TRANSPORT PROBLEMS 2009
PROBLEMY TRANSPORTU Volume 4 Issue 2

Zbigniew ŁUKASIK*
Technical University of Radom, Faculty of Transport
26–600 Radom, Malczewskiego 29, Poland
Waldemar NOWAKOWSKI
Z.A. KOMBUD S.A.
26–600 Radom, Wrocławska 7, Poland
**Corresponding author*. E-mail: z.lukasik@pr.radom.pl

ASN.1 NOTATION FOR EXCHANGE OF DATA IN COMPUTER-BASED RAILWAY CONTROL SYSTEMS

Summary. Development of railway control systems aims at computerization. In most cases these systems are Distributed Real Time Systems. However, a huge problem in their putting into practice is the lack of interface standardization in the range of data structures and information exchange methods. It results in a variety of solutions, and thus in problems concerning cooperation of systems that come from different software vendors. Specification of protocols for data exchanging applications should therefore be created with the use of generally accepted standards. One of them is ASN.1 (Abstract Syntax Notation One) language, which shall be presented in this article.

ZASTOSOWANIE NOTACJI ASN.1 W WYMIANIE DANYCH DLA KOMPUTEROWYCH SYSTEMÓW STEROWANIA RUCHEM KOLEJOWYM

Streszczenie. Rozwój systemów sterowania ruchem kolejowym zmierza w kierunku ich komputeryzacji. W większości przypadków systemy te można traktować jako rozproszone systemy czasu rzeczywistego (ang. *Distributed Real Time Systems*). Jednak dużym problemem w ich wdrażaniu jest brak standaryzacji interfejsów w zakresie struktur danych i metod wymieniany informacji. Prowadzi to do różnorodności stosowanych rozwiązań, a tym samym problemów ze współpracą systemów pochodzących od różnych producentów. Specyfikację protokołów dla komunikujących się aplikacji należy, więc tworzyć z zastosowaniem ogólnie przyjętych standardów. Jednym z nich jest, zaprezentowany w tym artykule język ASN.1 (ang. *Abstract Syntax Notation One*), czyli abstrakcyjna notacja składniowa.

1. ABSTRACT SYNTAX NOTATION ONE

The syntax of ASN.1 is very similar to the ones used in most of the programming languages. However it lacks keywords related to execution flow, such as loops or conditional statements. The language contains a few built-in data types and a set of rules, which enable individual creation of new, more complex data types, as well as assignment of constant values to some elements contained in data types defined this way. Abstract types related to each other are grouped together in modules to which

one may refer externally via their names. Simplifying, the module may be introduced as the following structure [1, 2]:

ModuleIdentifier
DEFINITIONS ::=
BEGIN
module_body
END

where: **DEFINITIONS**, **BEGIN**, **END** are keywords for ASN.1,
ModuleIdentifier is the value that identifies the module,
module_body –the body of the module containing links, symbols and assignments.

Data types in ASN.1standard may be divided into basic and constructed ones.

Basic types are types that can hold value from predefined set of values. The group of basic types includes:

- boolean type (BOOLEAN),
- null type (NULL),
- integer type (INTEGER),
- enumerated type (ENUMERATED),
- real type (REAL),
- bit string type (BIT STRING),
- octet string type (OCTET STRING),
- object identifier type (OBJECT IDENTIFIER)

and a whole group of character string types (CHARACTER STRING).

Constructed types are formed by combining basic types into structures. The group of constructed types includes:

- structural types:
- sequence types (SEQUENCE and SEQUENCE OF),
- set types (SET and SET OF),
- choice types (CHOICE and ANY).

By means of the above mentioned types one may define any data structures for information exchange protocols.

2. ASN.1 OBJECT ENCODING RULES

Every ASN.1 as a formal notation of abstract syntax enables defining data structures and assigning them values, yet it does not include the method for representation of the transmitted data. Thus, in order to ensure unambiguous interpretation of the transmitted information, it is essential to use the same data structure encoding mechanism. Currently there are many ASN.1 object encoding rules [1,2]:

- Basic Encoding Rules (BER),
- Distinguished Encoding Rules (DER),
- Canonical Encoding Rules (CER),
- Packed Encoding Rules (PER),
- XML Encoding Rules (XER).

