirose, Fumihiko Kitayama, Kazushi Kuse
Tokyo Research Laboratory, IBM Research

いる．アプリケーションの開発者は，本パーザーのファクトリを置き換えることによって単なるDOMの木でなく，DOMインターフェースを備えたアプリケーションをSGMLから作ることができる．

4.2 DOM実装クラス作成支援ツール

DOMの各インターフェースの実装はパッケージ内に隠蔽してあるので，本パーザーの利用者は唯一の公開クラスのParser のみを使えばよい．しかし，新たにDocument インターフェースの実装クラス，及びそれが生成するDOMインターフェースをもったアプリケーションのサブクラスを作る場合，DOMインターフェースのもつ全てのメソッドを一から書かなければならない．そこで，それらのクラスの作成するツールを提供する．

DOM Level 1 コア部はオブジェクト・モデルのデータ構造を抽象化したものであるので，それを実装するコードはある程度パターン化されている．そのためそのパターンをテンプレートとして用意しておき，ツールはそれをもとに DOM インターフェースを実装するクラスを作る．

まず，Documentインターフェースであるが，実際にインスタンスを生成しているファクトリ・メソッドは，以下の３つである．

```
createElement (String tagName)
createTextNode (String data)
createCDATASection (String data)
```

ツールにはこれらのメソッドのコードを生成するためのヒントとして以下のような入力を与える．

1. Document を実装させたいクラス
2. Element を実装させたいクラス
3. 2.のクラスがインスタンス化される条件．つまり上の createElementメソッドからどのような引数が渡されたときに生成されるか．
4. CDATASection を実装させたいクラス
5. Text を実装させたいクラス
6. 4.，5. それぞれのクラスで文字列を示すprotected なインスタンス変数，あるいは文字に対してget/setを行うメソッド．

これらの入力を受け取り，本ツールはDOMインターフェースを実装するクラスのファイルを出力する．

ここで注意しなければならないのが，アプリケーションのクラスとDOMインターフェースの直交性である．DOMのインターフェースは，木というデータ構造の基本的な性質のインターフェースとなっている．このような性質をすでに持っているアプリケーションに対してDOMインターフェースを実装させたい場合には両方のインターフェースに対して正しく動作する実装を与えなければならない．

木構造をもつアプリケーションにも，いろいろあるが，本ツールでは最も一般的と思われる，各ノードは１つの親とVectorで保持された子供を持つというデータ構造をサポートすることにした．そして，新たに以下の条件を入力として与えれる．

7. 2.，4.，5. のスーパークラス
8. 7. のクラスにおいてノードの子供を示すprotectedなインスタンス変数，あるいは子供に対してget/setを行うメソッド
9. 7. のクラスにおいてノードの親を示すprotectedなインスタンス変数，あるいは親に対してget/setを行うメソッド

5. 本パーザーの応用例

本パーザーの応用例として考えられるのはHTMLパーザーである．本パーザーにHTMLのDTDを読み込ませ，Level 1 HTML部のDOMインターフェースを実装するようなファクトリに置き換えればHTMLパーザとして働かせることができる．

多種端末向けWebアプリケーション構築技術のDharma[2]では，本パーザとDOM実装クラス作成支援ツールを用いてDOMインターフェースを備えたDharmaアプリケーションオブジェクトをHTMLから生成する技術を開発中である．

6 今後の課題

本パーザーに付属のDOM実装クラス作成支援ツールは，まだサポートするデータ構造が１種類の単純なものである．今後は多様なデータ構造をサポートする必要がある．

参考文献

[1] http://www.w3.org/TR/WD-DOM-Level-1/

[2] 北山，広瀬，久世．多種端末向けビジネスオブジェクト Web アプリケーション構築システムのプロトタイプ実装．オブジェクト指向シンポジウム．情報処理学会，1998.