M-69

携帯用 XML 処理系のライブラリ開発

Development a Library for XML Processors on Cellular Phones

益子　由裕†　　並木　美太郎†
Yoshihiro　Mashiko　　Mitaro　Namiki

1. はじめに

現在、iモード等により、携帯電話は最も手軽な情報端末としての地位を確立しつつある。また、iアプリに代表されるように、新しい携帯電話には Java 実行環境（KVM）が搭載されている。これら携帯用 Java アプリケーション（以下、携帯用 Java）では HTTP 通信がサポートされ、インターネット上のコンテンツと連携したアプリケーションを作成することが可能で、将来性に期待が持たれている。しかし、現時点では、スタンドアロン型のゲーム等が主な利用法であり、インターネット上のコンテンツと連携した実用的なアプリケーションは少ない。これは、携帯用 Java を効果的に利用する方法が見出されていないことも要因として挙げられる。

そこで、携帯用 XML 処理系（以下、本システム）では、XML を用いて携帯用 Java とインターネット上のコンテンツを連携させるシステムを提案する。端的に述べれば、携帯用 Java から XML を扱うクラスライブラリを提供することである。XML を利用することで、他のシステムとのコミュニケーションを円滑化する効果がある。

2．ターゲット

本システムでは、NTT ドコモの i アプリをターゲットとするが、その理由は以下の２点である。

（1）記憶容量

先日発売されたドコモの 504i では、ScratchPad が、503i の 10KB から 10 倍の 100KB となった。この結果、アプリケーション部分を UI、データ処理として、ScratchPad をデータベースとして、効果的に i アプリを利用するという構成が確立された。本システムでもこの構成を用いる。

（2）赤外線通信機能

504i では、赤外線通信機能が搭載されており、これを i アプリから制御し、赤外線搭載機器のコントロールシステムを実現することも可能である。

3．目標

本システムの目標は、（1）ScratchPad への XML のストアと（2）i アプリ上での XML の解析と参照を満たしたライブラリの開発である。（1）、（2）の要求を満たすことは、以下に示す意味を持つ。

（1）ScratchPad への XML のストア

ScratchPad に XML をストアしておかなければ、i モード圏外時には、システムが利用不可能となる。また、i モードはパケット課金制であり、システムを利用する度にインターネットにアクセスし、XML を取得することは非常にコストがかかる。赤外線通信を利用し、i アプリを赤外線搭載機器のコントローラ化する場合にも、ScratchPad への XML のストアは必須の事項となる。

（2）i アプリ上での XML の解析と参照

（1）により、ScratchPad へ XML をストアするため、当然 i アプリ上で XML の解析を行わなければならない。携帯電話の処理能力を考えると、i アプリ上で、XML を処理することは非常に負荷がかかる。しかし、ScratchPad に XML をストアし、i アプリ上で XML の解析を行うことは、システムを常時利用可能にするという重大な利点をもたらすことになる。

4．設計方針

前節の目標を実現するために、以下の設計方針を掲げる。

（1）実行速度の向上

携帯電話の処理能力は PC と比べ非常に低く、その上で XML の解析を行うことはアプリケーションの実行速度の低下を招く。また、アプリケーション実行毎の解析も実行速度の低下に大きく影響する。そこで、本システムでは、INDEX を用いた事前解析方式を採用することで、実行速度の向上に努める。

（2）ライブラリおよび関連するデータサイズの縮小

503i におけるアプリケーションサイズの上限は、JAR ファイルで 10KB である。504i では、3 倍の 30KB となったが、依然として快適にアプリケーションを作成できる環境ではない。更には、ヒープ領域も小さい。そこで、本システムは、無駄な機能を一切省き、必要とされる最小限の機能のみで構成することを方針とする。現在、TinyXML[3]、AElfred[4]といった KVM 上で動作する XML パーサがあるが、これらはターゲットデバイスを palm とし、パーサ容量が大きく、i アプリへの実装は難しい。

5．システム構成と機能

5．1　全体構成

本システムは、以下の２つのライブラリから構成される。

（1）i アプリ用 XML パーサ

（2）ScratchPad におけるファイル管理

次に、本システムのアーキテクチャを図１に示す。

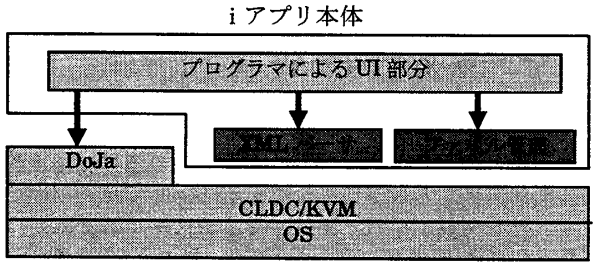

図１　携帯電話上のアーキテクチャ

5．2　XML パーサ

本システムで提供する i アプリ用 XML パーサは、本システムの目標の一つである i アプリ上での XML の解析を実現するライブラリである。この i アプリ用 XML パーサの特徴は、以下の２点である。