3. DESIGNING COMMUNICATION SOFTWARE

In the case of communication software a crucial thing is data exchange between systems. Thus, in designing communication software formal specification methods should be used. One of such methods is ASN.1 due to its numerous advantages, such as:

- Abstractness: the notation uses abstract syntax which enables description of data structures irrespective of their further representation,
- Structurality: the notation enables decomposition of data structures into smaller units called modules. This enables preservation of greater clarity, which in turn facilitates verification and modification of data structures,
- Formal definitions: the syntax and semantics of the notation have a complete and formal definition. The formal model allows using analytical theory in order to prove correctness.

Using methods of specification that do not possess the above mentioned qualities while creating computer railway control systems results in high costs of designing, testing and maintenance of the software.

If ASN.1 is used to define data structures of information exchange protocols, then there is no necessity to design application modules responsible for data encoding and decoding. ASN.1 source files undergo the compilation process, as a result of which ready-made application modules responsible for data encoding and decoding are formed. These modules may be written in one of the programming languages which is chosen during the compilation (Fig. 1).

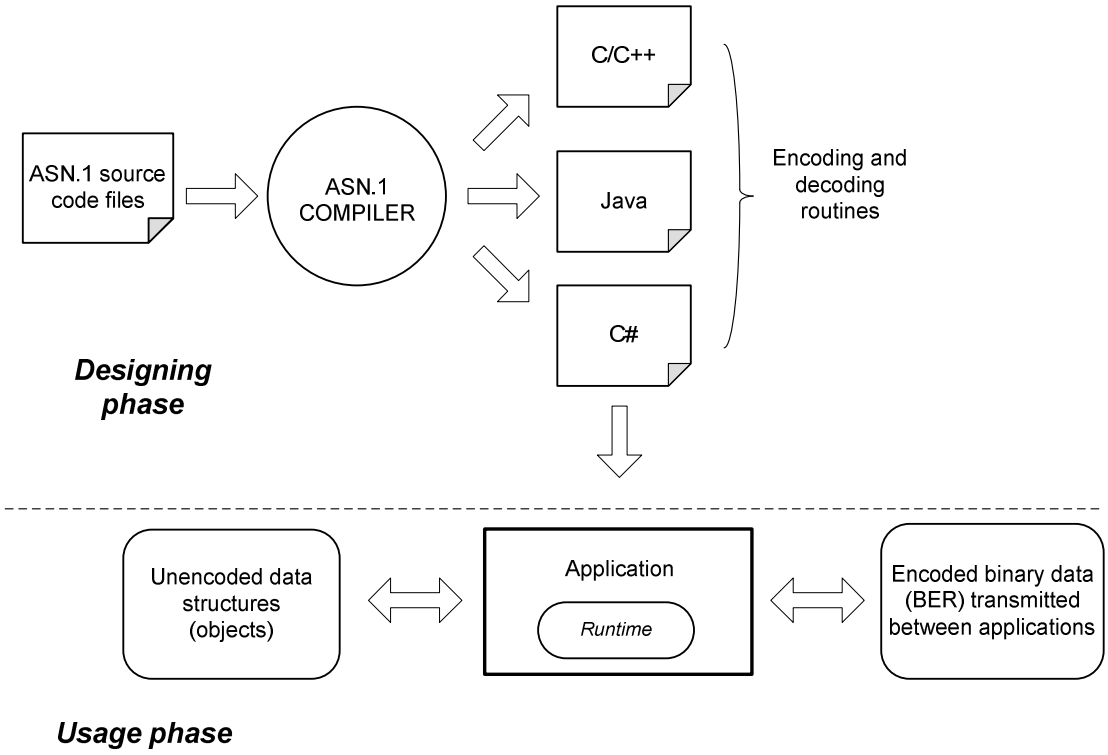

Fig. 1. Designing and usage of communication software in ASN.1
Rys. 1. Projektowanie i wykorzystanie oprogramowania do komunikacji w ASN.1

Using ASN.1 as formal specification language results in obtaining communication software that possesses the below mentioned qualities.

