DATA FORMAT ANALYSIS USING AUTOMATON APPROACH

Maria Fátima Mattiello Francisco
Rubens Cruz Gatto
Sérgio Norio Itami
Wilson Yamaguti
National Institute for Space Research, Ground System Division, Brazil

Commission I, Working Group 6

KEY WORDS: Acquisition, Data, Software, Analysis, Recognition, Format, Pattern.

ABSTRACT

The availability of the environmental data acquired by the increasing number of both the Data Collecting Platform networks - DCPs, in the Brazilian territory, and the Receiving Ground Stations has been improved for their users. It was reached through the development of a dedicated module in the data processing and dissemination software system SCMCD, which performs both the data reception automatically and the data format analysis more efficiently in the Mission Center. Because the DCP data format can follow different message standards, for instance, the data format standard used by the DCPs in the ARGOS system and that used in the METEOSAT system, it is desirable to enable the Mission Center with intelligence in order to distinguish these different formats. This paper presents the design and implementation of that SCMCD acquisition module which provides the Mission Center with event-driven procedures. In order to perform the recognition and error analysis of several formats of DCP messages and also the Satellite Control Center messages (orbit prediction and the status of the Ground Station receivers), the acquisition module was developed based on a finite state automaton approach.

With the automaton approach, the implementation of the SCMCD acquisition module is prepared to incorporate as many different formats of DCP messages as the existing different types of Ground Stations working with environmental data transmission from DCPs. The aggregation of a new format to be processed is easy, since only the actions associated with each event must be coded. The state transition mechanism is already implemented by the acquisition module. It is only necessary to represent the state/event mapping from the automaton in form of a decision table.

1. INTRODUCTION

A DCP is a data collecting, formatting and transmitting device used in environmental monitoring applications such as stream flow and flood warning, irrigation control, water quality, water resource management, climate and air quality measurements, animal migration, etc. The data collecting function is performed by a set of sensors which usually monitor water level and quality of rivers and reservoirs, air temperature and humidity, atmosphere pressure, wind speed and direction, solar radiation, soil temperature and humidity, ozone concentration, precipitation, etc.

The acquired data, including DCP identification and status, are stored, formatted in a standard message and sent to the satellite . As other satellites, the Brazilian Complete Space Mission (MECB) with the Data Collecting Satellites - SCD series and the China-Brazil Earth Resources Satellite (CBERS) work in order to retransmit the DCP data back to the ground (INPE, 1992). For that communication, the DCP message format is not fixed by the satellite and depends only on the compatibility of the DCP message format and the data processor used in the Receiving Station.

When these satellites pass over the mutual visibility of the DCP and the Receiving Ground Station, a communication link is established between the DCP and the Receiving Ground Station, where the received messages are pre-processed, formatted according to the data processor used there, and further transmitted to the Mission Center. In addition, this Center receives from the Satellite Control Center both the orbit prediction data and the status of the Ground Station receivers. Because they are related with that satellite pass, they must be treated together.

In the Mission Center, the software system - SCMCD is responsible for the reception, treatment, storing and disseminating the environmental data (Francisco & Gatto, 1995). In order to support the flexibility of the environmental data being received from different standards of Ground Stations, with whatever standard format, and without predefined scheduling, an automatic acquisition module was incorporated in the software system SCMCD. The acquisition module was developed based on a finite automaton approach.

2. FINITE STATE AUTOMATON APPROACH

According to theory (Hopcroft & Ullman, 1979), "a finite automaton is a mathematical model of a system, with discrete inputs and outputs. The system can be in any one of a finite number of internal configurations (or states). The state of the system summarizes the information concerning past inputs that is needed to determine the behavior of the system on subsequent inputs".

After each satellite pass over a Ground Receiving Station, for every file received in the Mission Center, the SCMCD acquisition module identifies its origin (the logical channel) and, if it is an environmental data message (payload data), the system works on the extraction of all the individual messages from each DCP in the message block. This block is received in the Mission Center as a sequential file.

