

## 携帯電話向け DOM パーサ「cDOM」の開発

山口 真吾<sup>†</sup>

東京農工大学工学部  
情報コミュニケーション工学科<sup>‡</sup>

並木 美太郎<sup>‡</sup>

東京農工大学工学教育部  
情報コミュニケーション工学専攻<sup>‡</sup>

### 1 はじめに

現在、多くの携帯電話は Java 実行環境を搭載しており、Web 上のコンテンツと連携したモバイルアプリケーションを動作させることが可能である。一方、Web 上のコンテンツは利便性・汎用性の点で優れた XML をベースとしたシステムへと移行が進められている。従って、携帯電話で XML を扱うようにすることで、携帯電話の使用用途を広げ、より便利になると考えられる。そこで、Java CLDC に準拠した XML パーサを作成し、携帯電話で XML を扱うアプリケーションの作成を支援する。

### 2 目標

携帯電話の Java 実行環境で動作する DOM パーサ「cDOM」を開発する。XML パーサの標準 API には SAX と DOM があるが、XML 文書の編集が可能な DOM を採用した。

既存の DOM パーサ「JAXP[3]」や「JDOM[4]」では 10KB の XML 文書で DOM 木を構築した場合、メモリ使用量は 200KB ほどになる。しかし携帯電話は CPU・メモリ共に性能が低いので、メモリ使用量はもっと小さい方が望ましい。そこで、10KB の XML ファイルを DOM 木にしたときのメモリ使用量を 100KB 以内、かつ、木の構築や検索・編集速度も実用に耐えうるレベルを目指す。

### 3 compactDOM

携帯電話で扱うにはフルセットの XML や DOM は大きいので、サブセット化したものを用いることにした。

本開発では、過去に携帯電話上で稼働する SAX を作成するために、軽量化を目的とした XML のサブセット compactXML を策定した [6]。compactXML はフルセットの XML から DTD と実体参照 (内部・外部実体) を削除したもので、この compactXML を採用する。

この compactXML に対応させた DOM のサブセット compactDOM を策定した (表 1)。DOM のバージョンは省メモリの観点から DOM1、それも Core 部分のみとした [2]。

正規の DOM1 Core から、compactXML で非サポートの機能を提供するインタフェース DocumentType, Notation, Entity, EntityReference を削除している。また、NodeList は Vector で代用できるので削除した。属性は要素のプロパティ値が格納されることが多いので、属性を扱う Attr や NamedNodeMap は Element にその機能を組み込むことで削除した。

表 1: compactDOM

基本インタフェース	
DOMException	
DOMImplementation	
DocumentFragment	
Document	
Node	
CharacterData	
Element	
Text	
Comment	
拡張インタフェース	
CDATASection	
ProcessingInstruction	

### 4 cDOM

cDOM とは前節の compactDOM を採用した DOM パーサである。cDOM コンポーネントは大きく 2 つに分けられる。全体の概要図を図 1 に示す。

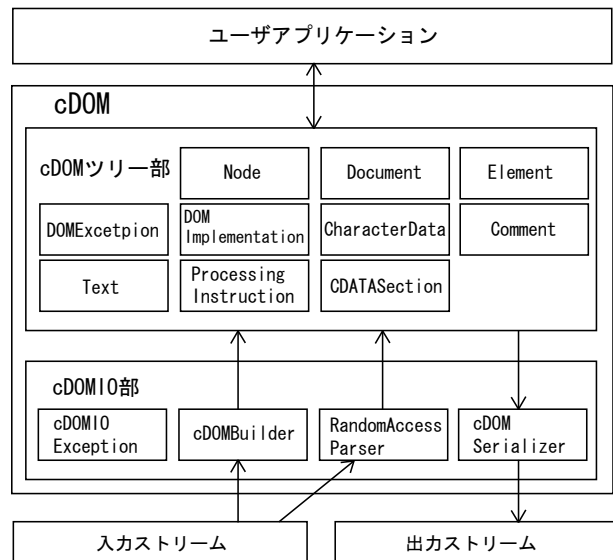


図 1: cDOM コンポーネント全体の概要図

Development of a DOM Parser“ cDOM ”for Cellular Phones

<sup>†</sup> Shingo Yamaguchi

Department of Computer, Information and Communication Sciences, Tokyo University of Agriculture and Technology

<sup>‡</sup> Mitaro Namiki

Department of Computer, Information and Communication Sciences, Tokyo University of Agriculture and Technology

#### 4.1 cDOMIO 部

入出力ストリームと cDOM ツリー部との架け橋となる部分である。XML 文書から cDOM ツリーを構築する cDOMBuilder クラスや、必要なデータのみ取得、解析し、その結果を cDOM ツリーに渡す RandomAccessParser クラス、cDOM ツリー部の内容を XML 文

書としてファイルに書き出す cDOMSerializer クラスから構成される。

## 4.2 cDOM ツリー部

DOM 木を構成する部分で、cDOM の中心部分となっている。compactDOM の仕様に沿ったインタフェース・クラスで構成される。

## 5 cDOM のアルゴリズムと特徴

### 5.1 構造木と部分解析法

処理方式として、初めに XML 文書全体をパースして DOM 木を作る際に、テキストやコメントなどの内容に関する部分はメモリ上に展開しない。つまり、XML 文書の構造だけを表す DOM 木を作る。そしてテキストノードなどのデータ部分にアクセスが生じた際に、RandomAccessParser を用いて必要部分だけ解析し、データを取得することでメモリを節約する。

### 5.2 要素名マップ・属性名マップ

要素名を各要素ノードに文字列として保持するのは大変無駄が生じる。そこで、ハッシュテーブルを用いて要素名を管理する。これを要素名マップと呼ぶ。

要素名マップのキーとして、要素名を保存した String クラスのハッシュコードを用いる。値には要素名の String と、要素名の参照元ノードの配列を入れる。なぜ参照元ノードの配列が必要かという点、要素ノードの検索効率を上げるためである。

