ASN.1: A Powerful Schema Notation for XML and Fast Web Services

Olivier DUBUISSON
ITU-T ASN.1 & OID project leader
France Telecom Orange

Olivier.Dubuisson@orange-ftgroup.com
http://www.itu.int/itu-t/asn1/

http://www.itu.int/itu-t/asn1/xml/ASN1-XML-FastWebServices.pdf

Part 1:
Overview of ASN.1

Olivier Dubuisson
ITU-T ASN.1 & OID project leader

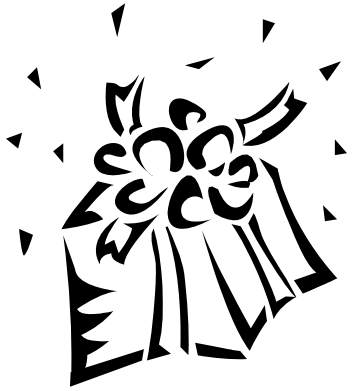
ASN.1: What is it?

- *Abstract Syntax Notation number One*
- Widely adopted International Standard (joint work between ITU-T SG17 and ISO/IEC JTC1/SC 6)
- Free of charge: http://www.itu.int/ITU-T/studygroups/com17/languages/
- Allow to choose the most suitable operating system and programming language
- Describes precisely types of data exchanged between two communicating applications
- Formal notation supported by tools:
  - no ambiguity;
  - makes validation easier, at low cost;
  - reduce the *time-to-market;*
  - readable by experts of the application domains;
  - easily understandable by implementors;
  - translates easily into any programming language (C, C++, Java, Cobol, and over 150 others)

- "*ASN.1 is a critical part of our daily lives; it's everywhere, but it works so well it's invisible!*"
- More information: http://www.itu.int/itu-t/asn1

More advantages of ASN.1

- Several associated standardized encodings, such as:
  - efficient (binary) encoding: *Packed Encoding Rules* (PER)
  - canonical encoding for digital signatures: *Distinguished Encoding Rules* (DER)
  - XML encoding rules (XER)
- Mature, long record of reliability and interoperability
- Sends information in any form (audio, video, data…) anywhere it needs to be communicated digitally
- Offers extensibility: interworking between previously deployed systems and newer, updated versions designed years apart
- Full and direct support of international alphabets (Unicode)
- Has evolved over time to meet industry needs

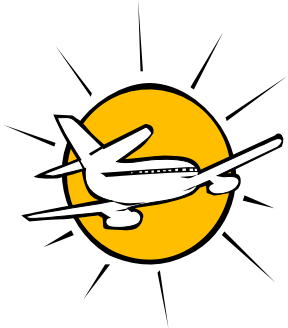


ASN.1 and the telecom industry

- Intelligent network
- GSM
- UMTS (3G cellphones)
- Voice over IP
- Videoconference (Microsoft NetMeeting®…)
- Interactive television
- Secured electronic transaction:
  e-commerce, m-commerce
- Computer-supported telecommunications applications (CSTA)
- …

Olivier Dubuisson
ITU-T ASN.1 & OID project leader

ASN.1 in other domains



- Intelligent transportation
- Radio-frequency identification (RFID)
- ATN (Aeronautical Telecommunication Network)
- Embedded systems
- Financial operations (ASC X.9, ISO TC 68)
- Control of manufacturing systems
- And a lot of other application domains:
  see http://www.itu.int/itu-t/asn1/uses

- Fully integrated in the formal languages SDL, TTCN, GDMO

Example of an ASN.1 module

```
Module-order DEFINITIONS AUTOMATIC TAGS ::=
BEGIN
Order ::= SEQUENCE {
    header          Order-header,
    items           SEQUENCE OF Order-line }
Order-header ::= SEQUENCE {
    reference       NumericString (SIZE (12)),
    date            NumericString (SIZE (8)) -- MMDDYYYY --,
    client          Client,
    payment         Payment-method }
Client ::= SEQUENCE {
    name            PrintableString (SIZE (1..20)),
    street          PrintableString (SIZE (1..50)) OPTIONAL,
    postcode        NumericString (SIZE (5)),
    town            PrintableString (SIZE (1..30)),
    country         PrintableString (SIZE (1..20)) DEFAULT "France" }
Payment-method ::= CHOICE {
    check           NumericString (SIZE (15)),
    credit-card     Credit-card,
    cash            NULL }
Credit-card ::= SEQUENCE {
    type            Card-type,
    number          NumericString (SIZE (20)),
    expiry-date     NumericString (SIZE (6)) -- MMYYYY -- }
Card-type ::= ENUMERATED {cb(0), visa(1), eurocard(2), diners(3), american-express(4)}
-- etc
END
```

Using ASN.1

ASN.1 can be used with most modern programming languages, including Java and C++, as well as older ones such as C and COBOL.

