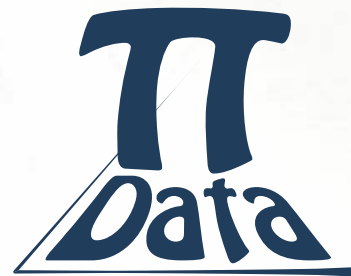


Kompression von XML-Daten



PI-Data, Peter Rudolph, Arthur Miskolczi

<http://www.pi-data.de>

Einführung

Warum Kompression?

Kompressionsverfahren

Lösungen

Fazit

Links

XML

Wie sieht XML aus?

- XML ist eine Baumstruktur im Text-Format
- Knoten werden durch Tags identifiziert (z.B. <Person> </Person>)
- Knoten kann Attribute haben (z.B. <Person Name="Egon"> ...)
- Struktur wird mittels DTD oder XML Schema definiert. XML Schema ist zusätzlich streng typisiert.

Warum XML auf dem Handy?

- Zugriff auf Backoffice
- Einzig mögliche Schnittstelle in MIDP ist HTTP
- SOAP definiert Kommunikation mittels XML über HTTP
- WebServices verwenden SOAP zur Kommunikation
- Geschäftsprozess-Definition mit BPEL4WS baut auf WebServices auf

XML im PI-Data Framework

XML

- möglichst vieles wird in XML beschrieben
- Server interpretiert XML
- für Mobilgerät wird Code aus XML generiert
- eigener Parser, der extrem sparsam mit Speicher umgeht
- Kommunikation über SOAP

Eigene Modellklassen

- Ersatz für fehlende Collections
- Implementierung auf Basis von MIDP
- Definition in XML Schema



Einführung

Warum Kompression?

Kompressionsverfahren

Lösungen

Fazit

Links

Warum Kompression bzw. binäres Format?

Gewünschte Verbesserungen

- Verhältnis Nutzdaten zu Gesamtdaten verbessern
- Bandbreite besser nutzen (vor allem bei Mobilfunk)
- reduzieren von CPU-Last und Speicherbedarf beim Parsen
- Verarbeitungsgeschwindigkeit verbessern
- zusätzliche Eigenschaften werden möglich, z. B. Random Access mittels Index

Kompression zu welchem Preis

- Lesbarkeit durch den Menschen geht verloren
- Interoperabilität geht mangels Standard verloren

W3C und XML Kompression

W3C Workshop on Binary Interchange of XML Information Item Sets

- Zeitraum März 2004 März 2005
- sollte untersuchen ob Bedarf für einen binären XML-Standard besteht

Durchgeführte Untersuchungen

- Charakterisierung für einen binären XML-Standard
- Untersuchen von Use-Cases, aufstellen von Anforderungen
- Vorstellung von Lösungen der Mitglieder sowie anderer externer Lösungen

(binäre) XML-Lösungen basierend auf

- XML Infoset
- XML Schema
- Hybridlösung: Mischung aus Infoset- oder Schemabasierter Lösung und Standardkompressionsverfahren bzw. einer individuellen Kodierung

Ergebnis des W3C Workshops - Februar 2005

Binäres XML wird auf jeden Fall benötigt

- Als Optimierung für bestehende Systeme
- Als einzig möglich Lösung (bei ressourcenbeschränkten Geräten)

Eine einheitliche Lösung die allen Anforderungen genügt gibt es nicht

- Eine Infoset & Schema basierte Lösung könnte die meisten Anforderungen decken wenn Redundanzbasierte Verfahren optional unterstützt werden

Weitere Arbeit für einen Standard ist nötig

- Forum für weitere Diskussion einrichten
- Neue Arbeitsgruppe die sich weiter mit Anforderungen befassen sollte
- Arbeitsgruppe soll keine Spezifikation erstellen, sondern sich mit Use-Cases, Anforderungen, Funktionalität, Performance und vorhandenen Lösungen beschäftigen

Anforderungen

Unsere Anforderungen

- schnell und wenig Speicher (kurze Laufzeit, Akku schonen)
- MIDP kompatibel
- zu allen XML Dokumenten ist das Schema verfügbar
- kein "mixed content"

Weitere Anforderungen (W3C)

- es gibt zu viele

Algorithmic Properties	Format Properties		
Processing Efficiency	Accelerated Sequential Access	Format Version Identification	Robustness
Small Footprint	Compactness	Fragmentable	Roundtrip Support
Space Efficiency	Content Type Management	Generality	Schema Extensions and Deviations
	Deltas	Human Language Neutral	Schema Instance Change Resilience
	Directly Readable and Writable	Human Readable and Editable	Self Contained
	Efficient Update	Integratable into XML Stack	Signable
	Embedding Support	Localized Changes	Specialized codecs
	Encryptable	No Arbitrary Limits	Streamable
	Explicit Typing	Platform Neutrality	Support for Error Correction
	Extension Points	Random Access	Transport Independence

Einführung

Warum Kompression?