- Compatibility: compatibility with other ASN.1standard-based systems,
- Modifiability/Extendibility: easiness to make changes in the application due to the use of transformation method,
- Correctness: ability to operate in accordance with the assumptions and expectations,
- Testability/Verifiability: easiness to verify the correctness of the program, as well as source codes readability,
- Efficiency/Capacity: high efficiency and low CPU and memory consumption, as well as high performance of the data encoding process.

4. TEST RESULTS

In order to verify the suggested method of designing communication software for computer railway control systems with the use of ASN.1 applications have been written. They enable simulation of data exchange between interlocking system and control system (Fig. 2).

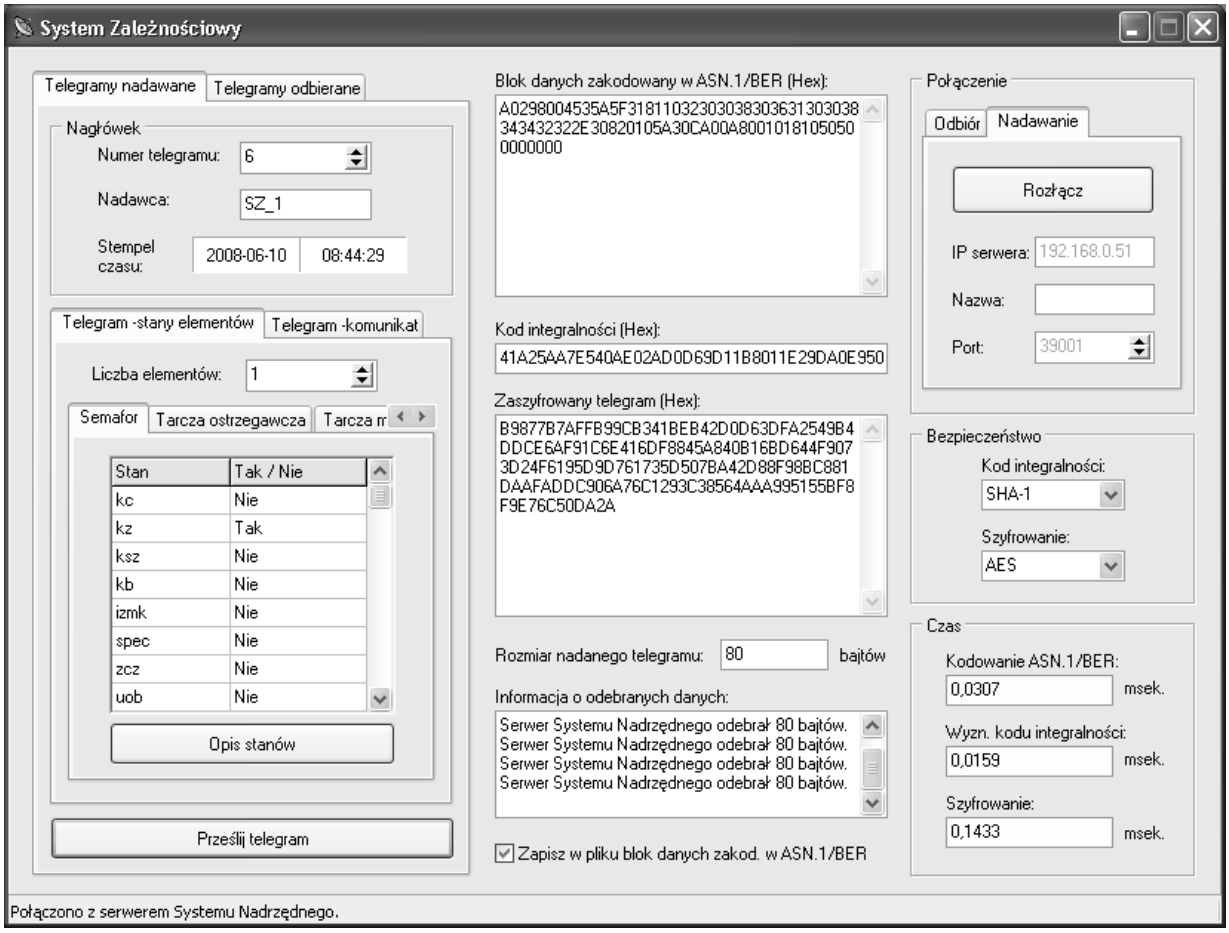

Fig. 2. Main window of communication software
Rys. 2. Główne okno programu do komunikacji

Moreover, apart from the ability to encode data in ASN.1/BER, the applications have been fitted with security functions responsible for integrity and confidentiality