ASN.1:

```
PersonalInfo ::= SEQUENCE {
        married BOOLEAN,
        age     INTEGER (123456..124000),
        name    PrintableString}
```

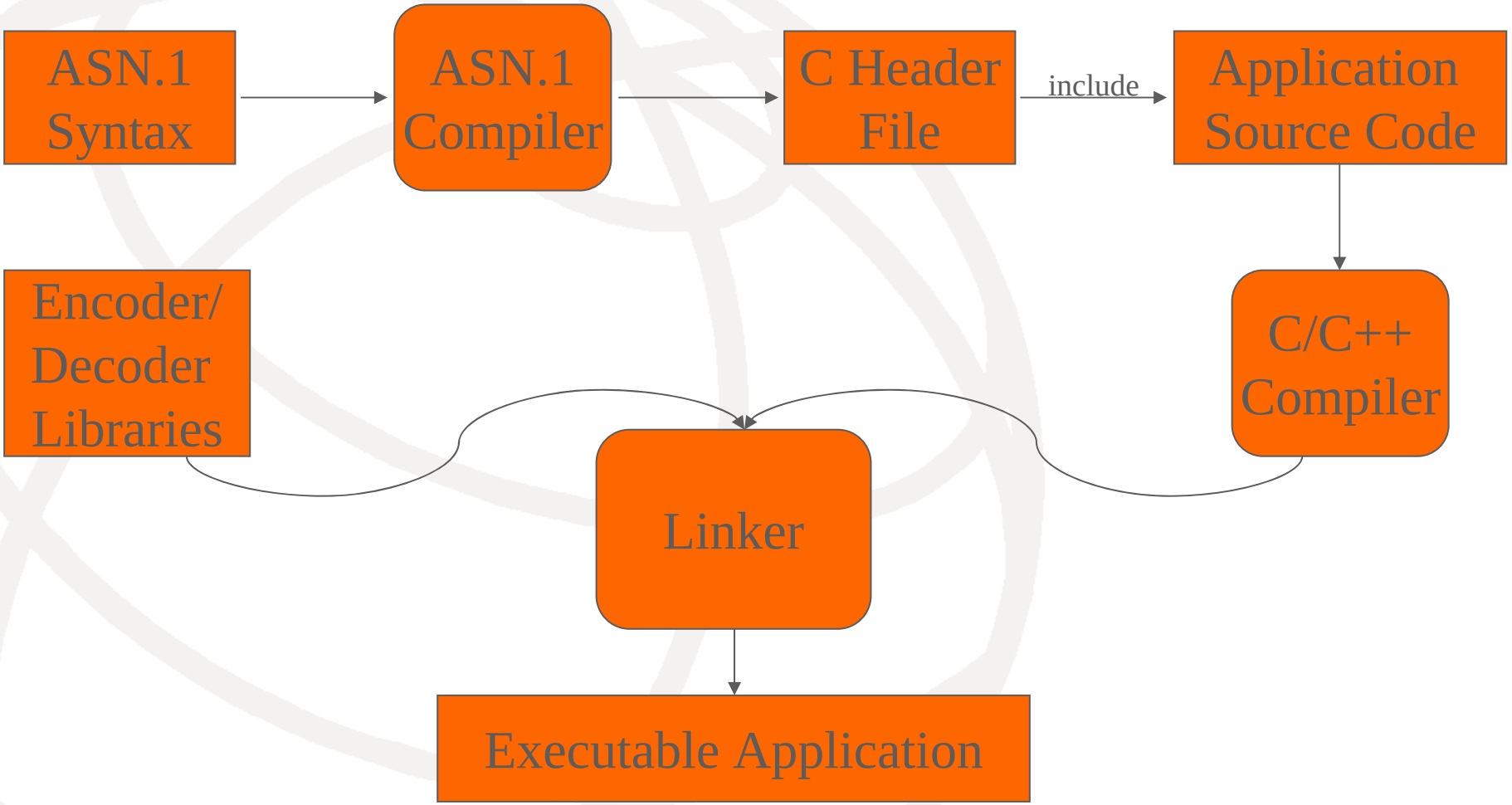
Generated C header file:

```
typedef struct PersonalInfo {
    boolean married;
    int     age;
    char    *name;
} PersonalInfo;
```

