

IMPLEMENTASI RSS READER DENGAN kXML PADA TELEPON GEGGAM

M. Deri Taufan, Fazmah Arif Yulianto, Maman Abdurohman
Jurusan Teknik Informatika, Sekolah Tinggi Teknologi Telkom
Jl. Telekomunikasi Dayeuh Kolot Bandung 40257

Abstrak

RSS merupakan singkatan dari Rich Site Summary, RDF Site Summary atau Really Simple Syndication. RSS yang formatnya mengikuti standar XML 1.0, mempunyai struktur yang sederhana dan ramping, dan didesain untuk dapat membungkus informasi dengan lebih efisien. Pada penelitian ini dibuat aplikasi RSS Reader pada telepon genggam. Aplikasi RSS Reader memanfaatkan API parser kXML. Parser kXML mendukung parsing secara text-based dan parsing secara binary encoding (WBXML --Wireless Binary XML--). Kedua metode parsing tersebut dibandingkan dari sisi waktu (delay parsing) dan kebutuhan sumber daya (memory).

Dari implementasi dan analisis perbandingan kedua aplikasi tersebut didapat bahwa metode parsing dengan binary encoding membutuhkan waktu proses yang lebih singkat dibandingkan dengan parsing secara text-based. Dokumen WBXML lebih kecil ukurannya daripada dokumen text-based karena dokumen WBXML direpresentasi dalam tokens dan literal string. Karena itu memory yang dibutuhkan untuk melakukan parsing dokumen WBXML juga lebih sedikit.

Kata kunci: RSS, telepon genggam, kXML, WBXML

1. Pendahuluan

Pola kehidupan masyarakat yang terus berkembang seiring perkembangan teknologi telekomunikasi, menjadikan kenyamanan, fleksibilitas dan mobilitas menjadi acuan dalam pengembangan teknologi. Perkembangan dunia *mobile device* yang dewasa ini meningkat dengan pesat sedikit banyak menjawab tuntutan masyarakat akan kebutuhan fleksibilitas dan mobilitas dalam ruang lingkup komunikasi data.

RSS yang merupakan kependekan dari *Rich Site Summary*, *RDF Site Summary*, dan *Really Simple Syndication* adalah bentuk sederhana dari XML, yang sejak awal telah digunakan untuk saling berbagi (*sharing*) content dari suatu web dan juga untuk sarana weblogging. Karena kesederhanaannya itu, banyak situs yang menyediakan layanan akses *content/isi* dari web mereka dalam format RSS.

Dalam penelitian ini diimplementasikan aplikasi pembaca format dokumen RSS (RSS Reader) pada perangkat mobile dengan menggunakan API kXML yang mendukung parsing secara *text-based* dan *binary encoding*. Aplikasi ini dibuat sehingga pada akhirnya dapat diketahui metode parsing mana yang paling optimal untuk lingkungan perangkat bergerak (dari hasil analisa perbandingan parsing secara *text-based* dan *binary encoding*).

Beberapa batasan dalam penelitian ini:

- Aplikasi akan dibuat dengan J2ME, diinstall dan bekerja pada telepon genggam.
- Versi RSS yang digunakan adalah RSS versi 1.0
- Encoding dokumen dari format RSS ke format WBXML dilakukan secara offline menggunakan

tool bantuan yang sebelumnya telah dimodifikasi untuk memenuhi keperluan penelitian ini.

- Komunikasi data menggunakan protokol HTTP melalui teknologi GPRS

2. RSS

2.1 Overview RSS

RSS adalah sebuah metode terbuka untuk pengumpulan dan penyatuan sebuah isi halaman web. Sebuah file RSS memberikan data informasi ringkas (summary) tentang *headlines*, *links*, dan *article* dari web site [2].

File RSS termasuk dalam tipe metadata. Metadata adalah unit-unit informasi tentang informasi yang biasa digunakan untuk menyediakan informasi deskriptif tentang content, context dan karakteristik dari data. Metatag *keywords* dan *description* di HTML merupakan contoh metadata, yang digunakan untuk menyediakan informasi tentang web page.

RSS menggunakan XML sebagai formatnya. Sebuah file RSS menerangkan isi dari sebuah saluran informasi (channel) yang berisi *logo/image*, *site link*, *input box* dan *items*. Items pada RSS sering disebut "*news items*". Web site tertentu dapat menyalin dan menggunakan RSS yang dimiliki web site lain untuk menginformasikan sebuah berita atau artikel, hal ini sering disebut dengan *RSS feeds*.

