A General Analysis Method for Data Link Log Files

Zhengqin Xu*, Juan Chen*, Laixian Peng*, Wendong Zhao*, Xiaojing Chu

Institute of Communications Engineering, Army Engineering University of PLA, Nanjing, China, 210007,
*corresponding author: 155267201@qq.com, pumpkin0910@163.com, lxpeng@hotmail.com, nj_mouse@163.com

Abstract: Aiming at the problem that the data analysis software of data link experiment has strong dependence on the log data format and the analysis function is difficult to expand dynamically, this paper puts forward the design idea of the log file general parsing software and the general description language of the log file of the bit-oriented data link system, and designs and implements the log file data analysis system of the general data link system. The file format description language is used to separate the analysis software from the specific data format. The log file data of various formats can be imported into Excel software on demand. By using the powerful data analysis function of Excel software, the business departments can achieve targeted data analysis.

1. Introduction

Data analysis is the main means to evaluate the performance and operational effectiveness of data link system and to locate problems. Usually, data recording function is set up in the system, data log format is stipulated, special data analysis software is developed according to the need of data analysis and specific log format, and the required function, performance and operational effectiveness index are obtained[1]. This kind of software development method, which is bound with log data format and index system, is not flexible enough to adapt to the changes of log format, interface protocol format and data link message format. At the same time, the analysis function of data analysis software can not be expanded, and it is difficult to adapt to the changes of data analysis requirements[2].

This paper designs and implements a general data parsing system for log files of data link system based on formal description of message format, which is suitable for parsing and processing of various format files. Each Business Department of the army can extract the data in log files according to its business characteristics and generate Excel reports according to its needs, making use of the powerful number of Excel. The performance analysis and effectiveness evaluation of data link system related services based on processing capability provide a general means for data link system data analysis independently by data link users of various operational departments in the army.

2. Based on Formal Description General Data Analysis System

2.1. Data Link Message Format and Log File Format
The data record file contains the header of the log file and the record data area, as shown in Table 1. For data link log files, the recording data area usually receives and receives data through the application layer connectionless reliable transmission protocol[3].
2.2. Variable Type
The data types of the file format description language include bits, bytes, double-byte words and four-byte double-word, as shown in Table 2.

| type   | Explain                                      |
|--------|----------------------------------------------|
| BIT    | Bit                                          |
| BYTE   | Single byte integer                          |
| WORD   | Double byte integers, high bits first        |
| Word   | Double-byte integers, low-bit first          |
| DWORD  | Four-byte integers, high bits first          |
| DWORD  | Four-byte integers, low first                |
| IPV4   | IP address, four bytes, high bit first       |

The definition of variables is similar to the definition of variables in C language, including variable type and variable name, such as "word year", "IPV4 destination IP;", each statement must end with ";". Array description similar to C language is used to represent multi-bit and multi-byte.

Description language supports nesting of structures. The description of recorded data is shown in Figure 1, where both "recorded data" and "SRProtocol" are structural names.

2.3. Conditional Statement
Because of the different interface types and different protocols, the data recorded in the data area are different, so in data analysis, we may only focus on one part of the data and filter out other data.
Therefore, it is necessary to introduce conditional sentences and branch sentences similar to language to distinguish data types under different conditions. The above unconditional "SRProtocol" field can be rewritten as follows:

```c
If (Interface Type== 3) // Processing Interface Data of Type 3
{
    SRProtocol SR protocol-data;
}
Else
{
    BYTE [length] independent data;
}
```

The conditional expression of log file format description language is similar to that of C language. The basic form is as follows:

- If (condition) type definition statement;
- If (condition) type definition statement 1 else type definition statement 2;

If and else are keywords; "condition" is a conditional expression similar to C language, because the data analyzed are bit-oriented messages, conditional expressions only support the addition, subtraction, multiplication and division of integers; "type definition statement", "type definition statement 1" and "type definition statement 2" are single type definition statements, or are types enclosed by "{" and "}". Define the sequence of statements.

2.4. Output Data Description

The description of output data includes the definition of output file and the definition of output data field. The output file definition includes the output file name and the corresponding field name, and the output file definition is represented by the keyword OutPutFileList. Figure 2 defines three output files and corresponding output field names, which will be used as column names of Excel files.

```
Figure 1 Basic description of recorded data
Figure 2 Output File Definition
```

For the definition of output data, it is stipulated that Output is used as a keyword to indicate that the data should be exported to Excel file, and the corresponding file name and column name are specified. For example, the definition of recording time in log files can be expressed in the form of Figure 2-4.

In Figure 3, the keyword Output is followed by three file names separated by commas, indicating that the field is exported to three files. The output column names are recorded time and the output form is the result of the TimeFormat transformation.
The use of dictionaries is similar to a function that specifies the corresponding decoding dictionary after the output column name. For example, the output of target attributes in track data can be defined as: BIT [3] Target attribute Output track data. xlsx: attribute Dict IDENTITY. Among them, dict is the key word and IDENTITY is the attribute encoding dictionary defined in Figure 3. The output results will be converted into enemy, self and friend attributes according to the target attribute values of 3 bits.