Encoding/decoding:

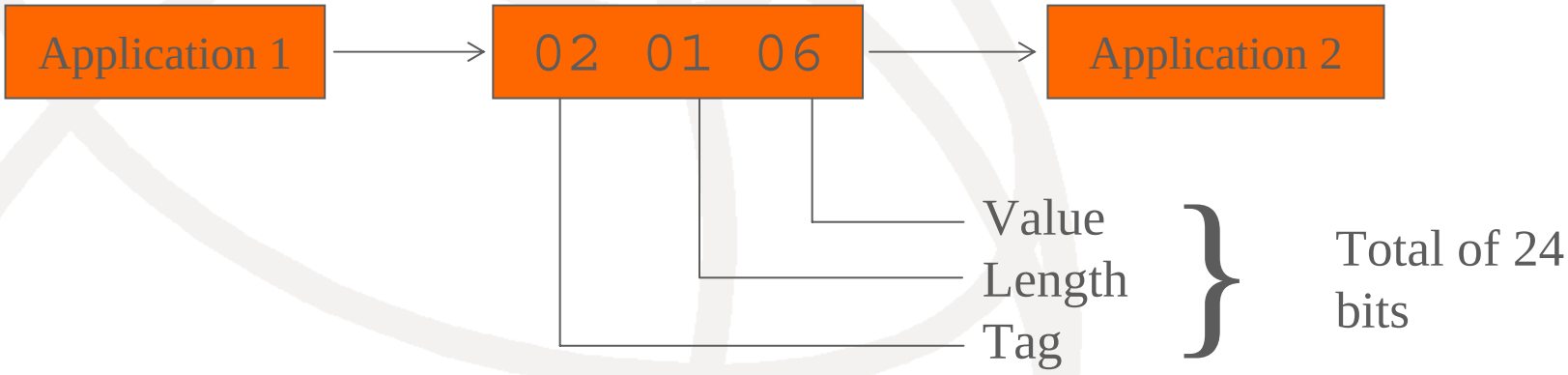
```
encode(world, PersonalInfo_PDU,
        &inBuf, &outBuf);
decode(world, &pdunum, &inBuf,
        &outBuf);
```

Development Work Flow



What are the Basic Encoding Rules (BER)?

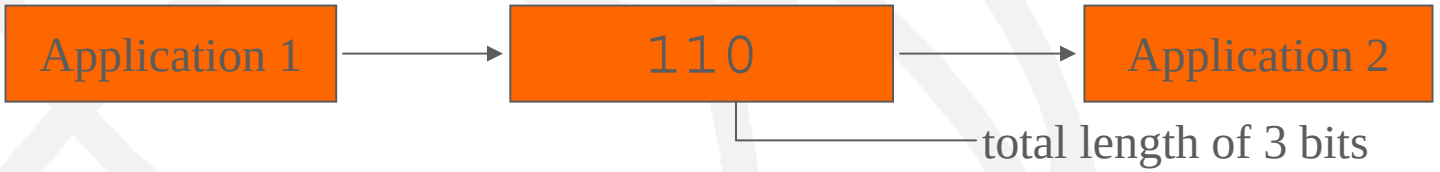
```
Age ::= INTEGER (0..7)
firstGrade Age ::= 6
```



- BER specifies how data should be encoded for transmission, independently of machine type, programming language, or representation within an application program.
- BER is highly structured, prefixing all values with a tag and a length.

What are the Packed Encoding Rules (PER)?

