ASN.1: Defining a Grammar for the UMLS Knowledge Sources

Alexa T. McCray, National Library of Medicine, Bethesda, Maryland
Guy Divita, Management Systems Designers, Inc., Vienna, Virginia

ABSTRACT

The Unified Medical Language System (UMLS) project provides resources on an experimental basis to the research community. In 1995 the four UMLS Knowledge Sources have been provided in an additional data format, Abstract Syntax Notation One (ASN.1). The benefits of ASN.1 are that it provides a standard, formal grammar for complex data and allows exchange of that data in a way which is independent of the particular software and hardware environment in which the data are created and stored. The paper begins with an introduction to the ASN.1 standard itself. It continues with a discussion of the ASN.1 implementation of the UMLS Knowledge Sources and some of the consequences for the newly released UMLS Knowledge Source Server. It concludes with a discussion of some of the benefits of using ASN.1 encoded data.

INTRODUCTION

Each year since the fall of 1990, the Unified Medical Language System® (UMLS®) project has provided resources on an experimental basis to the biomedical and medical informatics research community [1]. Because biomedical information is stored in a large number of databases with varying access vocabularies and other access conditions, and because it is difficult for most health professionals to be proficient in all of these differing access conditions, the UMLS goal is to provide integrated and flexible access to biomedical information sources through a unified conceptual structure.

In 1994 the four Knowledge Sources were distributed to over 500 researchers or institutions. The Metathesaurus, the largest of the Knowledge Sources, was released as a set of ASCII relational tables. The Semantic Network and the SPECIALIST Lexicon were released in both ASCII unit record and relational table form. The Information Sources Map was released as a set of relational tables. In 1995 all four Knowledge Sources have been released in an additional format, known as ASN.1 (Abstract Syntax Notation One).

ASN.1 not only provides a consistent description of the data in the Knowledge Sources, but it also provides a grammar of the data so that it is possible to validate the data, uncovering any inconsistencies in the process. Further, ASN.1 is used within the Z39.50 protocol, an emerging standard for information retrieval applications.

The ASN.1 Standard

ASN.1 is a formal language for describing structured, and potentially quite complex, data that allows unambiguous data exchange across some communication medium, such as the Internet [2]. It allows a machine, programming language, and database independent way of characterizing the data and provides a standard for sending (through encoding rules) and receiving (through decoding rules) the data. ASN.1 was originally designed to work with the Internet X.400 electronic mail message handling system [3], but it is increasingly used in other applications, for example, the Z39.50 information retrieval protocol [4].

ASN.1 is the Open Systems Interconnection (OSI) standard for the upper layers of the OSI network model [5]. That is, the data are sent through the Presentation layer (as a formally encoded stream of bits sent over the network), and they are interpreted by the Application layer (whatever the application or computer system may be).

The ASN.1 standard comprises two ISO international standards, one that specifies the basic language [6] and another one that specifies the basic encoding rules for transmitting data in ASN.1 format [7]. The Basic Encoding Rules were designed to minimize data size at the possible expense of encoding and decoding time and CPU cycles. There is an effort to develop certain extensions to the Basic Encoding Rules [2:121-124] and these may address efficiency issues as well as issues of data security.

An ASN.1 specification is built by using a set of

0195-4210/95/$5.00

predefined types such as integers, sets, sequences, strings, and the boolean values true and false. New types are defined using these building blocks, allowing the description of an arbitrarily complex set of data. The notion of "type" is a fundamental concept in ASN.1 and is closely related to the notion of type in programming languages, but there are subtle differences. In ASN.1, types are abstract since they need not be implemented directly by machines.

The basic strategy adopted in ASN.1 to define types is the following. Simple types are defined that identify sets of values that are easily characterized, by enumeration or by some other descriptive means, independently of other types. Structured types are defined that construct a more complex type from other known types, using a handful of construction methods. Modules provide packages of related type definitions, and tags are defined that add information to a type to facilitate encoding and decoding of values using a particular syntax. The entire specification is put into a module, or a collection of related definitions of types, values and other modules. Definitions are composed of types imported from other modules, locally defined types, and a specification of types to be exported which can be imported by other modules.

