

Xebu: A Binary Format with Schema-based Optimizations for XML Data

Web Information Systems Engineering 2005

Jaakko Kangasharju Sasu Tarkoma Tancred Lindholm

Helsinki Institute for Information Technology

November 22, 2005

Introduction

- ▶ XML messaging (SOAP) increasing
- ▶ Wireless environment weak in processing power and network capabilities
- ▶ XML heavy on bandwidth and processing time
- ▶ Simple compression of XML not always suitable
- ▶ Alternate serialization formats, “binary XML”, being investigated

Basic Tokenization

- ▶ XML document a sequence of **items** (cf. Infoset), each containing strings
- ▶ Serialized form item-by-item translation of sequence
- ▶ Serialize each string either as itself or as reference (**token**) to previous appearance
- ▶ Explicit token association for strings on first appearance
- ▶ No repetition of element name at end

Codec Omission Automata

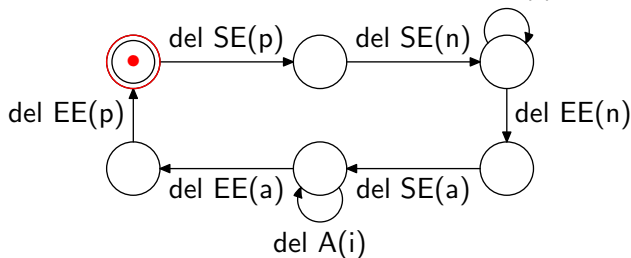
- ▶ Complete or partial schema often available
- ▶ **Pre-tokenize** strings appearing in schema
- ▶ Pre-compile schema into two **automata** translating item sequences to item sequences
- ▶ Transitions on serializer side either output or omit items
- ▶ Transitions on parser side contain omitted items, insert them at appropriate places
- ▶ Default **pass-through** transitions to handle cases not covered by schema

Processing Example

Schema	Document
<pre>start = element person { element name { xsd:string }, element age { xsd:int } }</pre>	<pre><person> <name>Alice</name> <work>Research</work> <age>30</age> </person></pre>
<p style="text-align: center;">Item sequence</p> <pre>SE(person) SE(name) TC(string, Alice) EE(name) SE(work) C(Research) EE(work) SE(age) TC(int, 30) EE(age) EE(person)</pre>	

Example Serialization

▲ SE(person) SE(name) A(type=string) TC(string, Alice)
 ▲ EE(name) SE(work) C(Research) EE(work) SE(age)
 ▲ A(type=int) TC(int, 30) EE(age) EE(person)



<0C><05>Alice

<43><00><03><04>work

<07><08>Research

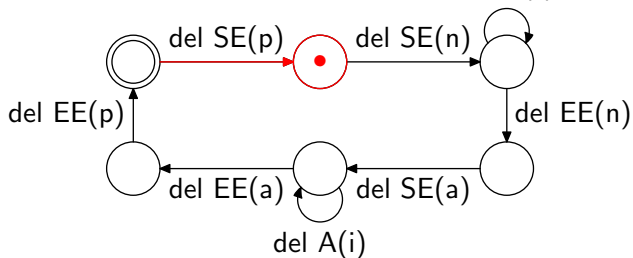
<04>

<0C><1E>

Example Serialization

SE(person) SE(name) A(type=string) TC(string,Alice)
EE(name) SE(work) C(Research) EE(work) SE(age)
A(type=int) TC(int,30) EE(age) EE(person)

del A(s)



<0C><05>Alice

<43><00><03><04>work

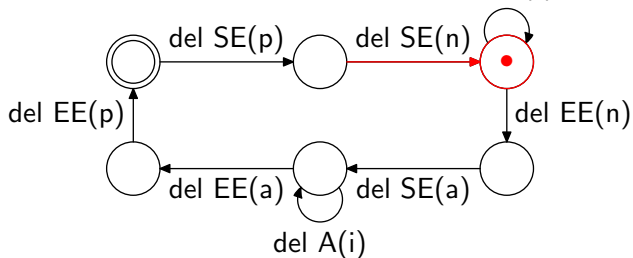
<07><08>Research

<04>

<0C><1E>

Example Serialization

SE(person) SE(name) A(type=string) TC(string, Alice)
 EE(name) SE(work) C(Research) EE(work) SE(age)
 A(type=int) TC(int, 30) EE(age) EE(person)
 del A(s)