In order to simplify the article, it will be addressed here only the environmental data received from Cuiabá Ground Station. In that case, the DCP messages transmitted to the Mission Center follows the SDID (Station Data Interchange Document) format, presented in TABLE 1.

octet nr.			octet nr.
0	resp code / msg type	msg subtype	1
2	satellite code	satellite subcode	3
4	source site code	syst. type / number	5
6	receive site code	spare	7
8	channel ID		9
12	Time stamp in milliseconds		13
14			15
16	measured up-link carrier frequency		17
18	message length	channel status	19
20	sync. word		21
22	platform identification		23
24	platform sensor data (N x 4 Octets) (1 <= N <= 8)		

TABLE 1: DCP Message Format

Considering that the average of DCP messages received per satellite pass over Cuiabá Ground Station is 100 (they will be increased four times in the near future) and their transmission to the Mission Center is not totally safe of errors, the DCP message analysis shall be careful with the occurrence of both truncated and wrong messages. Therefore it is not sufficient to separate the DCP messages only based on both the recognition of the pattern which begins each message and the message length field, since the DCP data field is variable in length. Problems such as synchronism loss might carry out wrong conclusions about the next messages in the messages block.

The finite states automaton, presented in FIGURE 3, shows the sequence of events (octets read sequentially in the message block) which are analyzed in order to detect both wrong messages and truncation data in different parts of a DCP message. As it might be followed in the TABLE 2, the presented events are sequentially searched for in the message.

EVENT	MEANING
0	response code/message type
1	message subtype
2	satellite code
3	unexpected data
4	synchronism word
5	expected end of data
6	unexpected end of data
7	expected data

TABLE 2: Events Description

The states of the automaton, followed by the SCMCD acquisition module during the DCP messages analysis, are:

INI - The initial state, in which the SCMCD acquisition module stays until the event 0 occurs. Such occurrence means that a beginning of a new DCP message might have happened;

TR1 - The state in which the system waits for the event 1 occurrence in order to follow a beginning of a new DCP message, started in the last state INI;

ZR1 - Analogue to the state TR1, in this case, the event 2 is waiting for in order to complete the three firsts character of the message header;

SC1 - This state represents the search for the synchronism word in the 20th header position. In that state the system waits for the event 4 occurrence in case of 20th header positions matching. If the header position in check is less than 20th, the event 7 occurrence is accepted. That means, any character is considered accepted data in the header until the 20th position;

LEN - Working similarly to the SC1 state, the LEN state deals with the message length information which is used to verify the end of that message followed by the beginning of a new message. In this state, the system waits for the event 0 occurrence immediately after the message length being reached. While the message length has not been matched, the event 7 occurrence is accepted because it represents message data field;

TR2, ZR2, SC2 - These states are analogue respectively to the states TR1, ZR1 and SC1 but they do not deal with the first message in the messages block anymore. In order to check the right length of the last message, the SC2 state has to be reached. That means, the header standard of the new message has been matched;

CFL - It is the state in which the system stays in case of a situation of conflict has happened. This situation happens when no matching of the expected event occurs in one of the following states: LEN, TR2 and ZR2. Therefore, a truncation has been identified in the message;

TR3 - During the situation of conflict, this state holds the system in case of an event 0 occurrence. That means, a probable beginning of a new message has happened;
ZR3 - Analogue to ZR1 and ZR2, in this case dealing with situation of conflict;
SC3 - Analogue to SC1, in this case dealing with situation of conflict;
RIN - This state holds the system whenever the following situation happens: the last message has probably been received wrong. In order to confirm that error, the message standard which occurs in the beginning of the message header has to be searched for in the message data field;
TR4, ZR4 - Analogue to TR2 and ZR2, in this case dealing with the identification of the wrong message;
END - The final state in which the system closes the messages block analysis.