THE UMLS KNOWLEDGE SOURCES EXPRESSED IN ASN.1

The ASN.1 file format is intended as a means of transferring heterogeneous information from application to application. There are several tools available for encoding and using data in ASN.1 format. The UMLS project uses tools that were developed and maintained by NLM's National Center for Biotechnology Information (NCBI) [10]. The NCBI toolkit has been distributed with the UMLS to aid developers who are using the ASN.1 encoded files. The two primary tools are "asntool", which parses and validates data expressed in ASN.1 format, and "asncode", which creates C++ objects and object methods from the ASN.1 specification. The toolkit also includes a library of C and C++ routines for reading and writing ASN.1 encoded data and for sequentially searching for individual components of the encoded data.

The design and implementation philosophy of the ASN.1 data model for the UMLS Knowledge Sources involved adhering to the existing UMLS data model. Thus, the ASN.1 translator does not alter the content of the data in any way. Only the form in which the data are expressed varies. The following shows a portion of the UMLS ASN.1 specification for the Metathesaurus. The specification has been modified in some details for readability.

```
UMLS-meta-release    ::= SEQUENCE

{
  concepts          SET OF UMLS-concept,
  cooccurrences     SET OF Cooccurrence,
  indexes           SET OF UMLS-index
}

UMLS-concept    ::= SEQUENCE

{
  name                      VisibleString,
  cui                       VisibleString,
  definitions               SET OF UMLS-definition,
  semantic-types            SET OF UMLS-semantic-type,
  umls-terms                SET OF UMLS-term,
  related-concepts          SET OF Related-concept,
  locators                  SET OF Locator,
  attributes                Concept-Level-Atts,
  associated-expressions    SET OF Associated-expression,
  occurrences               SET OF Occurrence
}

Related-concept    ::= SEQUENCE

{
  related-concept-name      VisibleString,
  related-concept-cui       VisibleString,
  relation                  Relation,
  relationship-attribute    VisibleString,
  sab                       Source-abbreviation
}

Relation    ::= ENUMERATED

{
  par    (1),
  chd    (2),
  sib    (3),
  aq     (4),
  qb     (5),
  rr     (6),
  ur     (7)
}
```

Several ASN.1 types are illustrated in the example. The simple type "VisibleString" refers to any character in the international version of ASCII. The type "ENUMERATED" is another simple type that is used when there is a finite set of values for a particular

attribute. The structured types "SET OF" and "SEQUENCE OF" are complex and are defined in terms of their component parts, so that their values are derived from the simple types. The values of the "SET OF" type are unordered collections of the values of the component types. In contrast, the values of the "SEQUENCE" type are ordered, with the order being determined by the order in the definition.

The portion of the specification given above begins with a description of the Metathesaurus data elements. As can be seen, these consist of concept information, co-occurring terms information, and several indexes. A Metathesaurus concept consists of a name, a concept unique identifier (cui), a single or perhaps multiple definitions, a single, or perhaps multiple semantic types, a set of synonymous terms, and a set of related concepts, potentially from a variety of thesauri. Additionally, there may be locator information, which gives an indication of the information sources in which a concept appears. There might be several types of attributes, such as those that are associated with a concept in a particular source vocabulary, and there may be associated expressions that give an indication of how that concept would be searched in a particular database. Finally, there might be instances of co-occurring concepts in a database such as MEDLINE.

The specification continues with a description of a related concept. Note that the description for the concept allows a set of related concepts for any given UMLS concept. A related concept is defined as a sequence of a concept name, a concept UI, the relation between the original concept and the related concept, the relationship attribute, such as "narrower than" or "broader than", and the source of the relationship.

Finally, the allowable relations are enumerated. A concept can be a parent (par) of another concept, a child (chd), or a sibling (sib). It can be an allowed qualifier (aq) in a Metathesaurus source vocabulary, or it can be restricted by a qualifier (qb). A concept can have a non-synonymous relationship with a concept that was reviewed and labeled by subject domain experts during construction of the Metathesaurus (rr), or it can be a relationship that is inherited from a source vocabulary but was not reviewed again during Metathesaurus construction (ur).