<0C><05>Alice

<43><00><03><04>work

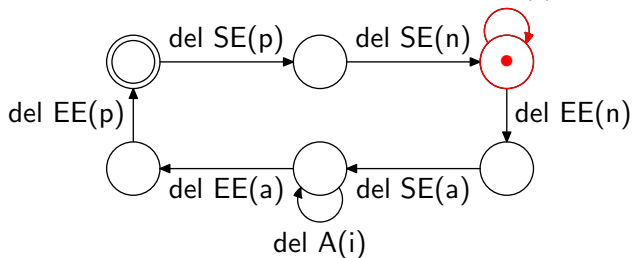
<07><08>Research

<04>

<0C><1E>

Example Serialization

SE(person) SE(name) A(type=string) TC(string, Alice)
 EE(name) SE(work) C(Research) EE(work) SE(age)
 A(type=int) TC(int, 30) EE(age) EE(person)
 del A(s)



<0C><05>Alice

<43><00><03><04>work

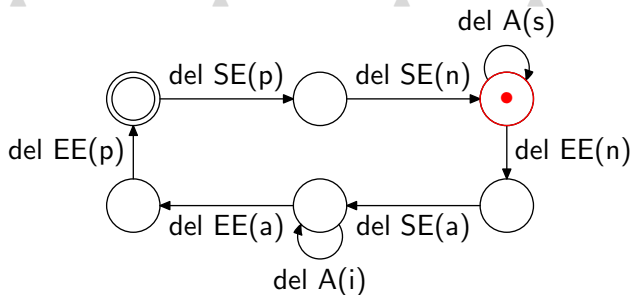
<07><08>Research

<04>

<0C><1E>

Example Serialization

SE(person) SE(name) A(type=string) TC(string, Alice)
 EE(name) SE(work) C(Research) EE(work) SE(age)
 A(type=int) TC(int, 30) EE(age) EE(person)



<0C><05>Alice

<43><00><03><04>work

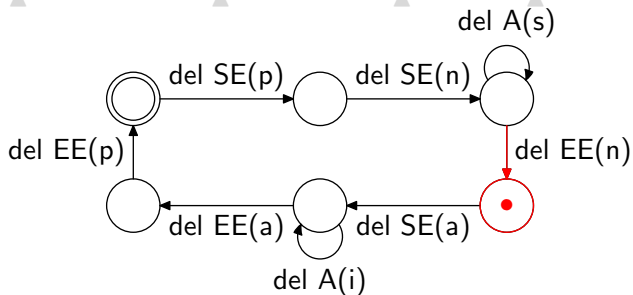
<07><08>Research

<04>

<0C><1E>

Example Serialization

SE(person) SE(name) A(type=string) TC(string,Alice)
 EE(name) SE(work) C(Research) EE(work) SE(age)
 A(type=int) TC(int,30) EE(age) EE(person)



<0C><05>Alice

<43><00><03><04>work

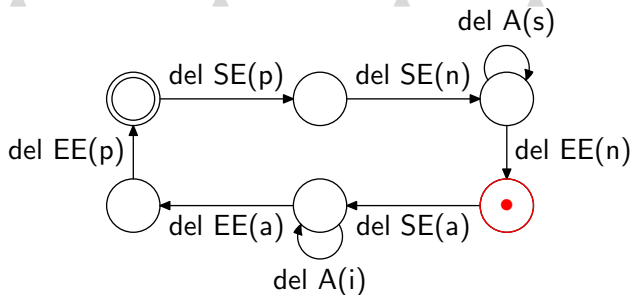
<07><08>Research

<04>

<0C><1E>

Example Serialization

SE(person) SE(name) A(type=string) TC(string,Alice)
 EE(name) SE(work) C(Research) EE(work) SE(age)
 A(type=int) TC(int,30) EE(age) EE(person)