The action, which has to be taken in each state of the system when an associated event occurs, is represented in the FIGURE 3 by the symbols ***Ai***, where ***i*** is the number of the action. The actions are responsible to make conclusions in each specific case of the finite state automaton. In addition, the actions always read the next character in the message in order to identify the next event. As presented in FIGURE 3, the following actions are executed by this finite state automaton:

A1 - It identifies the characters which represent the pattern of the beginning of the message header;
A2 - It is executed whenever an EOF character is found without none message header pattern has been recognized. That means, the message is unknown. It is stored in the data base as a special data in order to be analyzed later on;
A3 - It is executed whenever a message header pattern is supposed to be considered data field;
A4 - It is executed whenever, in the message beginning, a different character is read instead of the expected character in the message header;
A5 - It is executed whenever an EOF character is found just after the last DCP message in the messages block being completed. That means, although some additional characters have been found at the end of the message block, the last DCP message is successful;
A6 - It is executed whenever an EOF character is found in the message data field. That means, the last DCP message in the messages block is truncated;
A7 - It is executed whenever an EOF character is found immediately after the last DCP message in the messages block being completed. That means, the last DCP message is successful;
A8 - It is executed whenever the header standard of a new message is identified immediately after the last message being successful completed. As a conclusion, the last message format analysis was successful and a new DCP message is in analysis right then;
A9 - It is executed whenever an EOF character is found during the recognition process of the pattern that identifies a new DCP message in the messages block. Therefore, the conclusion in that case is that the last DCP message is wrong;
A10 - It is executed whenever, during a situation of conflict, the pattern that identifies a new DCP message in the messages block, is found. That means, the last DCP message has been considered truncated or some extra characters have been aggregated to it. Anyway, a new DCP message is in analysis right then;
A11 - It is executed after the three first octets of the message header being recognized. It works on the other header fields of the message in order to consume them, until matching the synchronism word;
A12 - It is executed after a synchronism word being successful detected. It works on data field consumption;
A13 - Although the pattern which represents the beginning of a new DCP message has been found, the action A13 is executed whenever the synchronism word has not been matched. That means, the last message is considered wrong;
A14 - It is executed whenever, during a situation of conflict, an EOF character is found, just before the synchronism word being verified. In that case, the situation of conflict has not been well solved. As a conclusion, the last DCP message in the messages block is considered wrong.

3. IMPLEMENTATION

In order to implement the SCMCD acquisition module, the finite state automaton which was designed to analyze the DCP message format and presented in FIGURE 3 was mapped in a decision table. This table was implemented in C-language, using a matrix where their rows represent the automaton states and their columns characterize the associated events. The elements of the matrix define both the action which has to be taken in every state of the system for each event occurrence, and the next state in which the system shall remain waiting for the next event occurrence.

The advantage of the decision table implementation is concerned with the walkthrough facility on the table. This mechanism of control is independent of the actions, events and states which are managed by the decision table. Therefore, this kind of implementation allows the code re-utilization. In case another finite state automaton implementation is required, beside the actions specific code, only the decision table contents have to be changed.

Another relevant aspect of the SCMCD acquisition module implementation refers to its capacity of distinguishing the different message formats. That is done by associating each message to a logical channel which is designed by software mechanisms. This is so because the original sites of the messages are different.

4. GENERALIZATIONS

Just like the finite state automaton presented for DCP message analysis in SDID format, other finite state automatons have been implemented in the SCMCD software acquisition module