Dengan menggunakan file RSS dari web site lain, dapat ditampilkan intisari berita dari web site tersebut. Cara ini baik sekali apabila diinginkan agar pengunjung yang mengunjungi web site tertentu menjadi betah dengan membaca intisari informasi-informasi yang berupa news dan artikel yang up to date.

Berikut merupakan contoh dokumen RSS versi 1.0.

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
  syntax-ns#"
  xmlns="http://purl.org/rss/1.0/">

  <channel
    rdf:about="http://www.xml.com/xml/news.rss">
    <title>XML.com</title>
    <link>http://xml.com/pub</link>
    <description>
      XML.com features a rich mix of
      information and services
      for the XML community.
    </description>

    <image
      rdf:resource="http://xml.com/universal/image
      s/xml_tiny.gif" />

      <items>
        <rdf:Seq>
          <rdf:li
            resource="http://xml.com/pub/2000/08/09/xslt
            /xslt.html" />
          <rdf:li
            resource="http://xml.com/pub/2000/08/09/rdfdb
            /index.html" />
        </rdf:Seq>
      </items>
    </channel>

    <image
      rdf:about="http://xml.com/universal/images/x
      ml_tiny.gif">
      <title>XML.com</title>
      <link>http://www.xml.com</link>

    <url>http://xml.com/universal/images/xml_tin
    y.gif</url>
  </image>

  <item
    rdf:about="http://xml.com/pub/2000/08/09/xsl
    t/xslt.html">
    <title>Processing Inclusions with
    XSLT</title>

    <link>http://xml.com/pub/2000/08/09/xslt/xsl
    t.html</link>
    <description>
      Processing document inclusions with
      general XML tools can be
      problematic. This article proposes a
      way of preserving inclusion
      information through SAX-based
      processing.
    </description>
  </item>

  <item
    rdf:about="http://xml.com/pub/2000/08/09/rdf
    db/index.html">
    <title>Putting RDF to Work</title>

    <link>http://xml.com/pub/2000/08/09/rdfdb/in
    dex.html</link>
    <description>
      Tool and API support for the Resource
      Description Framework
      is slowly coming of age. Edd Dumbill
      takes a look at RDFDB,
```