<0C><05>Alice

<43><00><03><04>work

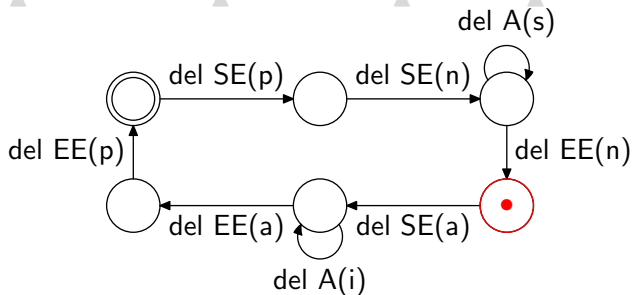
<07><08>Research

<04>

<0C><1E>

Example Serialization

SE(person) SE(name) A(type=string) TC(string, Alice)
 EE(name) SE(work) C(Research) EE(work) SE(age)
 A(type=int) TC(int, 30) EE(age) EE(person)



<0C><05>Alice

<43><00><03><04>work

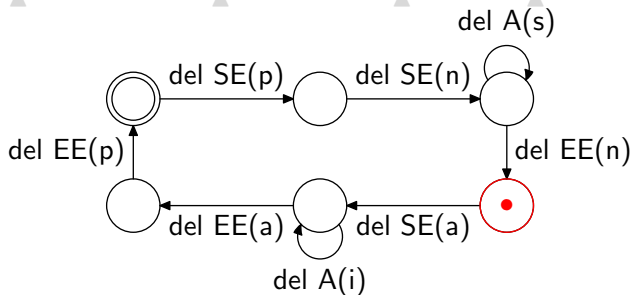
<07><08>Research

<04>

<0C><1E>

Example Serialization

SE(person) SE(name) A(type=string) TC(string,Alice)
 EE(name) SE(work) C(Research) EE(work) SE(age)
 A(type=int) TC(int,30) EE(age) EE(person)



<0C><05>Alice

<43><00><03><04>work

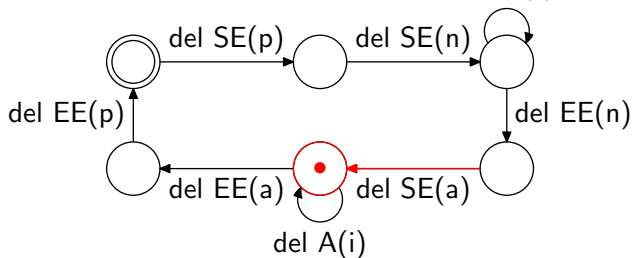
<07><08>Research

<04>

<0C><1E>

Example Serialization

SE(person) SE(name) A(type=string) TC(string,Alice)
 EE(name) SE(work) C(Research) EE(work) SE(age)
 A(type=int) TC(int,30) EE(age) EE(person)
 del A(s)



<0C><05>Alice

<43><00><03><04>work

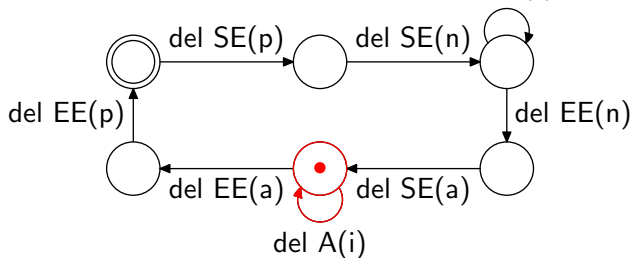
<07><08>Research

<04>

<0C><1E>

Example Serialization

SE(person) SE(name) A(type=string) TC(string, Alice)
 EE(name) SE(work) C(Research) EE(work) SE(age)
 A(type=int) TC(int, 30) EE(age) EE(person)
 del A(s)



<0C><05>Alice

<43><00><03><04>work

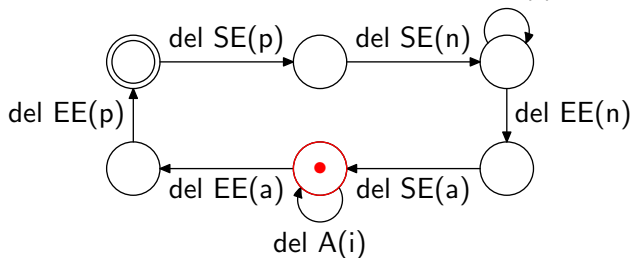
<07><08>Research

<04>

<0C><1E>

Example Serialization

SE(person) SE(name) A(type=string) TC(string, Alice)
 EE(name) SE(work) C(Research) EE(work) SE(age)
 A(type=int) TC(int, 30) EE(age) EE(person)
 del A(s)



