

## NEW DIRECTIONS IN ASN.1: TOWARDS A FORMAL NOTATION FOR TRANSFER SYNTAX

Colin Willcock

*Expert Telecoms Ltd, Mittlerer Pfad 26, 70499 Stuttgart Germany,*

*Phone +49 711 1398 8220*

C.Willcock@Expert-Telecoms.com

**Abstract** At present although the abstract structure of a protocol message can be formally defined, the actual format of the bits transmitted on the line or through the air cannot be formally specified. The small number of standardised encoding rules available are only defined in informal text and in many application domains such as radio interfaces, they do not satisfy today's requirements for very efficient transmission of information. This paper describes a number of the problems associated with the definition of transfer syntax and then discusses a possible solution to these issues involving the development of an advanced encoding control notation.

**Keywords:** Transfer syntax, abstract syntax, ASN.1, formal description techniques, encoding rules

### 1. INTRODUCTION

The advantages of using formal description techniques in the design and development of telecommunications standards has been established beyond doubt [3]. The existing formal description techniques include Abstract Syntax Notation One (ASN.1) [4, 5, 6, 7] for describing data and signal structures, the Specification and Description Language (SDL) [8] for definition and validation of system behaviour and the Tree and Tabular Combined Notation (TTCN) [2] for definition of test suites.

Currently there is a gap in the formal description techniques relative to the definition of encoding rules. This means that the actual transfer syntax of a system cannot be formally specified.

The development of advanced encoding control would allow the completion of the formal description infrastructure, it should also greatly extend the use of ASN.1 in protocol specifications which presently use other less formal notations due to the implied encoding restrictions of the current language.

## **2. STATE OF THE ART**

ASN.1 is a language for describing structured information, it allows a protocol designer to specify a syntax at a high level without the need to explicitly consider the representation of that syntax in terms of bits and bytes on the line. The actual representation on the line for any message can be derived by applying a set of encoding rules to the associated ASN.1 definition of that message. This combination of abstract syntax and encoding rules which defines the actual bits on the line is known as the transfer syntax.

Relative to encoding rules there are a small number of standardised sets. The first set to be standardised was the Basic Encoding Rules - BER [9]. BER provides a relatively straight-forward and verbose encoding scheme which is still used today in many applications. The relatively poor encoding efficiency of BER lead to the development and standardisation of the packed Encoding Rules - PER [10]. PER is considerably more bandwidth efficient than BER, requiring only between 40 and 60 percent of the encoded size. The encoding efficiency of PER is achieved however at the cost of a considerably more complicated encoding scheme including the use of context specific encoding. In addition to BER and PER, the Canonical Encoding Rules (CER) and the Distinguished Encoding Rules (DER) have been standardised. Both these encoding rule sets are purely restricted forms of BER.

At present the only way to define the transfer syntax for a protocol in Recommendations is to informally associate one of these standardised encoding rules with the ASN.1 abstract syntax definition by means of an ASN.1 comment or simply by stating it in writing in the non-ASN.1 text description.

The TTCN language is typically used in the definition a of test cases to prove that a particular implementation under test conforms to a specified protocol or standard. TTCN supports the use of ASN.1 for type and value definitions and in addition allows the definition of transfer syntax by use of an optional reference mechanism for encoding rule and encoding variant. With this mechanism the test specifier can reference by name the required encoding rule set and required variant of that encoding rule and associate this encoding information with a specific type.

In contrast, the SDL language which is used for specifying system behaviour, although it also supports ASN.1 for type and value definitions, has no concept what so ever for specifying the required transfer syntax. In effect this means

that since SDL cannot define the transfer syntax it can never fully specify a real world system that must communicate via cable or air interface.

In addition to the gap in formal description techniques, there are a number of other issues which point towards the need of some form of encoding control. The two areas considered in this paper are tabular based protocols and application specific encoding rules.

## 2.1 TABULAR BASED PROTOCOLS

Before the general adoption of ASN.1 most telecommunications protocols were defined in the form of tables. The tables define the name, size and order of the constituent fields of the protocol and in addition implicitly define the associated transfer syntax. An example of such a tabular definition is shown in figure 1. Tabular based definitions include many of the worlds most widely used protocols (e.g. Integrated Services Digital Networks (ISDN) basic call [11], Message Transfer Part (MTP) [12], Signalling Connection Control Part (SCCP) [13] and ISDN User Part (ISUP) basic call [14]).