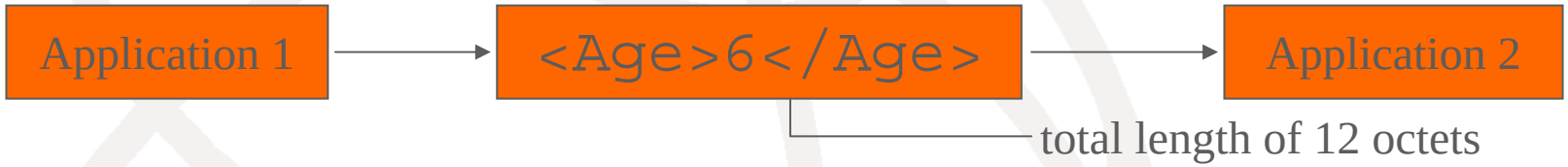
```
Age ::= INTEGER (0..7)
firstGrade Age ::= 6
```



- Like BER, PER specifies how data should be encoded for transmission, independently of machine type, programming language, or representation within an application program.
- Unlike BER, tags are never transmitted, while lengths and values are not transmitted if known by both peers.
- PER's reason for existence is to conserve bandwidth. It is valuable in audio and video over the Internet, air-ground communication, radio-paging, or wherever bandwidth is at a premium.

What are the XML Encoding Rules (XER)?

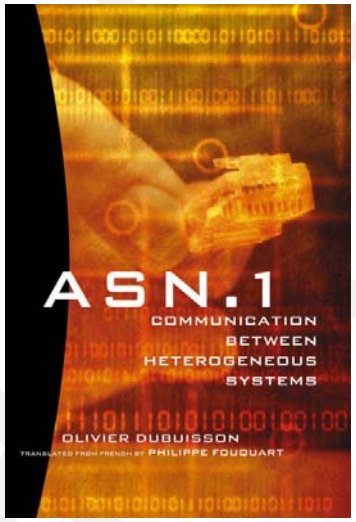
```
Age ::= INTEGER (0..7)
firstGrade Age ::= 6
```



- Just like BER and PER, XER also specifies how data should be encoded for transmission, independently of machine type, programming language, or representation within an application program.
- Unlike the more compact encoding rules, XER is immediately legible.
- XER's reason for existence is ease of legibility (no tools are needed), but…
- XER uses significantly more bandwidth.

To find more information

- ASN.1 website: http://www.itu.int/itu-t/asn1/



– *ASN.1 - Communication Between Heterogeneous Systems*
(Olivier Dubuisson, Morgan Kaufmann, 2001)
download : http://www.oss.com/asn1/dubuisson.html

– *ASN.1 Complete*
(John Larmouth, Morgan Kaufmann, 2000)
download : http://www.oss.com/asn1/larmouth.html

Some free tools

- OSS Nokalva syntax and semantics checker:
  - http://www.oss.com/products/checksyntax.html
- ITU-T ASN.1 module database:
  - http://www.itu.int/itu-t/recommendations/fl.aspx
- Object identifier (OID) repository:
  - http://www.oid-info.com/

Part 2:
ASN.1, a schema notation for XML

Olivier Dubuisson
ITU-T ASN.1 & OID project leader

April 2004

Using ASN.1 as a schema notation for XML (1)

- Compare:

```
LineItem ::= [XER:UNCAPITALIZED] SEQUENCE {
        part-no   INTEGER,
        quantity  INTEGER }
```

with:

```
<xsd:complexType name="lineItem">
    <xsd:sequence>
        <xsd:element name="part-no"  type="xsd:number"/>
        <xsd:element name="quantity" type="xsd:number"/>
    </xsd:sequence>
</xsd:complexType>
```

- All information such as data types used, optional fields, and constraints must be carried in the schema in the transfer (as in XML) or understood one time only (as in ASN.1) and assumed for all subsequent transfers

Using ASN.1 as a schema notation for XML (2)

- ASN.1 provides a reliable and easy way to manage extensibility (the need for version 1 systems to be able to interwork with yet-to-be-defined version 2 systems) and versioning of schemas. This is done with help of extensibility markers "..." and version brackets "[[ ]]":

```
LineItem ::= SEQUENCE {
                part-no     INTEGER,
                quantity    INTEGER, -- version 0
                ...,
                [[1: code   NumericString,
                     price  REAL ]]  -- added in version 1 --}
```

- Including extensibility in version 0 of the ASN.1 schema allows an application (using any version of the ASN.1 schema) to communicate with another application (based on any other version of the ASN.1 schema). An XML Schema would reject an extended XML document while the ASN.1 schema would accept it.
- The exception handling "!" can also be specified in the ASN.1 schema, so that decoders can report when they receive unexpected data