<0C><05>Alice

<43><00><03><04>work

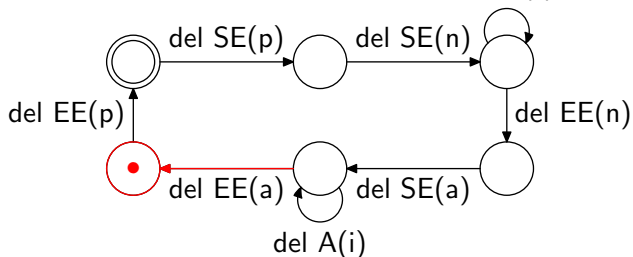
<07><08>Research

<04>

<0C><1E>

Example Serialization

SE(person) SE(name) A(type=string) TC(string, Alice)
 EE(name) SE(work) C(Research) EE(work) SE(age)
 A(type=int) TC(int, 30) EE(age) EE(person)
 del A(s)



<0C><05>Alice

<43><00><03><04>work

<07><08>Research

<04>

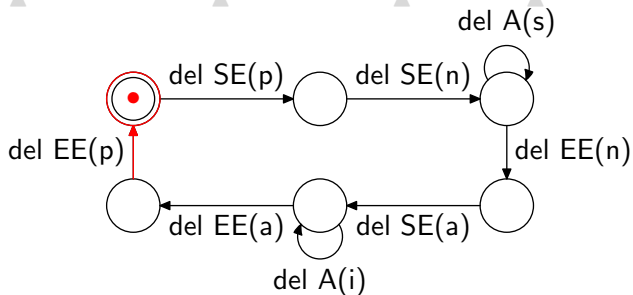
<0C><1E>

Example Serialization

SE(person) SE(name) A(type=string) TC(string,Alice)

EE(name) SE(work) C(Research) EE(work) SE(age)

A(type=int) TC(int,30) EE(age) EE(person) ▲



<0C><05>Alice

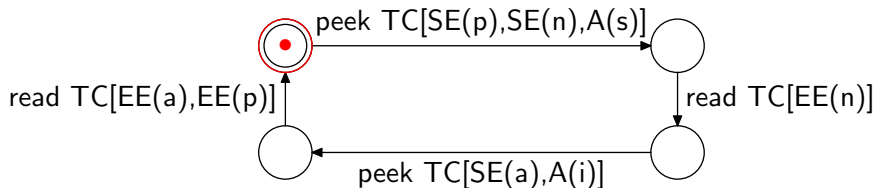
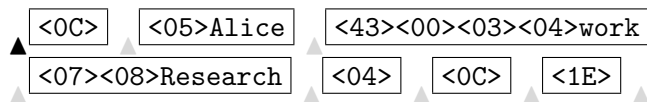
<43><00><03><04>work

<07><08>Research

<04>

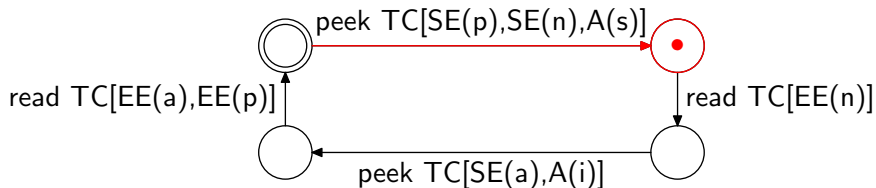
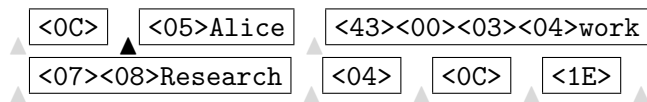
<0C><1E>