Kompressionsverfahren

Lösungen

Fazit

Links

Kompressionsverfahren

Verkürzte Tags und Attributnamen

- normaler Parser/Writer
- Mapping zwischen langen und kurzen Tags/Attributnamen

Stream-Kompression (gzip)

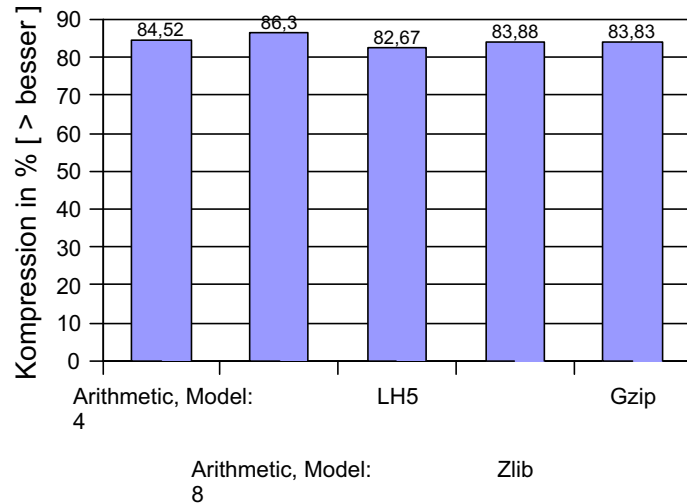
- normaler Parser und Writer
- Kompression mit zip auf XML-Stream

"Intelligente" Kompression

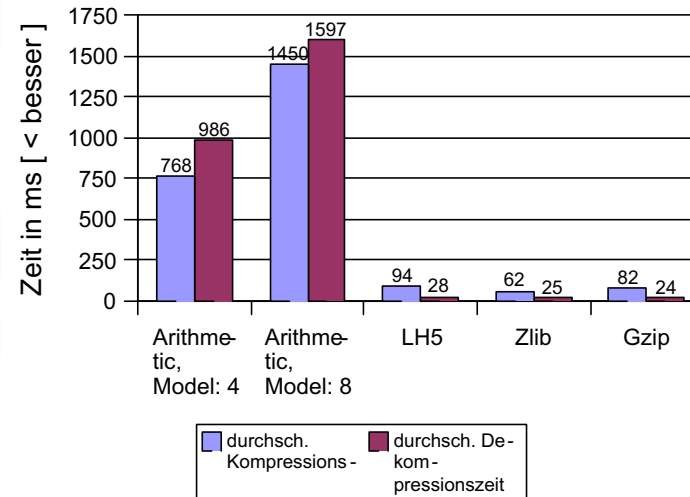
- Binäre Tags und Attributnamen
- Struktur-Symbole (Anführungszeichen, Spitze Klammern, Leerzeichen, Umbrüche) weg lassen
- String-Wiederholungen

Stream-Kompression (z. B. gzip)

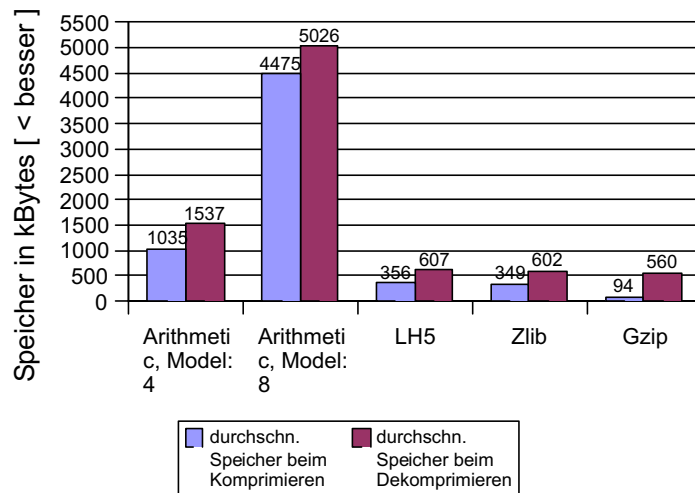
Durchschnittliche Kompressionsrate



Durchschnittliche Kompressions- & Dekompressionszeit



Durchschnittlicher Speicherverbrauch beim Komprimieren & Dekomprimieren



Vorteile

- hohe Kompression
- einfach zu implementieren
- Standard
- Frei
- Teilweise auch für MIDP