Examples of BASIC-XER encoding

```
AnyName ::= SEQUENCE {
   givenName    VisibleString,
   initial      VisibleString (SIZE (1))  OPTIONAL,
   familyName   VisibleString }
ChildInformation ::= SET {
    name        AnyName,
    dateOfBirth INTEGER (1..MAX) } -- YYYYMMDD
-------------------------------------------------
<AnyName>
   <givenName>Hubert</givenName>
   <initial>L</initial>
   <familyName>Owen</familyName>
</AnyName>
-------------------------------------------------
<ChildInformation>
   <name>
      <givenName>Lee</givenName>
      <familyName>Owen</familyName>
   </name>
   <dateOfBirth>19501003</dateOfBirth>
</ChildInformation>
```

Olivier Dubuisson
ITU-T ASN.1 & OID project leader

BASIC-XER encoding

- Whenever possible, the identifier name is used as the default markup tag. Otherwise, the user-defined type name is used:

```
Employee ::= SEQUENCE {                    -- type
             number      INTEGER(0..MAX),  -- identifier
             dateOfHire  Date }            -- identifier

<Employee>
  <number>51</number>
  <dateOfHire>19710917</dateOfHire>
</Employee>
```

- No use of attributes, no use of lists, no use of xsi:type or xsi:nil, no support for namespaces: simple and easy!
- But, with EXTENDED-XER, this can be changed with help of XER encoding instructions:

```
Employee ::= [XER:UNCAPITALIZED] SEQUENCE {
             number      INTEGER(0..MAX),
             dateOfHire  Date }
```

- Defined in ITU-T X.693 (2008) | ISO/IEC 8825-4:2009

XER encoding instructions

- Allow use of XML attributes vs. elements, lists for `SEQUENCE OF` values, mixed content, removing wrappers around multiple occurrences of things, `CHOICE`s corresponding to XSD type derivation hierarchies, `CHOICE`s corresponding to XSD element substitution groups or XSD unions, support of wildcards …
- As an encoding section at the end of the ASN.1 module (see below) or as encoding instructions between square brackets in front of type expressions (see previous slide)
- Overlook of the syntax:

```
MyModule DEFINITIONS AUTOMATIC TAGS ::= BEGIN
……
ENCODING-CONTROL XER
  NAME ALL AS UNCAPITALIZED
  NAME ALL AS PREFIX "xyz"
  NAME Type1.field1 AS "fred"
  ATTRIBUTE attr1, attr2 IN Type2
  LIST Type3.seqOfInteger                -- space-separated list
  TEXT Bool, Enum, BitString, Integer —- instead of <true/>…
  NAMESPACE Type4 AS "http://www.w3.org/2001/XMLSchema-instance"
    PREFIX "xsi"
  -- other properties available… --
END
```

- `MODIFIED-ENCODINGS` causes the ASN.1 XML encodings to be the same as XSD encodings for the same type
- More information: http://www.oss.com/products/exer-whitepaper.pdf
- EXTENDED-XER is defined in ITU-T X.693 (2008) | ISO/IEC 8825-4:2009