要素名マップによって、要素ノードはハッシュコードを保持する int 型変数をメンバに持つだけで要素名を取得できる。また、属性名も重複が多いので、同様に属性名マップを作り管理することにした。

### 5.3 部分保持法

前述の通り、テキストデータは必要な部分だけメモリ上に展開する。しかし、テキストデータへのアクセスが生じたとき、テキストデータを展開するには十分なメモリが残っていない事態が起こり得る。この場合、すでに展開済みのテキストデータを対象に、次の処理を行う。

1. 最近もっともアクセスのない展開済みテキストデータを探す
2. 1 のデータが未変更 (ファイル上のデータと一致する) の場合はデータを破棄して 4 へ
3. データに変更があった場合は携帯電話のストレージ上のテンポラリファイルにデータを一時退避する
4. 十分な空きが確保できていなかったら 1 へ戻る

つまり LRU 方式で必要なデータのみ保持する。これによって、大きな XML 文書を少ないメモリで扱うことを可能にした。

## 6 実装と評価

cDOM の性能を測るために、DOM 木を作るまでの時間とそのときの DOM 木のメモリ使用量、およびノードの検索時間を測定するアプリケーションを作成し、DoCoMo の SHARP 製携帯電話 SH900i と Nokia 製携帯電話 N-Gage 上で動作させ、結果をグラフにした (図 2-5)。比較対象として、kXML2[5] の値を一緒に掲載する\*。図 3 は getElementByTagName() メソッドを用いて、文書全体から特定の要素ノードのリストを取得する時間を計測した結果である。図 4 は返すリスト内のノードの並び順を考慮しない、cDOM 独自の検索メソッドを用いた場合の結果である。SH900i はメモリが足りなくなるまでガベージコレクションが行われず、メモリの使用量は参考にならないので、図 5 から除外した。

\* kXML2 には getElementByTagName() メソッドがないので、検索には同等の動作をするメソッドをアプリケーション側で作成し、それを用いた。

なお、これらの結果はファイルの入力に関する部分は含めていない。

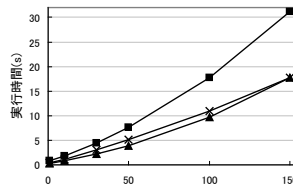


図 2: DOM 木構築の実行時間

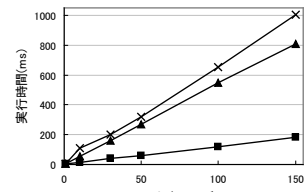


図 3: 要素ノード検索の実行時間

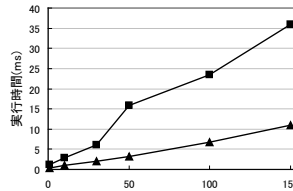


図 4: 並び順を考慮しない要素ノード検索の実行時間

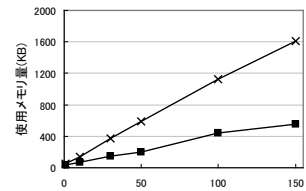


図 5: DOM 木構築後の使用メモリ量

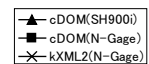


図 2 において、cDOM(SH900i) が速いが、これはガベージコレクションが働いていないためと考えられる。cDOM(N-Gage) は全体的に kXML2 より遅いものの、10KB の XML 文書で木構築に 2 秒ほど、50KB で 6 秒ほどであり、十分実用的な速度である。図 3 を見ると、同じ機種で kXML より cDOM が圧倒的に速い。これは検索に要素名マップを効果的に使ったためと考えられる。SH900i でも 150KB の XML 文書の検索が 0.8 秒と速い。また、図 4 では 150KB ほどの XML 文書でも 0.05 秒以内に検索が終わっていて、これは大変速い。図 5 からは、cDOM のメモリ使用量は 10KB の XML 文書で約 66KB と、kXML2 の約 1/2 である。50KB の XML 文書では cDOM は 200KB ほどだが、kXML2 はその 3 倍の 600KB になっている。

目標であった、10KB の XML 文書でメモリ使用量は 100KB 以内、かつ、実用に十分耐えうる処理速度を十分に達成した。また、cDOM 全体の JAR ファイルサイズは 35KB ほどとなった。kXML2 の JAR ファイルサイズは 27.7KB ほどである。

## 7 おわりに

本論文では、携帯電話向け DOM パーサ「cDOM」について述べた。今後、携帯電話の性能は今後も上昇すると考えられるので、将来的に cDOM を DOM の Level2 または Level3 や XPath に対応させることで、携帯電話で XML を扱うアプリケーションの作成をより支援することができる。

### 参考文献

- [1] Extensible Markup Language (XML) 1.0 (Second Edition), <http://www.w3.org/TR/2000/REC-xml-20001006>
- [2] Document Object Model (DOM) Level 1 Specification, <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/>
- [3] Sun Microsystems, Inc. : Java API for XML Processing (JAXP), <http://java.sun.com/xml/jaxp/index.jsp>
- [4] Jason Hunter, Brett McLaughlin : JDOM, <http://www.jdom.org/>
- [5] Stefan Haustein, Frank Schorder, Nicola Fankhauser, Jorg Pleumann, David Li : kXML2, <http://kxml.kobjects.org/>
- [6] 益子由裕 : 携帯電話向け XML 処理用ミドルウェアの開発, 修士論文 (2004)
- [7] Robert van Engelen : Code Generation Techniques for Developing Light-Weight XML Web Services for Embedded Devices, Proceedings of the 2004 ACM symposium on Applied computing, Vol.19, No.162, pp.854-861(2004)