The following ASN.1 description of part of a Metathesaurus concept illustrates how concepts are instantiated in keeping with the ASN.1 specification.

```
UMLS-metathesaurus-module ::= {
 Concepts {
  {
   name "Achromia parasitica" ,
   cui "C0001083" ,
   definitions {
    {
     sab DOR27 ,
     def "a variant of tinea versicolor
        occurring in dark-skinned infants,
        particularly in the tropics ..." } } ,
   semantic-types {
    {
     tui "T047" ,
     name Disease or Syndrome } } ,
   umls-terms {
    {
     name "Achromia parasitica" ,
     lui "L0001083" ,
     language ENG , ...
   related-concepts {
    {
     related-concept-cui "C0040262" ,
     relation RR ,
     relationship-attribute "O" ,
     sab MTH } } } } } ...
```

The name of the concept is "Achromia parasitica"; its unique identifier (cui) is listed; and its definition is from the Dorland's Illustrated Medical Dictionary, 27th edition (DOR27). The semantic type of the concept is "Disease or Syndrome", whose unique identifier (tui) in the UMLS Semantic Network is "T047". The unique identifier (lui) of the term itself is listed, and it is noted that the language is English. The unique identifier of a related concept is listed (C0040262). A look up in the Metathesaurus reveals that this is the concept "Tinea Versicolor". The concepts are related, but since the relationship between the concepts is neither "broader than" nor "narrower than", it is marked as "other" ("O"), and the source abbreviation (sab) is for the Metathesaurus.

Note that the UMLS ASN.1 description collects all information about the concept in one place, much like a unit record format, and in contrast with the relational table format, where information about a concept may be found in several tables.

Together with the release of the UMLS data in ASN.1 format, a set of cross reference tables is provided. These cross reference tables list the correspondences between ASN.1 data elements and the relational table in which those data elements are found. A portion of

the cross reference table for the Metathesaurus data elements is shown below.

ASN Data Element	MR Table
UMLS-concept.name	MRCON
UMLS-concept.cui	MRCON
UMLS-concept.definitions	MRDEF
UMLS-concept.semantic-types	MRSTY
UMLS-concept.umls-terms	MRCON
UMLS-concept.related-concepts	MRREL
UMLS-concept.locators	MRLO
UMLS-concept.attributes	MRSAT
UMLS-concept.associated-expressions	MRATX
Cooccurrences	MRCOC

THE UMLS KNOWLEDGE SOURCE SERVER, ASN.1, and Z39.50

The UMLS Knowledge Source Server is a tool for providing Internet access to the information stored in the UMLS Knowledge Sources [8]. It has been in use by several test sites since 1994 and was announced for use by the UMLS community in the spring of 1995. The purpose of the Knowledge Source Server is to make the UMLS data more accessible to users, and in particular to systems developers. The centrally managed server provides developers with UMLS information remotely and on demand, and, thus, developers do not need to invest their time and effort in understanding the structure of the data files and other details in order to use the UMLS data in their applications.

The system architecture is based on the client-server paradigm wherein remote site users send their requests to the centrally managed server at the NLM. The client programs can run on platforms supporting the TCP/IP communication protocol. UMLS information can be retrieved through the client program by typing in the names of specific queries at the command-line. For example, the user may request information about particular Metathesaurus concepts, including attributes such as the concept's definition, its semantic types, the concepts that are related to it, etc. The user can also request information about the attributes themselves, for example, by asking for all the concepts that have been assigned to a particular semantic type.

The command-line interface provides an option of specifying a file name containing terms for which UMLS information is desired. The command-line interface has been implemented using the Knowledge Source Server application programming interface (API). The API consists of simple functions for establishing and breaking a connection to the server, for sending queries to the server, and for receiving results from the server. Each function returns an indication of success or failure of the requested operation.

The Knowledge Source Server is being redesigned to take advantage of the UMLS data in the ASN.1 format for the Z39.50 information retrieval protocol [9]. The Z39.50 protocol allows clients to send queries to information databases, servers to send summary reports of retrieved elements, and clients to control the incremental transfer of the result set. It also provides methods for querying the resources available in the server.

