

Requirements and Design for XML Messaging in the Mobile Environment

Workshop on Next Generation Networking Middleware 2005

Jaakko Kangasharju Tancred Lindholm Sasu Tarkoma

Helsinki Institute for Information Technology

May 6, 2005

Introduction

- ▶ XML a standard way to represent structured data, even in previously unconsidered areas like Web services (SOAP)
- ▶ Analysis and measurements indicate that SOAP is not suitable for the mobile environment and also point possible improvements
- ▶ An alternate XML serialization format to reduce bandwidth use and processing time
- ▶ New programming interfaces for XML processing needed to leverage improvements
- ▶ A suitable message transfer protocol is vital in the mobile environment

Requirements for the XML Processing API

- ▶ XML in messaging systems only an interchange format
- ▶ API needs to support streaming of XML into and out of application data structures
 - DOM rejected (XML in memory)
- ▶ Natural structured data handling needed for easy conversion of application data (marshaling and unmarshaling)
- ▶ Standardized input (parsing) and output (serialization) APIs
 - SAX rejected (push-based, no output)

XML as Sequences

- ▶ Despite issues with SAX, its event-based view is a good one; use pull-based XmlPull instead of SAX
- ▶ An XML document can be viewed as the sequence of events that an XmlPull parser produces reading it
- ▶ Each event will contain a certain number of strings (e.g. namespace and local name for element start, a value for content)
- ▶ Sequence-based view very natural when considering marshaling and unmarshaling

Implementation

- ▶ Add a new event type for representing typed data, partly driven by the alternate serialization format
- ▶ Different ways to construct sequences: serializer API, parser wrapper, list-like API
- ▶ Transformation sequence a wrapper on an existing sequence, permits arbitrary alternate views of a sequence
- ▶ Transformation sequences used to construct typed events for structured data, and for filtering

Requirements for the XML Serialization Format

- ▶ XML Binary Characterization Group of W3C produced use case Web Services for Small Devices with required properties
 - ▶ Small Footprint, Streamable, Compactness, Processing Efficiency, Directly Readable and Writable
- ▶ Additional useful properties
 - ▶ Schema Extensions and Deviations (open content) or Schema Instance Change Resilience, Specialized Codecs
- ▶ For Compactness, document analysis (e.g. gzip) not feasible on small mobile devices

A Simple Binary Format

- ▶ Direct serialization of the event sequence representation
- ▶ Each event serialized as a token identifying its type, followed by strings contained in the event
- ▶ Assign a small binary token to each string on first appearance, later appearances just refer to this token
- ▶ Serialize primitive typed data in binary form

Utilizing Schema Information

- ▶ Available schema information is feasible to utilize even in the mobile environment
- ▶ Construct automata, one for each side, from a schema
- ▶ Serialization automaton omits events deducible from schema, parsing automaton inserts these back
- ▶ Other methods achieve more compactness, but at the cost of Schema Extensions and Deviations

Requirements and Design for the Message Transfer Protocol

- ▶ HTTP's request-response was appropriate for RPC, not current messaging with SOAP
- ▶ Communication needs to be possible in one direction, but from both ends of a connection
- ▶ HTTP headers are too large, compact headers are required
- ▶ The header-body split of HTTP is a useful abstraction; headers can be used to carry metadata needed to implement advanced functionality

Implementation

- ▶ A single full-duplex connection between peers, static capabilities negotiated at beginning
- ▶ Connection divided into pipes, each having its own semantics
- ▶ Header names two characters long, both mnemonic and compact
- ▶ Features
 - ▶ Persistent connections across mobility
 - ▶ Sequence numbering for ordered delivery
 - ▶ Message bundling to reduce header overhead
- ▶ Implemented on top of HTTP using empty requests to permit the HTTP server to send at arbitrary times

XML Serialization Measurements

	XML	XMLZ	Basic	Schema
Size (B)	3033	674	807	493
	Parsing			
Time (ms)	1.37	1.56	1.03	0.78
Memory (KB)	17.68	17.11	10.76	11.13
	Serialization			
Time (ms)	1.76	2.80	0.46	0.40
Memory (KB)	133.94	136.96	38.92	33.05

Protocol Overhead Measurements

	HTTP	PHTTP	Proto
Data (KB)	144	143	31
TCP (KB)	123	2	12
Invoc (B)	578	574	123
Proto (B)	192	278	202

Effects of the Chosen Solutions

- ▶ Largest concern is the use of an alternate serialization format for XML
- ▶ Most applications can be provided for with low-level API compatibility, though binary-awareness may bring performance benefits
- ▶ Security features require the actual serialized form, making API compatibility an insufficient solution
- ▶ Mobile device should never need to process XML if alternate format is used

Conclusions

- ▶ XML used for messaging is not incompatible with the mobile environment
- ▶ Interest in XML optimization in this and other environments is considerable
- ▶ Interoperability is not infeasible, though it may prove inefficient in some cases
- ▶ Mobile computing can be XML-based

Thank You

Questions ?