```
one of the most exciting new RDF
toolkits.
  </description>
</item>

</rdf:RDF>
```

2.2 Elemen Utama File RSS

- Elemen Channel, terdiri atas metadata yang menjelaskan channel itu sendiri. Sebuah elemen channel setidaknya terdiri dari: nama dari channel `<title>`, deskripsi channel `<description>` dan URL channel `<link>`.
- Elemen Image, adalah elemen optional yang memasukkan sebuah logo pada RSS.
- Elemen Item, merupakan elemen yang penting dan bagian dinamis dari sebuah file RSS. Sementara elemen lainnya bersifat konstan, elemen item justru terus dapat berubah setiap saat bergantung dari isi berita yang ingin ditampilkan. Elemen item terdiri atas tiga bagian yang sangat penting: nama dari item (*headline*) `<title>`, deskripsi dari item (ringkasan) `<description>` dan URL dari item yang bersangkutan `<link>`. Dalam file RSS item-item yang diperbolehkan hanya sampai 15 item. Sedangkan untuk title dibatasi kurang dari 100 karakter sementara untuk pendeskripsian dibatasi kurang dari 500 karakter [2].
- Elemen Textinput, merupakan elemen optional dalam file RSS. Elemen textinput berfungsi sebagai system feedback dari sebuah berita. Biasanya digunakan untuk mensubscribe ke newsletter atau digunakan untuk fasilitas pencarian di web site yang bersangkutan.

2.3 Versi-versi RSS

Ada berbagai macam versi dari format RSS yang berkembang secara parallel. Format RSS, versi 0.90, didesain oleh Netscape untuk membangun portal berita. Dengan mempertimbangkan kekompleksan dengan tujuannya, maka kemudian dikeluarkan versi sederhana 0.91, tapi kemudian ditinggalkan begitu saja ketika Netscape tidak tertarik lagi dalam hal bisnis pembuatan portal. Tapi kemudian versi 0.91 diambil alih oleh vendor lain, UserLand software, yang kemudian berniat untuk menggunakannya dalam basis pembuatan produk webbloging dan program menulis berbasis web lainnya.

Dan selang beberapa waktu, pihak ketika, kelompok non-profit mendesain format RSS terbaru berdasarkan format RSS asli versi 0.90. Format ini berdasarkan RDF (*Resource Description Framework*) dan dikenal dengan RSS versi 1.0. Dan pihak UserLand tidak terlibat dalam proses desain versi 1.0 ini, sehingga begitu RSS versi 1.0 dirilis pihak UserLand merasa tidak senang. Dan kemudian pihak UserLand malah menyempurnakan RSS versi 0.91 menjadi 0.92, 0.93, 0.94 dan terakhir versi 2.0.

3. kXML Parser API

RSS merupakan dokumen yang mengandung data sekaligus format data. Ada beberapa proses parsing yang dikenal saat ini, antara lain:

- a. *model*: merupakan parser yang melakukan parsing dengan teknik membaca keseluruhan isi dokumen dan membuat representasi dokumen tersebut di dalam memory. Parser type ini secara signifikan membutuhkan memory yang lebih di banding type lainnya.
- b. *push*: merupakan parser yang melakukan parsing dengan teknik membaca keseluruhan dokumen dan memberikan message kepada object listener ketika menjumpai, atau menemukan element atau bagian dari dokumen. (Teknik inilah yang kebanyakan dilakukan parser SAX). Parser tipe ini juga membutuhkan memory yang besar.
- c. *pull*: merupakan parser yang melakukan parsing dengan teknik membaca sedikit demi sedikit dokumen pada suatu saat. Kemudian aplikasi akan memandu parser ke bagian dokumen selanjutnya dengan meminta kembali bagian dokumen selanjutnya.

kXML merupakan kumpulan API yang stabil yang didesain untuk tugas membaca dokumen yang berformat XML atau turunannya. kXML merupakan contoh dari parser yang bertipe *pull*. Sehingga sangat memungkinkan untuk mengimplementasikan atau menggunakannya dalam lingkungan perangkat mobile, karena membutuhkan memory yang lebih sedikit dari metode lainnya.

3.1 Text Based Parsing

Saat ini kXML mendukung parsing secara *text-based* dan juga parsing secara *binary-encoding*. Parsing secara *text-based* yang dimaksud disini adalah parsing document XML maupun turunan dari document XML. Karena seperti diketahui bahwa document XML merupakan document text biasa.