of the data transmitted through computer networks. Efficiency of ASN.1/BER may be measured by the number of railway control devices which can be supported by the system within the given time limit. This results from the fact that as the size of the controlled object rises, the information exchange proceeds with bigger time limitations. In the article are presented test results for two cases of information exchange.

- Closed transmission system (Case 1). Data were encoded and decoded with the use of ASN.1/BER. CRC64 checksum was responsible for data integrity,
- Open transmission system (Case 2). Data were encoded and decoded with the use of ASN.1/BER. SHA-1 hash function was responsible for data integrity and the encrypting algorithm was AES.

In both cases data blocks were created in the form of lists of elements within the range of 1-16 devices for each block. The cycle of data exchange was fixed within the range of 200-1000 ms and the available bandwidth was 512 kb/s.

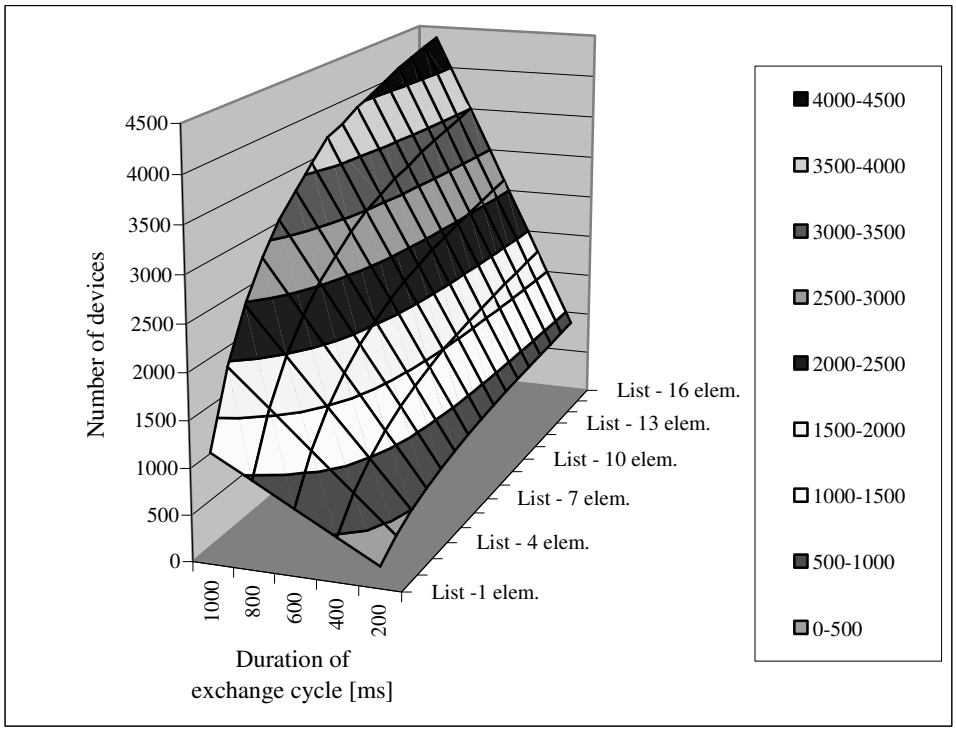

Fig. 3. Case 1 (ASN.1/BER – CRC64)
Rys. 3. Przypadek 1 (ASN.1/BER – CRC64)