With the existing encoding rules it is not possible to specify the transfer syntax of such protocols using ASN.1. This inability leads to major problems when fields from tabular defined protocols are embedded in ASN.1 protocols. An example of such embedded fields is the Intelligent Network (IN) protocol [15] which has ISUP and ISDN fields embedded inside an ASN.1 octet string as shown in the extract from the recommendation shown below.

**CalledPartyNumber ::= OCTET STRING (SIZE (minCalledPartyNumberLength..  
maxCalledPartyNumberLength))**

*– Indicates the Called Party Number. Refer to Recommendation Q.763 for encoding.*

**CalledPartySubaddress ::= OCTET STRING**

*– Indicates the Called Party Subaddress. Refer to Recommendation Q.931 for encoding.*

## 2.2 APPLICATION SPECIFIC ENCODING RULES

In many application areas the existing standardised encoding rules are insufficient either in terms of encoding efficiency or simply in terms of ease of data manipulation.

One application area where the requirement for encoding efficiency has led to the adoption of proprietary non-standardised description techniques is the specification of mobile air interfaces. Within the GSM standardisation community the perceived inadequacy of the current ASN.1 language plus standardised encoding rules has led to the development of CSN.1 (Concrete Syntax Notation 1) [1]. This notation allows the definition of a non-generic bit-wise transfer

	8	7	6	5	4	3	2	1
1	Odd/ even	Nature of address indicator						
2	NI	Numbering plan Ind.			Present. Ind.		Screening	
3	2nd address signal				1st address signal			
⋮	⋮							
n	Filler (if necessary)				n th address signal			

NOTE - When the address presentation restricted indicator indicates address not available octets 3 to n are omitted.

Figure 1 ISUP Calling Party number parameter field from Q.763

syntax, which is claimed to be considerably more efficient than ASN.1 together with PER encoding.

In addition to pure data efficiency, a number of new encoding rules have been recently proposed purely to enhance the data manipulation or mapping to the application domain. Examples of this class of encoding rules are those proposed for the eXtensible Markup Language (XML) (XML Encoding Rules XER) [17] and intelligent highways (Octet Encoding Rules OER) [16].

XML is a recommendation from the World Wide Web Consortium (W3C) to support client-server applications on the Internet. At present the encoding techniques used in communications are specific to this application and not based on any standardised information description technique. The basic idea of XER is to standardise a set of encoding rules that would allow information defined in ASN.1 to be carried in XML.

OER is based on concepts developed during the National Transportation Communications for ITS (Intelligent Transportation Systems) Protocol. The encoding rules provide a more compact transfer syntax than BER, whilst maintaining full octet alignment for all fields. This allows efficient and fast processing of the messages which is vital for this application.

At present the only solution available to the standardisation bodies relative to these proposed new encoding rules is either to standardise all these encoding

rules or incorporate these requirements as alternative forms within existing encoding standards. The former solution could lead to a plethora of new encoding standards which by definition are only intended for a small specific application domain. The latter solution could cause already complex encoding recommendations to become completely unintelligible.

### **3. PROPOSED SOLUTION**

Clearly it is desirable that any proposed solution for encoding control can at least solve the current problem areas described in clause 2. In addition any solution should try to build on the existing formal description techniques which have already proved their worth. In line with this requirement the proposed solution should provide a common link between ASN.1, SDL and TTCN to allow a universally applicable transfer syntax definition.

In general the encoding control must have the ability to specify for one or more types within an abstract syntax definition the required associated encoding scheme. The encoding control should be general enough to allow definition of entire encoding rules and in addition to allow the direct implementation of the proposed encoding control into associated tool sets the notation must be machine processable.

Considering the motivations and requirements highlighted in section 2 the proposed solution is to develop a new notation for the definition of encoding control modules.

These encoding control modules would provide the ability to define encoding tables which allow the association of particular encoding rules with particular types. The overall structure of such an encoding control module is shown in figure 2. The module consists of a header part which contains a module identifier and optionally a default encoding rule. The module body contains one or more encoder control assignments to form an encoding table.

The encoder control assignments associate various encoding attributes in the form of a property list to a single specified type.