Untuk melakukan parsing secara *text-based* atau parsing standard, kXML menyediakan package *org.xml.parser* [9]. Package ini menyediakan class *XmlParser* yang bertugas langsung melakukan parsing terhadap dokumen XML secara *pull parser*.

3.2 Binary Encoding/WBXML Parsing

Seperti yang telah disebutkan di atas, bahwa kXML juga mendukung parsing document XML yang berformat binary (WBXML/Wireless Binary XML). Format biner dirancang untuk memungkinkan transmisi ringkas dengan tidak ada kehilangan fungsionalitas dan juga informasi semantik. Format ini dirancang untuk memelihara struktur unsur XML, membiarkan suatu parser untuk melompati elemen atau atribut yang tak dikenal. Format biner menyandi (encode) format fisik dari suatu dokumen XML, seperti, struktur dan isi dari entitas dokumen yang bersangkutan. Meta-

Information, mencakup definisi dari dokumen dan bagian *conditional*, dihilangkan manakala dokumen dikonversi kedalam format binary.

Content dan struktur dari dokumen WBXML direpresentasikan dengan *binary code*, seperti *tokens* dan *literal strings*. String yang sering digunakan dapat disimpan dalam *string table*, sehingga memungkinkan untuk menghindari informasi yang redundant.

Token merupakan *context-dependent binary code*, dengan kata lain suatu token biner tidak secara langsung menghadirkan informasi yang unik, tetapi pengertiannya tergantung dari context. Ini berarti tidak ada hubungan korespondensi satu-ke-satu antara token dan informasi yang dikandung olehnya. Token dapat dikelompokkan ke dalam dua katagori[6]:

- a. Global Tokens: token yang selalu memiliki pengertian yang sama dimanapun token tersebut muncul dalam dokumen. Global tokens terbagi atas; *Generic Set* dan *Extension Set*.
- b. Application Tokens: token yang memiliki pengertian yang berbeda dalam context yang berbeda. Token ini juga terbagi atas dua jenis:
- c. Tag code space, mempresentasikan nama tag yang spesifik.
- d. Attribute code space, terbagi kedalam dua range angka yang merepresentasikan prefix dan nilai (*value*) dari atribut.

Untuk melakukan parsing format WBXML, kXML menyediakan package *org.xml.wap* [9]. Dalam package ini terdapat class *WbxmlParser* yang merupakan class turunan dari *AbstractXmlParser* yang merupakan pull parser untuk melakukan parsing terhadap dokumen WBXML.

4. Deskripsi Sistem

Sistem dibangun pada sisi client, sedangkan pada sisi server akan memanfaatkan web server yang telah ada. Dalam penelitian ini tidak akan dibahas sisi server, karena fungsi dari server hanya sebagai penyedia atau penyimpan file-file yang berformat RSS dan WBXML. Aplikasi dibangun dengan menggunakan Forte for Java Micro Edition dan dengan J2ME Wireless Toolkit 1.0.4. Aplikasi ini akan diimplementasikan pada mobile phone dengan fitur Java Enable.

Cara kerja aplikasi adalah sebagai berikut:

1. Pemakai harus menginputkan URL dari RSS resource yang ingin diparsing atau menggunakan URL default dari aplikasi.
2. Aplikasi akan melakukan parsing terhadap RSS resource tersebut dan menampilkan list dari headline yang telah diparsing.
3. Pemakai selanjutnya dapat menampilkan:
 - Melihat waktu proses
 - Melihat jumlah memory yang digunakan dan yang tersisa

- Melihat jumlah item dalam dokumen RSS
- Melihat *description* dari setiap *headline*

5. Evaluasi Sistem

5.1 Skenario Pengujian

Pengujian hanya akan membahas *delay* dan *resource* (penggunaan memory) terhadap proses parsing. Pada penelitian ini tidak dibahas proses selama data berada di jaringan operator telekomunikasi. Dua aplikasi yang akan diuji memiliki struktur yang sama, dimana masing-masing mewakili parsing *text-based* dan *binary encoding*. Aplikasi akan diinstall pada telepon seluler Sony Ericsson Z600.

Untuk aplikasi 'text-based' parsing akan melakukan parsing terhadap dokumen RSS standard (dokumen text). Dan untuk aplikasi 'binary encoding' parsing akan melakukan parsing terhadap dokumen text-based RSS yang sama dengan yang digunakan oleh aplikasi pertama yang sebelumnya telah diencode menjadi dokumen WBXML. Proses encoding akan dilakukan secara *offline* menggunakan aplikasi tambahan yang sebelumnya telah dimodifikasi untuk memenuhi pengujian aplikasi.