Example Parsing



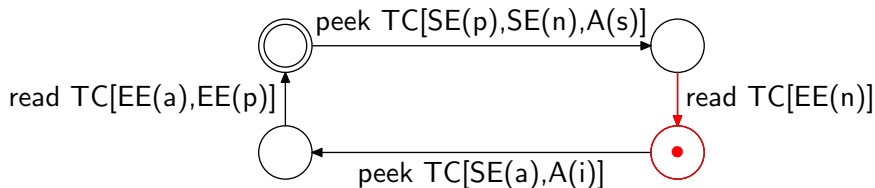
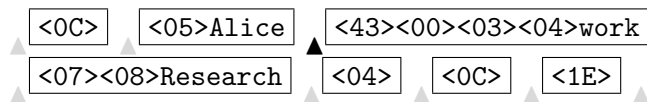
SE(person) SE(name) A(type=string) TC(string, Alice) EE(name)
SE(work) C(Research) EE(work) SE(age) A(type=int) TC(int, 30)
EE(age) EE(person)

Example Parsing



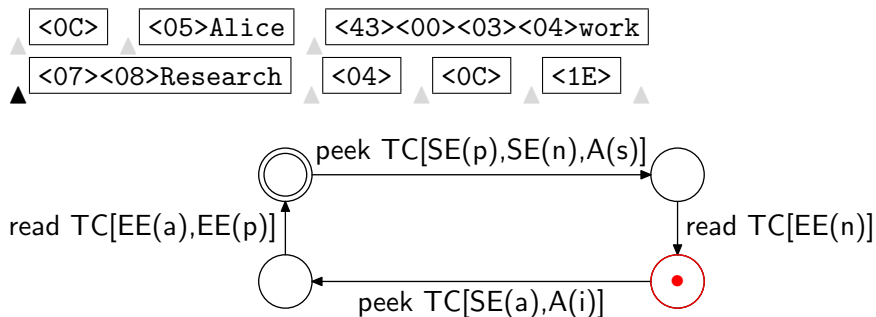
$SE(\text{person})$ $SE(\text{name})$ $A(\text{type}=\text{string})$ $TC(\text{string}, \text{Alice})$ $EE(\text{name})$
 $SE(\text{work})$ $C(\text{Research})$ $EE(\text{work})$ $SE(\text{age})$ $A(\text{type}=\text{int})$ $TC(\text{int}, 30)$
 $EE(\text{age})$ $EE(\text{person})$

Example Parsing



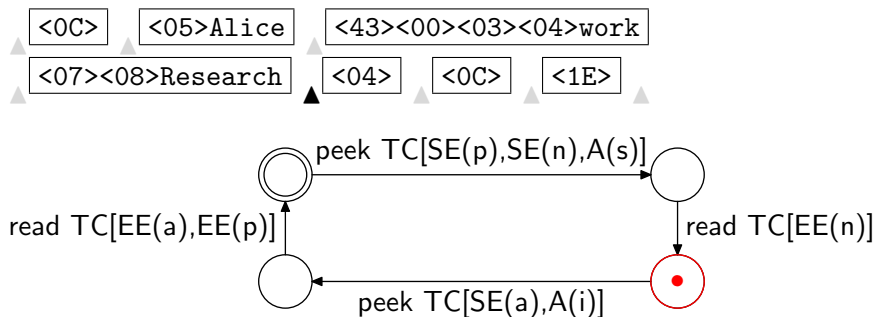
$SE(\text{person})$ $SE(\text{name})$ $A(\text{type}=\text{string})$ $TC(\text{string}, \text{Alice})$ $EE(\text{name})$
 $SE(\text{work})$ $C(\text{Research})$ $EE(\text{work})$ $SE(\text{age})$ $A(\text{type}=\text{int})$ $TC(\text{int}, 30)$
 $EE(\text{age})$ $EE(\text{person})$

Example Parsing



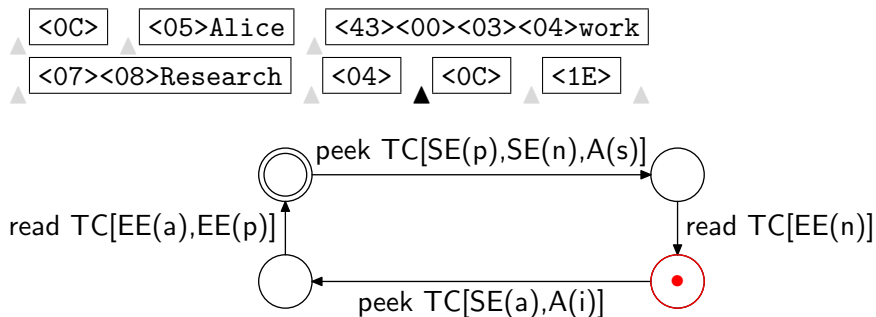
SE(person) SE(name) A(type=string) TC(string, Alice) EE(name)
 SE(work) C(Research) EE(work) SE(age) A(type=int) TC(int, 30)
 EE(age) EE(person)

