

1. はじめに

XML[2]は、そのさまざまな用途に対応する柔軟性と、XSLT[2]などと連携することによる高い可読性から急速に広まりつつある。XML をアプリケーションで操作する手法の一つに、DOM (Document Object Model)[2]が存在する。これは、XML ドキュメントをツリー構造で表現し、またそれらを生成、削除、編集するためのインターフェースの総称である。共通のデータフォーマットを使用するアプリケーションを複数開発する場合は、DOM を利用し共通のオブジェクトモデルを開発することで、開発期間の短縮やバグの抑制が出来ると思われる。

そこで本研究は、初等アセンブラプログラミング評価システム[1]の為のデータをXML化した際に、それに対応したアプリケーションのオブジェクトモデルを提案することを目的とする。また、提案モデルに基づいたツールとして問題エディタを試作した。

2. 提案するモデル

本研究では、上記のオブジェクトモデルを2種類のオブジェクトを用いて実現した。図1は、これらのオブジェクトとDOM の関係を示したものである。

(1)シリアライザオブジェクト

アプリケーションとDOM の間に介在する。DOM と似た階層構造を有し、アプリケーションで用いるデータに応じた派生オブジェクトが存在する。DOM の構造を元にオブジェクトを生成(Load)し、逆にデータからDOM ノード[2]を構築(Save)する。値は直接取得(Get)と設定(Set)が可能である。

(2)クリエイターオブジェクト

要求に応じて、内部にシリアライザオブジェクトを生成し、生成した実体への参照を返す。アプリケーション終了時にはガベージコレクションを行う必要がある。

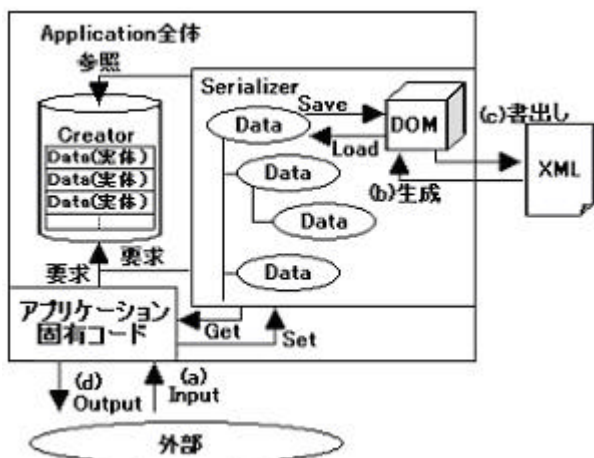


図1.オブジェクトとDOM の関係

3.問題エディタのシステム概要

問題エディタはC++[3]によるCGIプログラムで構成した。XMLパーサには、apacheプロジェクトのXerces[2]を使用した。図2に処理の流れを示す。図2の(a)~(d)は図1のそれに対応している。

- (a) ブラウザから取得したデータをもとにオブジェクトの生成とエディタの設定を行う。
- (b) XMLファイルが指定されていた場合は、XMLを読み込み、それを元にオブジェクトを生成する。
- (c) XMLをファイルに保存する。
- (d) 確認メッセージ(Html)やエラーをブラウザに出力する。



図2.処理の流れ

4.考察

(1)提案したオブジェクトモデル

シリアライザオブジェクトのデータのアクセス方法が煩雑になってしまい、操作性について改善の余地があると感じた。また、手動で生成した派生オブジェクトをXML解析による自動生成手法を考えていきたい。

(2)DOM

DOMには、扱いが比較的簡単な反面、XMLを解析してメモリに格納するため時間とリソースの消費という弱点もある。そのため、Web上のアプリケーションで用いる場合は実際どれほどの負荷が掛かるか調査が必要だと思われる。これに関しては軽量で動作の速いSAX(Simple API for XML)[2]というイベント駆動型のXMLパーサの使用も考えていきたい。

5.終わりに

本研究では、DOMを用いたアプリケーション間の共通データモデルの提案とそれに基づいたエディタの開発を行った。今後は、他のアプリケーションを試作し、親和性や操作性の向上を図っていきたい。

参考文献

- [1] 渡辺博芳、荒井正之、武井恵雄：「事例に基づく初等アセンブラプログラミング評価システム」(情報処理学会論文誌 Vol.42, No.1, pp.99~109, 2001)
- [2] David Hun, Kurt Cagle, Dave Gibbons, Nikola Ozu, Jon Pinnock, Paul Spencer 著 風工舎 役：「エキスパートから学ぶXML実践プログラミング」インプレス, 2001
- [3] 増田智明著：「C/C++実践プログラミングリファレンス」エーアイ出版, 2002