Nachteile

- zusätzlicher Speicher und CPU-Bedarf beim Lesen und Schreiben

"Intelligente" Kompression

Kompressionsrate

- viele Verfahren, meistens nicht frei verfügbar
- keine Tests möglich (wenn dann nur gegen SAX Parser vergleichbar)
- Herstellerangaben sind oft wenig aussagegekräftig

Vorteile

- Speicherbedarf und CPU-Last eher geringer als beim normalen Parser

Nachteile

- aufwändigere Implementierung
- meist SAX-Schnittstelle- für uns unbrauchbar
- kein Standard - W3C bemüht sich gerade darum

Einführung

Warum Kompression?

Kompressionsverfahren

Lösungen

Fazit

Links

Überblick Bibliotheken

Standardkompressionsverfahren

- GZip aus JDK (fehlt in MIDP)
- Arithmetische Kompression
- LHA
- Zlib

„Intelligente Kompressionsverfahren“

- ASN.1 Kodierung aus der Telekommunikationsindustrie
- Xmill AT&T und University of Pensilvania
- BinXML Expway
- XBIS Dennis Sonoski

Unsere Lösung

binäre Tags und Attribute (Index)

String-Wiederholungen (Tabelle)

Messergebnisse

	loops	Zeit KXML2	Zeit in ms: normaler Parser	Zeit in ms: binärer Parser
mit Option: -Xint	1	25	65	30
	50	25	34	23
ohne Option: -Xint	1	25	154	28
	50	25	9	8

Einführung

Warum Kompression?

Kompressionsverfahren

Lösungen



Fazit

Links

Fazit

Da kein Standard vorhanden kann nicht darauf zurückgegriffen werden

Eigene Lösung ist grundsätzlich die beste weil

- sie dem verwendeten Framework am nächsten liegt
- ausserdem sind zusätzliche Optimierungen möglich, bei externen Lösungen ist dies nicht unbedingt möglich.

Alternative: OpenSource Lösung anpassen oder SDK kaufen

Unsere Lösung: Tag- und Attributnamen aus Schema lohnt nur für kleine Dateien

Einführung

Warum Kompression?

Kompressionsverfahren

Lösungen

Fazit

Links

Links (1)

Report From the W3C Workshop on Binary Interchange of XML Information Item Sets

<http://www.w3.org/2003/08/binary-interchange-workshop/Report.html>

Binary XML Properties:

<http://www.w3.org/TR/xbc-properties/>

W3C Binary XML Workshop, Zusammenfassung der Ergebnisse:

http://www.idealliance.org/papers/dx_xml03/papers/05-01-02/05-01-02.pdf

XML Infoset

<http://www.w3.org/TR/xml-infoset>

XML Schema

<http://www.w3.org/XML/Schema#dev>

Links (2)

Arithmetische Kodierung (Java, frei)

<http://www.collopuial.com/ArithmeticCoding>

LHA (Java, frei)

<http://homepage1.nifty.com/dangan/Content/Program/Java/jLHA/LhaLibrary.html>

Zlib (Java, frei)

<http://www.jcraft.com/jzlib/>

Gzip

<http://www.gzip.org>

ITU-T X.680 (07/2002) Abstract Syntax Notation One (ASN.1): Specification of basic notation

<http://www.itu.int/ITU-T/studygroups/com17/languages/X.680-0207.pdf>

Links (3)

ITU-T X.690 (07/2002) ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and Distinguished Rules (DER)

<http://www.itu.int/ITU-T/studygroups/com17/languages/X.690-0207.pdf>

ITU-T X.691 (07/2002) ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)

<http://www.itu.int/ITU-T/studygroups/com17/languages/X.691-0207.pdf>

ITU-T X.694 (01/2004) ASN.1 encoding rules: mapping W3C XML schema definitions into ASN.1

<http://www.itu.int/ITU-T/studygroups/com17/languages/X.694.pdf>

Links (4)

Kommerzielle ASN.1 Tools

OSS Nokalva, <http://www.oss.com>

Objective Systems, <http://www.obj-sys.com>

Fast Infoset

<http://www.java.sun.com/developer/technicalArticles/xml/fastinfoset/>

Fast Web Services

<http://www.java.sun.com/developer/technicalArticles/WebServices/fastWS/>

XMILL

<http://www.cs.washington.edu/homes/suciu/XMILL>

BinXML Expway

<http://www.expway.com>

Links (5)

XBIS

<http://www.sourceforge.net/projects/xbis/>

Kxml

<http://www.sourceforge.net/projects/kxml>