3. Design of Compiler System for Description Language

The functions of the system include describing file analysis and data analysis, and the functional composition of the system is shown in Figure 4. Before data analysis, users first need to select the corresponding description file according to different log file formats, compile the description file and generate the virtual machine instruction sequence for data analysis. In the data analysis stage, the user selects the corresponding log data files, and the system automatically completes the system analysis according to the virtual machine instructions. In the process of data analysis, for the specified output data of the description file, the corresponding internal functions or user-defined dictionaries are invoked to complete the data transformation, and the Excel report is output according to the specified file name and column name.

3.1. Lexical Analysis

Lexical analyzer is a deterministic finite automaton (DFA). Its functions include skipping single-line and multi-line annotations, identifying keywords, identifiers, operators, strings, numeric constants, etc. [6]

The experimental instruction in document[7] gives a finite automaton for lexical analysis of a small programming language. The description language defined in this paper is simpler. It contains only type definition statements and a small number of control statements. Arithmetic and logical expressions have clear contexts. Therefore, lexical analysis only needs to distinguish the identifiers and numbers ending...
with spaces, skipping single-line and multi-line annotations. The deterministic finite automata extracted by simplified Token is shown in Figure 5 and the state of DFA is given in Table 3.

![Figure 5 Lexical Analysis of DFA](image)

### Table 3 Lexical Analysis DFA Status Description

| state | Explain | Handle |
|-------|---------|--------|
| 0     | Start state | Numbers: state 1; letters or Chinese characters: state 2; "/": Turn State 3; Others: Return Single Character |
| 1     | Receive figures | Encounter non-numeric, end |
| 2     | Identifier | Encounter non-alphabetic/numeric, end |
| 3     | Have received / | If /, turn state 4, if *, turn state 5, and others, return a single character'/'
| 4     | Processing single-line comments | In case of return, turn to 0 |
| 5     | Processing multi-line comments | Meet *, turn state 6 |
| 6     | Encountered in the commentary* | If /, turn state 0, if *, continue state 6, otherwise, turn state 5 |

### 3.2. Grammatical Analysis

This paper proposes that message description language (PDL) is a relatively simple data type and structure definition language. Without considering arithmetic and conditional expressions, it is easy to write LL (1) grammar. Using LL (1) of description language, recursive descent analysis program can be constructed manually, top-down parsing can be realized by recursive invocation, and data analysis virtual machine program can be constructed in the form of intermediate code sequence. The grammar of the message description language is shown in Figure 6.
As can be seen from Figure 6, the formal definitions of arithmetic expressions and conditional expressions are not given in the message description language grammar. Because arithmetic expressions only appear in "\[\]" and conditional expressions only appear in "()'", which has a clear context. At the same time, arithmetic expressions and conditional expressions cannot be calculated at the stage of parsing. It is necessary to construct a complex sequence of intermediate codes to express them. Therefore, in the grammar analysis, we pass the arithmetic expression and conditional expression as a string parameter to the corresponding intermediate code instructions, which are interpreted and executed by the virtual machine interpreter in the data analysis phase.

4. Concluding remarks

This paper designs and implements the log file data analysis system of the general data link system. It separates the analysis software from the specific data format by using the file format description language. By modifying the description file, it can adapt to the data extraction of different data format files. According to the need, it can extract the relevant fields in the data file, and import the log file data of various formats into Excel table on demand. By using the powerful data analysis function of Excel software, each business department can achieve targeted data analysis.

The formal description method of log file and data link message format proposed in this paper can also be applied to the encoding and decoding system of data link message. Without modifying the software, it can adapt to the upgrade of message format and improve the generality and flexibility of the software system. At the same time, the formal description method of log file and data link message format can better unify the understanding of relevant standards among different departments and R&D personnel, provide powerful tools for system implementation and conformance testing of relevant standards, ensure the consistency of system implementation, and enhance the interoperability of the system.

Reference

[1] Introduction to Tactical Data Link, self-compiled textbook of Communication Engineering College, PLA University of Science and Technology, 2014.

[2] Network Centric Interoperability - Using a Variable Message Format (VMF) Based Data Link to Improve Situation Awareness and Close Air Support (CAS). Douglas Dusseau, Clinton Brock. IEEE A&E Magazine. 2004.9.

[3] MIL-STD-2045-47001 C CONNECTIONLESS DATA TRANSFER APPLICATION LAYER STANDARD DEPARTMENT OF DEFENSE INTERFACE STANDARD, 22 March 2002.
[4] VMF TIDP-TE Reissue 4, VARIABLE MESSAGE FORMAT (VMF) TECHNIC INTERFACE Design PLAN (TEST EDITION), DEPARTMENT OF DEFENSEINTERFACE STANDARD, 7 July 2000.

[5] MIL-STD-6016DTACTICAL DATA LINK (TDL) 16MESSAGE STANDARD, DEPARTMENT OF DEFENSEINTERFACE STANDARD, 12 December 2008.

[6] Chen Xiaomei, Tian Yang and Wang Baosheng. Design and implementation of router user-defined message performance test based on AX4000. Computer Engineering and Science, 2011, 33 (1), P25-30.

[7] He Xun, Wu Chunhan, Zhang Xingyuan, Compiling Method, Machinery Industry Press, 2010.