Generation of JAVA classes from ASN.1 for BER transfer syntax

Viatcheslav V. Ostapenko

In order for communication to occur it is necessary to specify the type and format of the information to be exchanged. A general, distributed application protocols use very complex types of information. It is therefore useful to have a formal tool that permits the precise definition of the types of information exchanged. It is also useful to have a set of rules that specifies the format of the values for each of the types. Each instance or value of a type must be converted to this format before transmission. Abstract Syntax One (ASN.1) with its associated encoding rules is an example of a formal tool for specifying the type and transfer format of type values. ISO and ITU-T define it with its encoding rules as an external data representation language for communication of heterogeneous systems (ITU-T X.208, X.680). This tool has been used extensively with existing OSI specification protocols as well as many new applications.

External data representation languages support two components: facilities to describe data in terms of a set of application specific data type and facilities to define how values of these types are represented serially for a communication. The fires is termed abstract syntax and the second transfer syntax. In earlier representation languages these two concepts were combined. Current languages such as ASN.1 make a clear separation between them. This distinction permits multiple transfer syntaxes to exist. There was earlier the only one standard set of encoding rules for ASN.1 These are called Basic Encoding Rules (BER - ITU-T X.209, X.690). Now there are newer encoding rules, but this talk does not deal with them.

To use data structures described by ASN.1 they must be programmed in a common programming language. It would certainly be a great advantage to an application programmer if an automatic process existed which would convert the types in an abstract syntax into types used by a specific programming language. It would also be very useful if a set of encoding routines was generated that could be called by the programmer to convert a value of a type to and from particular transfer syntax. The biggest part of such translators is generate C source code with structures and encoding procedures for them. C language is considered portable language so source code can be used in different systems and platforms. But as practice shows there is a lot of incompatibilities between different compilers. Also a big number of different structure and procedures to process them gives a big possibility of making error by programmer and C type checking mechanism does not provide strong enough type checking.

In this talk we describes method of translation of ASN.1 data representation and BER rules into JAVA classes and methods to process data of that classes.

JAVA is a programming language environment suggested Sun Microsystems as common language for any platform. It is simple, object and network oriented, robust, secure, architecture neutral, high-performance, multithreaded, dynamic language. Java is designed to meet the challenges of application development in the context of heterogeneous network-wide distributed environments. Paramount among these challenges is secure delivery of applications that consume the minimum of system resources, can run on any hardware and software platform, and can be extended dynamically.

Object oriented structure of JAVA allows generation encoding methods for all classes with the same names inside classes, so each class will KNOW how to encode and decode data in it. Every constructed class including several other, may be also constructed, classes can easily encode them by calling encoding methods for each class. The only exclusion is a set of simple built in types. JAVA stream classes allows designing encoding methods to know how to store data to a stream and appropriate stream class will work with a hardware device or file. It simplifies programming process greatly.

Generation of JAVA classes from ASN.1 data description can be effectively used for programming Internet oriented applications. It is also useful for generation tests from TTCN (Tree and Tabular Combined Notation) that uses ASN.1 for specification of internal data types. Tests in JAVA will run in every system and there is no need to regenerate them for different platforms.

This talk describes method of generation only BER encoding rules in JAVA, but it can be easily modified to generate any other transfer syntax.