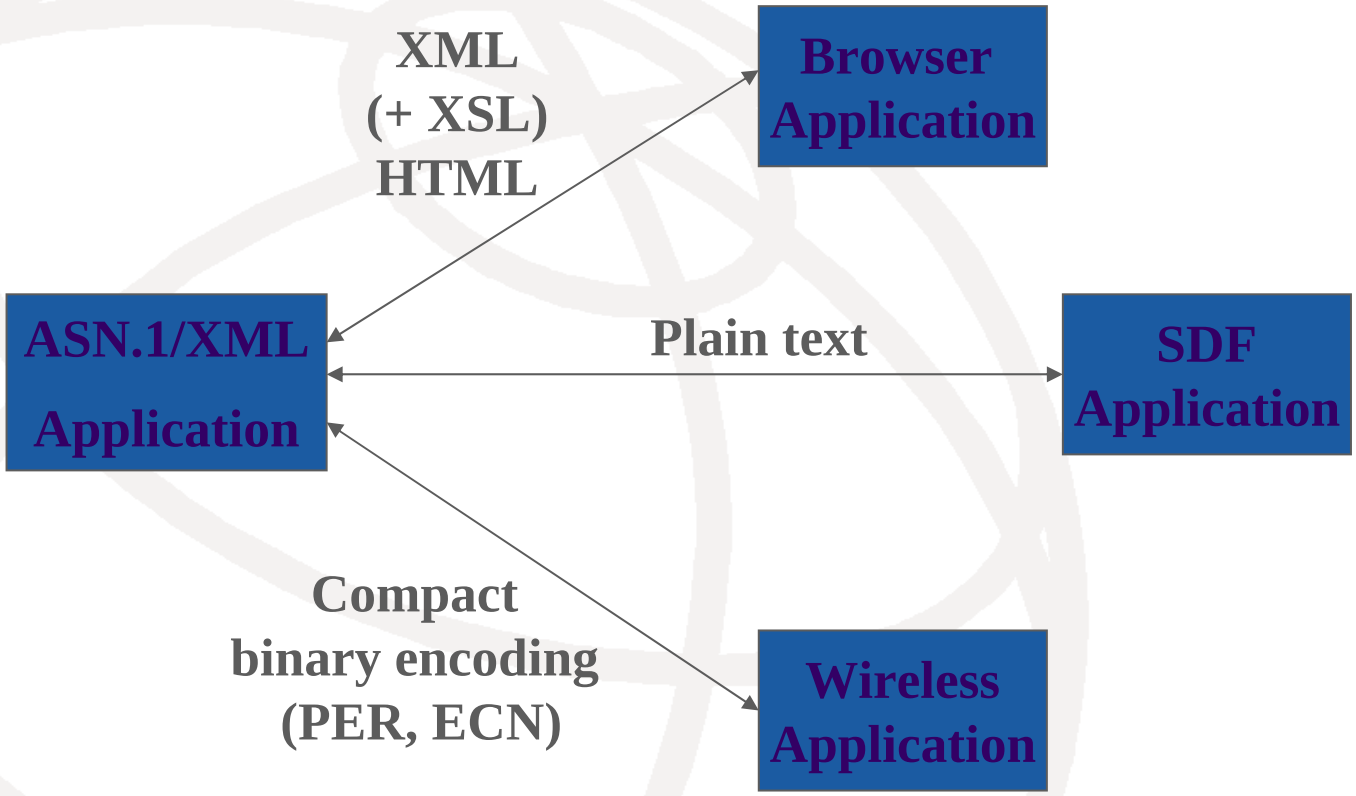
XER benefits (1)

- The ASN.1 module is the schema to validate data
- A single schema for all values:
  - Eliminates multiple schema mappings
  - ASN.1 is a mature, stable schema for XML markup
- Exploits the very real advantages of both techniques (win-win solution):
  1. Allows a closer integration of XML schema specification languages and of traditional tools for protocol implementation
  2. Displays (BER- or PER-encoded) ASN.1-data in an XML-browser
  3. ASN.1 applications can send and receive XML values
  4. Efficient (BER, PER, ECN) encoding for transmission:
     - XML has no standardized associated encoding that is very efficient and error-proof
       (Binary XML: not so efficient, jeopardized evolution, interoperability problems)
  5. Canonical (DER, CXER) encoding for secure messages (encryption, digital signatures…):
     - XML doesn't support an easy-to-use canonical encoding
- More information: http://www.itu.int/itu-t/asn1/xml

"Performance tests of XML and ASN.1 found that signed complex XML messages can be up to 1,000 percent slower to decode than an equivalent ASN.1 message."

An XML Alternative for Performance and Security: ASN.1, by D. Mundy, & D.D. Chadwick (IT Profess. Mag., Vol. 6, No. 1, Jan.-Feb. 2004, pp. 30-36)

XER benefits (2)



Using ASN.1 as a binary encoding for XML (1)

- No tree information is stored in the binary encoding.
- If a binary encoding was pointless, why would have areas like MPEG7 or WAP created their proprietary solution?
- On average the compression rate is better than tools like zip (see http://lists.xml.org/archives/xml-dev/200107/msg01248.html).
- High-throughput transaction processing systems, low-bandwith communications and low-power processors with small memory are not generally places where complex compression algorithms are worthwhile.
- The application is not slowed down because there is no need to build a dictionary on the fly (the encoder&decoder have been generated once for all).
- Some people say that the major drawback is that a binary encoding is not readable (unless it is converted back to XML markup), however this is not necessary for many applications where most messages are never read by humans.