The property list has three components to define encoding attributes associated with tag, length and value as shown below:

**<TYPE Name> ::= <Base TYPE>  
                  [<TAG Encoding Attributes>]  
                  [<LENGTH Encoding Attributes>]  
                  [<VALUE Encoding Attributes>]**

All three components within the property list are optional, if any component is omitted the property will take the default encoding rule definition for this

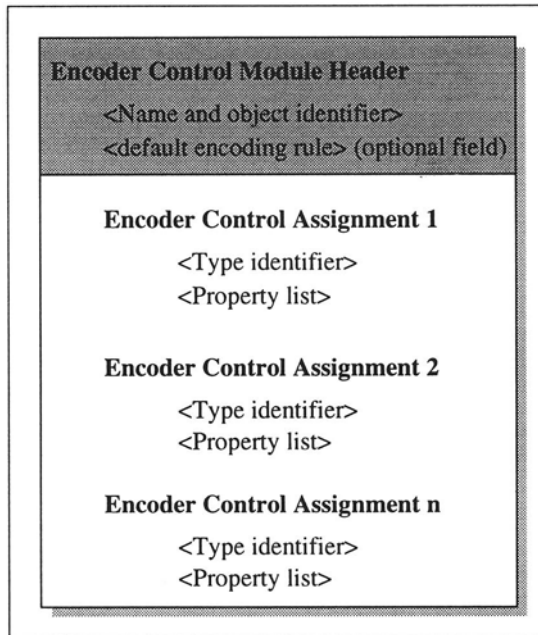


Figure 2 Encoder Control Module Abstract Format

aspect of encoding. In a similar way any type not explicitly included in the encoding table will also be encoded according to the specified default.

A provisional list of encoding attributes for the tag, length and value components is:

TAG Attributes - OMIT, PER, BER, CER,  
 LENGTH Attributes - OMIT, PER, BER, CER, ALIGNED LEFT,  
 ALIGNED RIGHT  
 VALUE Attributes - PER, BER, CER, ALIGNED LEFT, ALIGNED  
 RIGHT, OCTET ALIGNED,

The binding or association of encoding control module to abstract syntax definition could be achieved in one of two ways. Firstly it could be achieved by extension of the existing ASN.1 language to support definition and referencing of encoder control modules. Secondly a new formal notation (Transfer Syntax Notation 1 TSN1) could be defined which includes the current ASN.1 language as a subset, with added functionality for definition of encoder control modules (such a scheme is shown in figure 3).

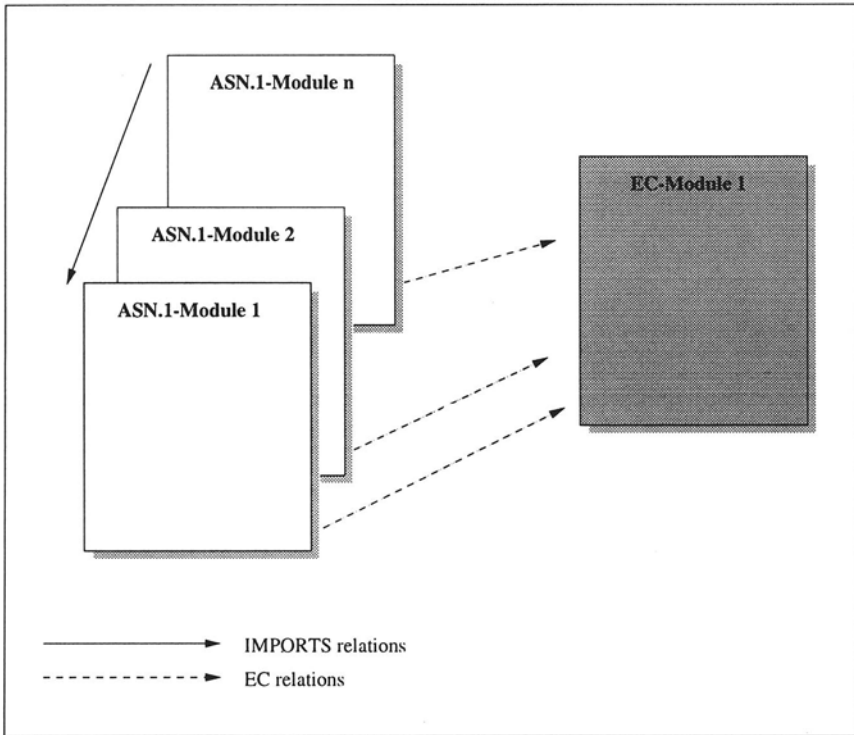


Figure 3 TSN.1 Module Structure

To illustrate the proposed scheme consider the embedded ISUP field shown in figure 1. Disregarding encoding, the field can be described in standard ASN.1 as follows

**Digits ::= SEQUENCE (SIZE(0..8)) OF Digit**

**CalledPartyNumber ::= SEQUENCE {**  
     **oddEven     OddEven,**  
     **nADI        NADI,**  
     **nI          NI,**  
     **numP        NumP,**  
     **presentInd  PresentInd,**  
     **screenInd   ScreenInd,**  
     **digits       Digits**  
     **}**

An associated encoder control module for such a type could look like:

```

SignProtocol {itu-t recommendation x 684 example(1)}
ENCODER CONTROL DEFINITIONS
DEFAULT {itu-t recommendation q 690} ::=
BEGIN
  CalledPartyNumber {
    oddEven ::=
      TAG           : OMIT,
      LENGTH        : OMIT,
      ALIGNMENT     : OCTET LEFT,
      SIZE-OF       : BIT STRING(SIZE(1));
    ...
  }

```

This example shows only the encoding part for the field oddEven. It specifies that no tag or length should be included in the encoding and that the field should be encoded in one bit with left alignment in the associated octet.