†東京農工大学工学研究科

（1）compactSAX

XML を解析する際の標準的なインタフェースとして、DOM、SAX の２つが存在する。DOM は、SAX に比べ多種多様の処理を行える反面、多くのメモリを必要とし、パーサ自体の容量も大きい。SAX は、イベント駆動型で DOM に比べ、少ないメモリで高速に動作し、パーサ自体の容量も小さい。少メモリ、少リソース、低処理能力というｉアプリの実行環境を考慮すると、ｉアプリ用 XML パーサのインタフェースとしては、SAX が適当である。しかし、SAX パーサ自体の容量が少ないといっても、UI 部分とｉアプリ用 XML パーサを併せて、10KB、または 30KB 以内に抑えることは難しい。そこで、本システムでは、携帯電話という環境では不要と考えられる SAX の機能を省いた compactSAX（以下、cSAX）の仕様を策定し、これをインタフェースとして採用している。

cSAX で削減した主な機能には、DTD、実体がある。DTD は XML の妥当性の確認に用いられる。本システムは、既存の XML を利用し、妥当性や整形式であることについては、既に確認してあることを前提としており、cSAX を DTD に対応させる必要性はない。その結果、DTD の構文を利用して宣言する実体についても非対応となる。

（2）事前解析方式

一般的には、SAX は高速で動作するという認識であるが、それはあくまでも DOM と比較した際のことである。ｉアプリ上での XML の解析、そしてアプリケーション実行毎の解析は、アプリケーション実行速度を大きく低下させ、ユーザに不快感を与える。そこで、本システムでは、INDEX を用いた事前解析方式を用いることで、XML の解析処理に要する時間を可能な限り抑える。

この方式では、実際に cSAX による解析を行う前に、事前解析を行い、ScratchPad 上に cSAX による解析を行う際に必要な情報を保持した INDEX を生成する。cSAX による解析では、事前解析で作成された INDEX を利用する。この INDEX を用いた事前解析方式は、事前解析時においては、INDEX の生成を行うこともあり、時間を要する処理となるが、それ以降の解析では、INDEX にある各カラムの情報を利用することで、XML 全体に対する解析を行う場合よりも、解析時間を短縮することが可能となる。この方式は、INDEX を用いて XML に対して簡単に高レベルな処理をしたい場合に有効であり、XML の書き換えといった低レベルな処理をしたい場合には、INDEX を介さず、ScratchPad 上の XML に直接アクセスすればよい。

５．３　ファイル管理

本システムで提供する ScratchPad におけるファイル管理は、本システムの目標の一つである ScratchPad への XML のストアを実現するライブラリである。本システムでは、XML の他に事前解析によって生成される INDEX も ScratchPad 上にストアしており、ScratchPad をデータベースとして活用し、ScratchPad の重要度は高い。しかし、ScratchPad にはファイルシステムが存在しないため、ScratchPad にファイルやデータをストアする際には、常にバイト単位でファイルやデータの管理を行わなければならず、非常に不便である。ファイル管理ライブラリでは、この問題を解決する。また、このファイル管理ライブラリを XML パーサライブラリと完全に独立させることで、このライブラリを単体で他のアプリケーションに応用することも可能にする。

６．API

XML パーサライブラリの API は、SAX をベースとした cSAX を用いているため、大部分は SAX の API から構成される。主なメソッドを下表１に示す。

表１　XML パーサライブラリの主な API

(1)	parse(InputSource source)
XML の解析を実行	
(2)	StartDocument(),EndDocument()
解析中のイベント発生時に呼ばれるイベントハンドラ	
(3)	preParse(InputSource source)
cSAX 独自のメソッドで、事前解析を実行	

ファイル管理ライブラリは、ファイル管理システムを表す FileSystem クラスを中心として構成される。主なメソッドを下表２に示す。

表２　ファイル管理ライブラリの主な API

(1)	getFilename (int FileID),getFilesize(String filename)
ファイルの各種情報を取得する get メソッドの一種	
(2)	deleat(String filename)
ファイルの削除を行う	
(3)	create (String filename)
新しいファイルを作成	

７．試作の評価

現在までに、事前解析処理部分とファイル管理の一部について試験的に実装を行った。この試作での事前解析に要した時間は、機種によって大きな差が生じたが、SO503i が最速で、3KB の XML の事前解析が約 20 秒であった。しかし、現在、実装方法を多少改善することで、処理時間をより短縮することが可能なことを確認している。また、事前解析によって生成された INDEX の容量は、対象となる XML ファイルに対し 1/2 の大きさであった。このオーバヘッドは小さいとは言えないが、実際の解析時間を短縮することを考慮すれば無駄ではない。更に、INDEX についても、改善の余地があり、容量をより縮小化することが可能なことも確認している。

８．おわりに

本論文では、携帯用 XML 処理系の核となるクラスライブラリについて、その概要を述べた。ライブラリの開発を終えた後には、INDEX による事前解析方式の有効性を確認する。また、前述した赤外線搭載機器のコントローラシステムを本システムで実現することで、本システムの有用性、活用法の一端を示したい。

参考文献

[1] 「ｉモード対応 Java コンテンツ開発ガイド ～詳細編～ 第 1.0 版」（株式会社 NTT ドコモ、2000）、「for 504i」（株式会社 NTT ドコモ、2002）

[2] David Megginson 「SAX The Simple API for XML」 (http://www.saxproject.org/)

[3] TinyXML (http://www.kvmworld.com/articles/techtalk/)

[4] AElfred (http://www.opentext.com/microstar/)