Example Parsing



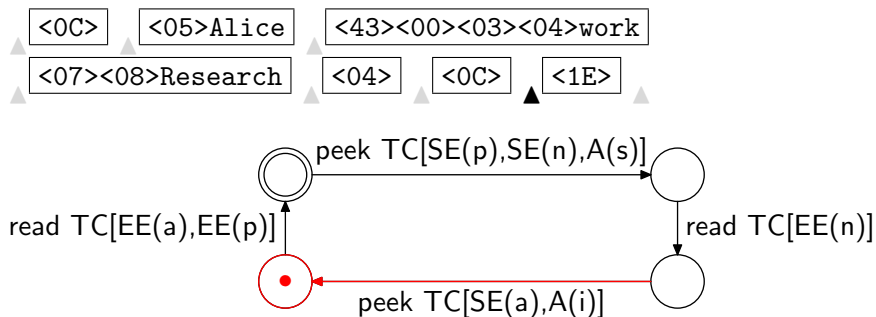
$\text{SE}(\text{person})$ $\text{SE}(\text{name})$ $A(\text{type}=\text{string})$ $\text{TC}(\text{string}, \text{Alice})$ $\text{EE}(\text{name})$
 $\text{SE}(\text{work})$ $C(\text{Research})$ $\text{EE}(\text{work})$ $\text{SE}(\text{age})$ $A(\text{type}=\text{int})$ $\text{TC}(\text{int}, 30)$
 $\text{EE}(\text{age})$ $\text{EE}(\text{person})$

Example Parsing



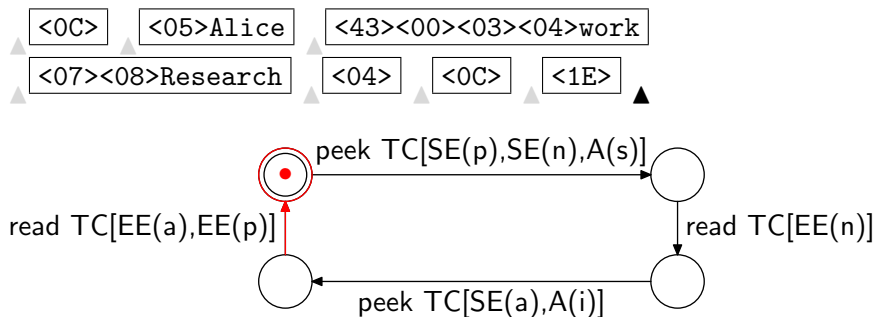
SE(person) SE(name) A(type=string) TC(string, Alice) EE(name)
 SE(work) C(Research) EE(work) SE(age) A(type=int) TC(int, 30)
 EE(age) EE(person)

Example Parsing



SE(person) SE(name) A(type=string) TC(string, Alice) EE(name)
SE(work) C(Research) EE(work) SE(age) A(type=int) TC(int, 30)
EE(age) EE(person)

Example Parsing



$\text{SE}(\text{person})$ $\text{SE}(\text{name})$ $A(\text{type}=\text{string})$ $\text{TC}(\text{string}, \text{Alice})$ $\text{EE}(\text{name})$
 $\text{SE}(\text{work})$ $C(\text{Research})$ $\text{EE}(\text{work})$ $\text{SE}(\text{age})$ $A(\text{type}=\text{int})$ $\text{TC}(\text{int}, 30)$
 $\text{EE}(\text{age})$ $\text{EE}(\text{person})$

Conclusions

- ▶ Low-level compatibility with XML useful for integration
- ▶ Binary-aware applications will need to be provided for
- ▶ Including token values allows flexibility in token replacement policies
- ▶ Rigid schema conformance not necessary to gain benefits
- ▶ Interoperability in encryption and signatures major open issue with binary XML

Thank You

Questions?

Xebu Open Source implementation:

<http://www.hiit.fi/fuego/fc/download.html>