Using ASN.1 as a binary encoding for XML (2)

```
AnyName ::= SEQUENCE {
   givenName    VisibleString,
   initial      VisibleString (SIZE (1))  OPTIONAL,
   familyName   VisibleString }

<AnyName>
   <givenName> Hubert </givenName>
   <initial> L </initial>
   <familyName> Owen </familyName>
</AnyName>
```

Mapping XML Schemas into ASN.1 modules

- Automatic translation (mapping) of XML Schemas into ASN.1 modules
- The **same** XML document is rendered by the ASN.1 decoder
- Compact (PER, ECN) or canonical (DER) encodings then available for data initially described in XML (test results -- in progress)
- Improves speed: decoding a binary stream improves performance by a factor 100 or more
- Decreases size: a binary encoding may save up to 80% or even more relative to corresponding XML text.
- Defined in ITU-T X.694 (2008) | ISO/IEC 8825-5:2009

- More information: `http://www.itu.int/itu-t/asn1/xml/#Mapping`

Olivier Dubuisson
ITU-T ASN.1 & OID project leader

In a nutshell

- ASN.1 + EXTENDED-XER is as powerful and expressive as XSD (XML Schema Definition), but less verbose and much more readable.
- ASN.1 is a mature schema notation for XML. No schema mapping is needed
- ASN.1/XER offers efficient binary encodings together with XML (+ XSL) display (or transfer if needed)
- There is a canonical variant of XER for secured transactions called CXER
- ASN.1/XER should be in the picture for high-throughput transaction processing systems, low-bandwidth communications and low-power processors with small memory where compression (zipping) cannot be the answer
- An XER-decoder is just another name for an XML parser

Press releases

- *An XML Alternative for Performance and Security: ASN.1, by D. Mundy, & D.D. Chadwick (IT Professional Magazine, Vol. 6, No. 1, Jan.-Feb. 2004, pp. 30-36)*
  "Performance tests of XML and ASN.1 found that signed complex XML messages can be up to 1,000 percent slower to decode than an equivalent ASN.1 message."
- *Binary Showdown*, by M. Leventhal, E. Lemoine, & S. Williams (XML-Journal, Dec. 1, 2003)
- *Binary Killed the XML Star?*, by K.G. Clark (XML.com, Oct. 19, 2003)
- *ASN.1, a schema language for XML* (whitepaper by OSS Nokalva, Oct. 2003)
- *ASN.1 is getting sexy again!* or *ASN.1, XML, and Fast Web Services*, by J. Larmouth (at ETSI MTS#37, Oct. 14-16, 2003)
- ITU releases latest version of ASN.1 (Convergence Plus, May 30, 2003)
- *Comparing the Performance of ASN.1 vs XML*, by D.P. Mundy, D.D. Chadwick, & Andrew Smith (Terena Networking Conference, Zagreb, May 2003)
- *Leading ITU specification language delivers major advances. New version of ASN.1 incorporated in biometrics and banking standards* (ITU-T press release, Geneva, April 25, 2003)
- *The emergence of ASN.1 as an XML schema notation* (XML Europe 2003 conf., London, May 5-8, 2003)
- *OASIS XML Common Biometric Format Moves Toward Standardization* (The XML Cover pages, Feb. 6, 2003)
- *Group says ASN.1 can field XML, save bandwith*, (in EE Times, Aug. 9, 2001)
  "As digital communications spreads from cell phones to wireless personal organizers and XML-powered informations appliances, [ASN.1] *claims to have the interoperability 'Rosetta stone' in place."*
- *ASN.1 and XML Messaging* (in Dr. Dobb's Journal, Jul. 17, 2001)
  "*It makes it possible for XML to go places that right now it just cannot go. Specifically,* [the ASN.1 group] *expects that cellphones, PDAs, and other constrained devices will want to use ASN.1 to transform XML messages into a bandwith-friendly for transmission.*"

Part 3:
Fast Web Services

Fast Web Services is the term applied to the use of ASN.1 to provide message exchanges based on a SOAP envelope and WSDL specification of services that can have a higher transaction-processing rate and less bandwidth requirements than use of a character-based XML representation.