Untuk memenuhi tujuan pengujian, ada beberapa parameter yang digunakan saat melakukan pengujian. Parameter yang digunakan: jumlah item berita dan jumlah *description* karakter. Untuk menguji apakah item berita akan berpengaruh terhadap *delay* dan *memory resource* yang digunakan, maka jumlah item berita yang akan diuji yaitu: 5, 10 dan 15 item dengan *description* karakter berjumlah 500 karakter. Sedangkan untuk menguji apakah jumlah *description* karakter akan berpengaruh terhadap sistem, maka jumlah *description* karakter yang diuji adalah: 100, 200, 300, 400 dan 500 karakter dengan jumlah item beritanya 15 item. Setiap parameter yang akan diuji di atas mewakili satu file yang akan diuji coba.

Tabel 1. Nama file, jumlah item, jumlah *description* character, ukuran teks RSS, dan ukuran WBXML

Nama File	Jumlah Item berita	Jumlah <i>description</i> karakter	Dokumen RSS size (byte)	Dokumen WBXML size (byte)
file1	5	500	5851	5208
file2	10	500	10527	9400
file3	15	500	15203	13575
file4	15	100	9290	7662
file5	15	200	10796	9168
file6	15	300	12217	10589
file7	15	400	13693	12065

Data atau dokumen RSS asli diambil dari site <http://www.ecommercetimes.com>, pada tanggal 16 Januari 2005 dengan memanfaatkan media syndication yang tersedia di situs tersebut. Dokumen

yang diperoleh memiliki 13 item berita dengan rata-rata jumlah *description* karakter 324 karakter. Dokumen tersebut disesuaikan dengan keperluan uji coba sesuai dengan yang diuraikan diatas.

File-file tersebut diupload ke internet untuk dapat diakses oleh aplikasi menggunakan jaringan GPRS dari telepon genggam. Upload dilakukan pada <http://www.balnommoc.net>. Dokumen text-based RSS, diletakkan di <http://www.balnommoc.net/rss/>, sedangkan untuk dokumen WBXML, file diletakkan di <http://www.balnommoc.net/wbxml/>.

5.2 Pelaksanaan Uji Coba

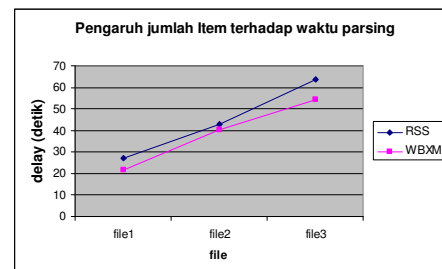
Pelaksanaan uji coba sistem dilakukan untuk setiap parameter yang ditetapkan. Mengingat faktor kestabilan jaringan GPRS, maka uji coba dilakukan selama 5 hari dimulai pada tanggal 17 Januari 2005. Untuk setiap harinya uji coba dilakukan 4 kali pada jam-jam sebagai berikut: 08.30, 11.30, 17.30 dan 00.30.

5.2.1 Pengaruh Jumlah Item Berita

Untuk mengetahui pengaruh jumlah item terhadap unjuk kerja parsing RSS, digunakan parameter sebagai berikut:

- Jumlah Item: 5, 10, dan 15 buah
- Jumlah karakter *description*: 500 karakter
- File: file1, file2, dan file3

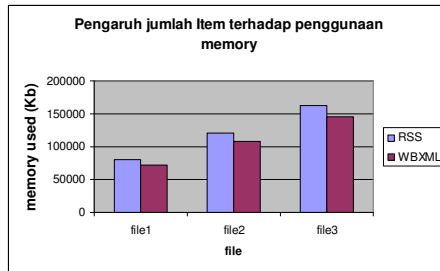
Dari percobaan didapatkan data sebagai berikut:



Gambar 1. Grafik pengaruh jumlah item terhadap waktu parsing

Dari grafik tersebut di atas terlihat bahwa makin besar ukuran file dari dokumen, baik itu dokumen text-based RSS maupun dokumen WBXML, waktu parsing yang dibutuhkan juga makin besar. Karena ukuran file dokumen teks RSS lebih besar dari ukuran dokumen WBXML, maka sangat logis jika dokumen text-based RSS membutuhkan waktu parsing yang lebih besar dari pada dokumen WBXML. Jumlah item berbanding lurus dengan ukuran file. Dengan demikian makin banyak jumlah item, maka ukuran file akan bertambah, dan juga waktu parsing akan bertambah.

Di bawah ini grafik penggunaan memory:



Gambar 2. Grafik pengaruh jumlah item terhadap penggunaan memori

Dari grafik penggunaan memory di atas, terlihat bahwa dokumen text-based RSS membutuhkan *resource memory* yang lebih besar dibanding dengan dokumen WBXML. Hal ini bisa dipahami mengingat dokumen text-based RSS lebih besar dari dokumen WBXML. Namun faktor ini bukan hanya karena ukuran file dari dokumen yang bersangkutan, tapi juga diakibatkan oleh representasi dari dokumen WBXML itu sendiri.

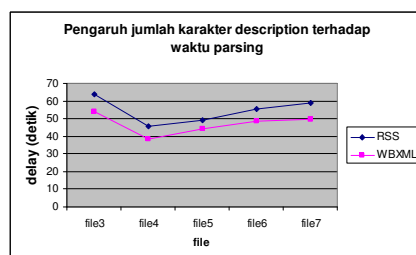