Within the Z39.50 protocol, the protocol data units, which are the units of communication between the client and the server, are defined in ASN.1. The redesign of the UMLS Knowledge Source Server in keeping with the Z39.50 protocol will take advantage of the UMLS data in ASN.1 format. The intention is to redesign the system such that users will be able to query the data in each of the Knowledge Sources with greater flexibility than is currently possible. Although many queries have been implemented to date, the user (client) will no longer be restricted to the ones that have been pre-defined, but rather will be able to query virtually any data element in any of the Knowledge Sources. Additionally, the client can construct complex, structured queries which will be encoded in ASN.1 by the server and decoded by the client.

The Z39.50 protocol provides for a powerful "Explain" facility which gives meta-level information about the server. This allows the client to ask which databases are available for searching, what data elements are available in a particular database and what the structure of those data elements is, and what sorts of attributes are supported.

THE BENEFITS OF USING ASN.1

Perhaps the most important benefit of using ASN.1 is that it is a standard. It is used internationally and is used in the specification of many communication protocols. Rose [5:137] claims that, "ASN.1 is destined to become the network programming language of the '90's, just as the C programming language is largely seen as having been the systems programming language of the '80's." Because ASN.1 allows data

description at a fairly abstract level, the developer is freed from the details of exactly how the data will be transmitted and can concentrate on higher level design issues. Further, ASN.1 allows the developer to build quite complex descriptions from a relatively small set of basic building blocks. The resulting ASN.1 specification of this complex data can be freely exchanged among a variety of users of that data in a way which is database management system, programming language, and platform independent.

A second important benefit of using ASN.1 relates to the nature of the description itself. An ASN.1 specification is essentially a BNF (Backus-Naur Form) grammar of the data being described. This forces certain "well-formedness" constraints on the data. In other words, the data must be expressed in a semantically coherent way in the database. Expressing data in ASN.1 form allows validation of the form and content of the data. If data attributes or values appear in the database that are not accounted for in the ASN.1 description, then either the description is wrong and needs to be modified, or the data have been erroneously represented in the database. In either case, the grammar has served to validate the data elements. This becomes invaluable when dealing with complex sources of data, such as the UMLS Knowledge Sources.

CONCLUSION

The translation of the UMLS Knowledge Sources into ASN.1 format has resulted in a structured and formal grammar for those Knowledge Sources. This makes it possible to consider and discuss the semantics of each of the Knowledge Sources and to look for any inconsistencies in the data structures. In addition, as actual data are instantiated, the ASN.1 grammar serves to validate the data elements and their structure.

The NCBI tools that are provided with the UMLS release should assist developers in exploring the ASN.1 data and will allow them to express local data in the same format if they choose to do so.

The reimplementation of the UMLS Knowledge Source Server consonant with the Z39.50 protocol and using the ASN.1 versions of the UMLS data files will provide more flexibility for the user in making any possible combination of queries, and it will allow users to define a view of the data suitable for their applications. The protocol will also provide a greater degree of control on the amount of data retrieved and presented to the user.

REFERENCES

1. Lindberg DAB, Humphreys BL, McCray AT. The Unified Medical Language System. Methods of Information in Medicine 1993:281-91.

2. Steedman D. Abstract Syntax Notation One (ASN.1): The Tutorial and Reference. Twickenham: Technology Appraisals Ltd., 1993, 171 pages.

3. Lynch DC, Rose MT. Internet System Handbook. Reading: Addison-Wesley Publishing Company, Inc., 1993, 790 pages.

4. Michael JJ, Hinnebusch M. From A to Z39.50: A Networking Primer. Westport: Mecklermedia, 1995, 166 pages.

5. Rose MT. The Open Book: A Practical Perspective on OSI. Englewood Cliffs: Prentice Hall, 1990; 651 pages.

6. Information Processing - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1). International Organization for Standardization and International Electrotechnical Committee, 1987. International Standard 8824.

7. Information Processing - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1). International Organization for Standardization and International Electrotechnical Committee, 1987. International Standard 8825.

8. McCray AT, Razi A. The UMLS Knowledge Source Server. To appear in Proceedings of Medinfo '95.

9. ANSI Z39.50; Information Retrieval Service and Protocol. American National Standard Information Retrieval Application Service Definition and Protocol Specification for Open Systems Interconnection. 1992.

10. Ostell JM. The NCBI Software Tools. Manuscript, National Library of Medicine, 1994.