Olivier Dubuisson
ITU-T ASN.1 & OID project leader

Goals for Fast Web Services

- SOAP rpc/encoded is slow
- Cut overhead of XML processing
- Get Web service latency and message size close to Sun's RMI (5x faster, 5x smaller)
- Improve Web services performance while maintaining the advantages of SOAP, WSDL and associated technologies
- Use ASN.1 and PER (Packed Encoding Rules), the most compact and CPU efficient standardized encoding rules
- International Standard for interoperability
- Minimize impact to Web service developers: same APIs as traditional Web services, platform and programming language independent
- Defined in ITU-T X.892 | ISO/IEC 24824-2

- More information: http://www.itu.int/itu-t/asn1/xml/#FastWeb

Fast Web Services annotations for WSDL allow services to explicitly state that a binding can support the PER encoding (in addition to XML).

Application domains

- Time- and resource-sensitive systems:
  - Mobile phones
  - Open Mobile Alliance (define how non-intermediated Web services can be used for value-add services between small devices and servers)
  - Satellites
  - Auto-ID (processing and management of RFID systems)
- Web services within the enterprise:
  - Enterprise Application Integration
- High-performance computing:
  - Grid and scientific computing

Fast Web Services

- **Fast Web Services** is good for binding applications:
  - Uses the XML Schema to ASN.1 mapping defined in ITU-T X.694 (2008) | ISO/IEC 8825-5
  - XML Infoset layer is hidden
  - Requires schema to process message parts: WSDL is used by client and service
  - Requires a binary representation of SOAP (ASN.1 schema for SOAP)
  - Preserves the semantics of SOAP 1.2
  - Cannot be used when there is xsd:any content because the schema is unknown
  - Loss of self-description
  - Defined in ITU-T X.892 (2005) | ISO/IEC 24824-2:2006

ITU-T X.694 | ISO/IEC 8825-5 (*Mapping W3C XML Schema definitions into ASN.1*) is an important building block for Fast Web Services because XSD is used in WSDL to define the structure of messages. Consequently, the XML schema referenced in a WSDL document can be considered an abstract schema, with an equivalent ASN.1 description, whose instances can be encoded using an ASN.1 encoding such as XER or PER.

The ASN.1 schema for SOAP is based on the W3C SOAP 1.2 specification and ensures that the SOAP semantics and processing model are preserved. This enables WSDL-defined content to be efficiently encapsulated in a SOAP-based envelope using the same encoding technology. No changes to the SOAP binding syntax are required.

Fast Infoset

- **Fast Infoset** is good for generic XML processing:
  - Uses an ASN.1 schema for the XML Infoset
  - Preserves XML data
  - Can be used when there is xsd:any content
  - Provides a compact serialization of any XML Infoset for which no Schema is available, without increasing parsing time or writing time
  - Suitable for applications for which the classic compression algorithms are too costly in terms of CPU time and memory
  - Defined in ITU-T X.891 (2005) | ISO/IEC 24824-1:2007
  - Test results (work in progress)

The ASN.1 Schema for the XML information set provides the capability to encode XML content that is not described by a schema. It defines an alternative to XML 1.0 serialization and, like the ASN.1 Schema for SOAP, ensures the application of a consistent encoding technology.

Olivier Dubuisson
ITU-T ASN.1 & OID project leader

Press releases and articles

- *Binary Showdown*, by M. Leventhal, E. Lemoine, & S. Williams (XML-Journal, Dec. 1, 2003)
- *Binary Killed the XML Star?*, by K. G. Clark (XML.com, Oct. 19, 2003)
- *ASN.1 is getting sexy again!* or *ASN.1, XML, and Fast Web Services*, by J. Larmouth (at ETSI MTS#37, Oct. 14-16, 2003)
- *Report from the W3C Workshop on Binary Interchange of XML Information Item Sets* (Sept. 24-26, 2003): position papers of France Telecom, Mitre, Sun Microsystems and OSS Nokalva
- *Sun touts Fast Web Services plan. Binary encodings key to proposal*, by P. Krill (Infoworld, Sept. 19, 2003)
  See also the article in ComputerWeekly
- Fast Web Services, by P. Sandoz, S. Pericas-Geertsen, K. Kawaguchi, M. Hadley, and E. Pelegri-Llopart (Aug. 2003)
- Fast web services (JavaOne conference, June 11, 2003)