Seperti yang dijelaskan sebelumnya, dokumen WBXML direpresentasikan dengan token atau binary code. Untuk suatu penamaan tag seperti tag `<entry>` misalnya, jika terdapat 15 tag `<entry>`, maka dalam dokumen *text-based*, semua tag yang sama juga harus ditulis. Ini tentunya merupakan masalah redundand dalam penulisan tag. Sebaliknya pada dokumen WBXML tidak terjadi masalah redundansi. Hal ini disebabkan karena untuk suatu *string* yang sering dipakai seperti nama tag dan atribut bahkan value dari atribut dapat diletakkan ke dalam suatu *string table* pada dokumen WBXML yang bersangkutan.

5.2.2 Pengaruh Jumlah Description Character

Untuk mengetahui pengaruh jumlah description character terhadap unjuk kerja parsing RSS, digunakan parameter:

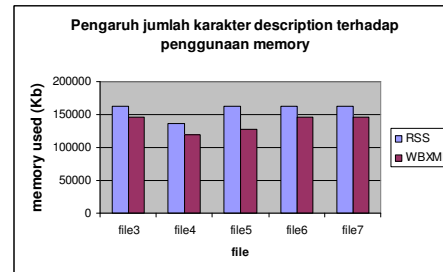
- Jumlah Item: 15 buah
- Jumlah karakter description: 100, 200, 300, 400, dan 500 karakter
- File: file3, file4, file5, file6, dan file7

Dari percobaan didapatkan data sebagai berikut:



Gambar 3. Grafik pengaruh jumlah description character terhadap waktu parsing

Di bawah ini adalah grafik penggunaan memory:



Gambar 4. Grafik pengaruh jumlah description character terhadap penggunaan memori

Grafik-grafik di atas menjelaskan bahwa jumlah karakter *description* juga sangat berpengaruh terhadap waktu yang dibutuhkan untuk melakukan parsing. Hal ini logis mengingat bahwa makin besar jumlah karakter *description* berdampak pada makin besarnya ukuran file dokumen yang bersangkutan.

Jika dibandingkan jumlah karakter description pada dokumen text-based RSS dengan dokumen WBXML, maka akan terlihat bahwa ukuran file dokumen text-based RSS lebih besar dari pada ukuran file dokumen WBXML. Ukuran file tentunya akan berbanding lurus terhadap waktu dan jumlah memory yang dibutuhkan untuk melakukan parsing. Sehingga tidak salah kiranya jika dokumen WBXML memiliki performansi yang lebih bagus dari dokumen text-based RSS.

6. Kesimpulan

Berdasarkan percobaan dan analisis yang telah dilakukan, disimpulkan hal-hal sebagai berikut:

1. Makin banyak jumlah item pada suatu dokumen RSS dan dokumen WBXML maka waktu parsingnya juga lebih besar dan juga resource memory yang digunakan juga lebih besar.
2. Makin banyak jumlah karakter description maka waktu parsingnya juga lebih besar dan memory yang digunakan juga lebih besar.
3. Ukuran dokumen WBXML lebih kecil 11.89% dibandingkan dengan ukuran dokumen text-based RSS yang mengandung informasi yang sama.
4. Metode parsing yang optimal untuk lingkungan mobile phone adalah metode parsing dokumen *binary encoding* / WBXML. Karena waktu parsing dokumen WBXML lebih kecil 13.38% dibandingkan dengan waktu parsing dokumen text-based RSS yang terdiri dari informasi yang sama. Dan juga Dokumen WBXML membutuhkan memory yang lebih sedikit 12.34% dibanding dengan dokumen text-based RSS.
5. Selama percobaan 5 hari dan 4 kali uji coba perhari, ternyata faktor padatnya atau sibuknya trafik GPRS tidak mempengaruhi keunggulan

(dari segi waktu dan memory yang digunakan)
parsing dokumen WBXML terhadap dokumen
text-based RSS.

Daftar Pustaka

- [1] Keogh, James, 2003, *The Complete Reference J2ME*, McGraw Hill.
- [2] Bachtiar, Edwin Maolana, 2003, *Pengantar RSS*, www.ilmukomputer.com.
- [3] Wu, C. Thomas, 2001, *An Introduction to Object-Oriented Programming with Java*; McGraw Hill.
- [4] Knudsen, Jonathan, 2002, *Parsing XML in J2ME*,
<http://developers.sun.com/techttopics/mobility/midp/reference/techart/index.html>
- [5] Giguere, Eric, 2002. *Compressing XML for Faster Wireless Networking*,
<http://developers.sun.com/techttopics/mobility/midp/reference/techart/index.html>.
- [6] W3C, 1999, *WAP Binary XML Content Format*, Wireless Application Forum Ltd.
- [7] Pilgrim, Mark, 2002, *What is RSS?*, www.xml.com.
- [8] W3Schools, 2004, *RDF Dublin Core*, http://www.w3schools.com/rdf/rdf_dublin.asp.
- [9] Hausteijn, Stefan, 2001, *kXML API Documents*, www.kxml.org
- [10] Larman, Craig, 1998, *Applying UML and Pattern*, Prentice Hall
- [11] Bray, Tim, 2001, *What is RDF*, www.xml.com