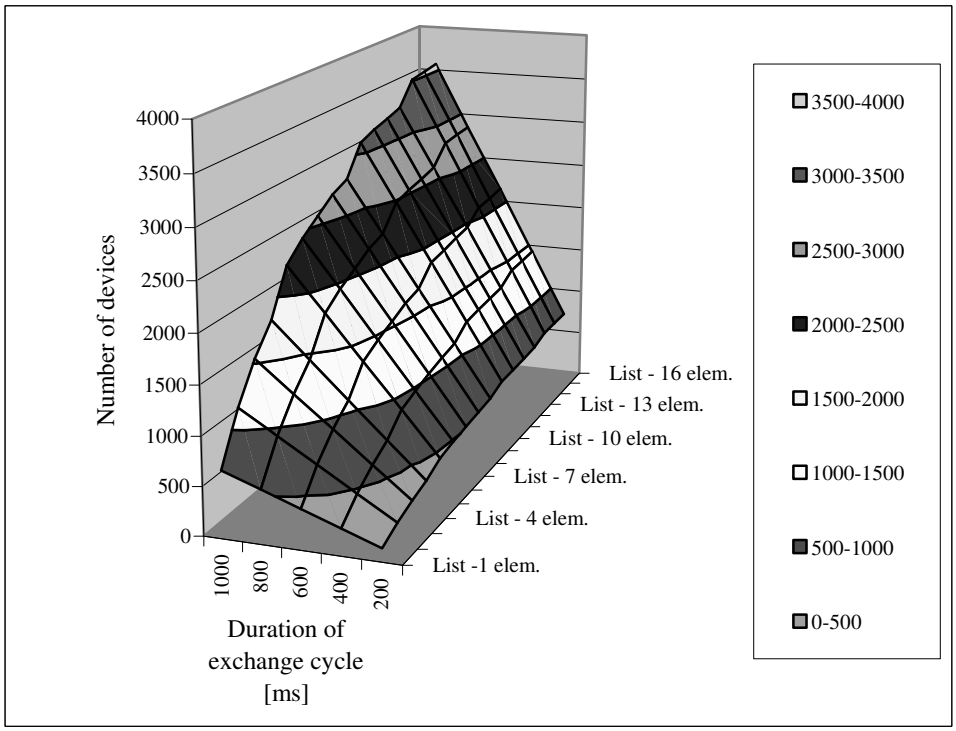

Fig. 4. Case 2 (ASN.1/BER – SHA-1 – AES)
Rys. 4. Przypadek 2 (ASN.1/BER – SHA-1 – AES)

5. CONCLUSIONS

Test software presented in the article enables verification in practice of ASN.1 usage in information exchange protocols of computer railway control systems. Test results confirm great usefulness of this notation. Due to the binary encoding (BER) the messages are of small sizes, i.e. of several dozen bytes. Time delays resulting from encoding and decoding processes are also a few times shorter than the delays that result from data encryption process. Moreover, the usage of ASN.1 significantly simplifies designing of communication and, at the same time, enables interface standardization of computer railway control systems.

References

1. Dubuisson, O.: *ASN.1 Communication between Heterogeneous Systems.* OSS, 2000.
2. Larmouth J.: *ASN.1 Complete*, OSS, 1999.
3. Łukasik Z., Nowakowski W.: „*Designing communication software for computer railway control systems with the use of ASN.1*", Monograph, Kazimierz Pułaski Technical University of Radom, Faculty of Transport, Radom, 2008, pp. 391-398.
4. Łukasik Z., Nowakowski W.: „*Wymiana informacji w systemach związanych z bezpieczeństwem*", XII Międzynarodowa Konferencja „TransComp", Zakopane, 2008.
5. Sacha K.: *Projektowanie oprogramowania systemów sterujących.* Warszawa, Oficyna Wydawnicza Politechniki Warszawskiej, 2006.

Received 24.11.2008; accepted in revised form 25.06.2009