#### 4. FUTURE WORK

The development of an encoding control notation is still at a preliminary stage. The motivations and possible solutions have been discussed both within the joint ISO/ITU-T ASN.1 experts group and the ETSI (European Telecommunications Standards Institute) Methods of Testing and Specification Technical Committee (MTS-TC). In general this issue has already generated considerable interest leading to the creation of a new work item within the ETSI MTS and the proposal for a ETSI specialist task force (STF) to develop the ideas contained in this paper by defining the required notation.

The next stage towards encoding control is to define a prototype notation which matches the specified requirements. This step requires the definition of syntax and semantics. One of the main issues needing to be resolved is the set of required encoding attributes necessary to provide powerful encoding control but still maintaining the realistic chance of implementing within the associated tool sets.

To validate the proposed notation trial implementations should be made one or more of the stated problem areas. If and when a general consensus is obtained relative to the form and desirability of encoding control modules the relevant standardisation bodies should be approached with a view to either extending ASN.1 or defining the new TSN.1 notation.



## 5. CONCLUSION

Without the proposed extensions to ASN.1 (either directly or via TSN.1) it is not possible to formally specify the message structures of any standardized protocol. In addition, the demands of new technologies will cause a proliferation of non-standardized, informal encoding rules which will be specific to a very narrow field of application

The proposed Encoding Control mechanism would provide a possible solution for the current problems and omissions highlighted in this paper. In addition it is intended to provide a generic extendable mechanism which will lend itself to the solution of future encoding problems and lastly an encoding control notation would complete the set of formal description techniques.

## References

- [1] ETSI GSM 04.07 V7.0.0 (1999-4) Digital cellular telecommunications system (Phase 2+); Mobile radio interface signalling layer 3; General aspects: Annex B Description of CSN.1
- [2] ISO/IEC 9646-3:1994, Information technology - Open System Interconnection Conformance testing methodology and framework Part3: The Tree and Tabular Combined Notation (TTCN)
- [3] ITU-T Recommendation Z.110 (1993), Formal Description Techniques - Criteria For The Use And Applicability Of Formal Description Techniques
- [4] ITU-T Recommendation X.680 (1994) | ISO/IEC 8824-1:1995, Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation.
- [5] ITU-T Recommendation X.681 (1994) | ISO/IEC 8824-2:1995, Information technology - Abstract Syntax Notation One (ASN.1): Information object specification.
- [6] ITU-T Recommendation X.682 (1994) | ISO/IEC 8824-3:1995, Information technology - Abstract Syntax Notation One (ASN.1): Constraint specification.
- [7] ITU-T Recommendation X.683 (1994) | ISO/IEC 8824-4:1995, Information technology - Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications.
- [8] ITU-T Recommendation Z.100 (1993), Programming Languages - CCITT Specification And Description Language (SDL)
- [9] ITU-T Recommendation X.690 (1994) | ISO/IEC 8825-1:1995, Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).

- [10] ITU-T Recommendation X.691 (1995), Information technology - ASN.1 encoding rules: Specification of Packed Encoding Rules (PER).
- [11] ITU-T Recommendation Q.931 (1993), Digital Subscriber Signalling System No. 1 (DSS 1) - ISDN user-network interface layer 3 specification for basic call control
- [12] ITU-T Recommendation Q.704 (1993), Signalling System No. 7 - Signalling Network Functions And Messages
- [13] ITU-T Recommendation Q.773 (1993), Signalling System No. 7 - SCCP formats and codes.
- [14] ITU-T Recommendation Q.763 (1993), Formats and codes of the ISDN User Part of Signalling System No. 7.
- [15] ITU-T Recommendation Q.1218 (1995), Interface recommendations for intelligent network CS1.
- [16] ITU-T/ISO ASN.1 Experts Meeting Lannion Jan 99 Document 27: ASN.1 Encoding Rules - Octet Encoding Rules
- [17] ITU-T/ISO ASN.1 Experts Meeting Lannion Jan 99 Document 28: XER (XML Encoding Rules)