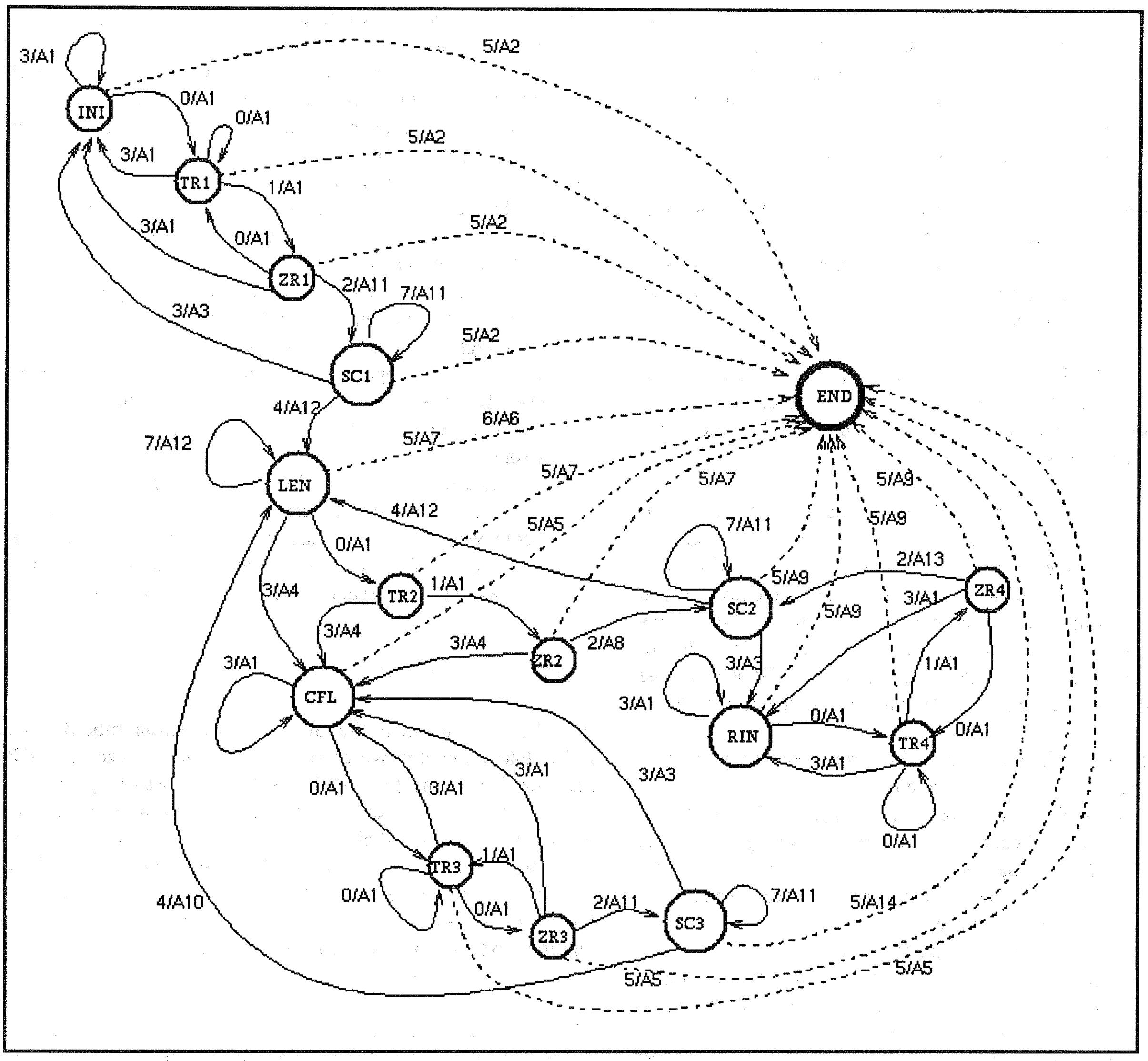

FIGURE 3: Finite State Automaton For DCP Message Analysis

such as the message standards which are used to transfer both orbit prediction and status of the Ground Station receivers from the Satellite Control Center. The automaton designed for that purpose is independent of the automaton presented in FIGURE 3. In that case, because the logical channel is the same for both messages, only one automaton was designed to analyze both messages formats.

Whenever a new format message has to be treated by the SCMCD and the original site of the message does not match a logical channel already in use, the implementation of such aggregation requires only the problem modeling to a new finite state automaton, without any modification in the format analysis previously implemented. Otherwise, some modifications have to be done in the decision table previously implemented in order to enable the existing automaton to distinguish the additional formats being received by the same logical channel.

5. CONCLUSIONS

With this technique, the software SCMCD is already prepared to support the automatic acquisition of other kinds of DCP message formats. It is only necessary to work on the software analysis in order to model the message format problem using a finite state automaton. That work is followed by the mapping of the automaton in the decision table and further by the aggregation of such data structure to the software.

Message features such as fix length and end of message field are very helpful for the message format analysis. These kind of messages might be modeled in a very simple finite state automaton. For instance, the finite state automaton designed to analyze the format of both the orbit prediction and status of the Ground Station receivers messages has only 3 states, 5 events and 6 actions.

In terms of the results of the message format analysis, with the finite state automaton modeling, it is possible to detect truncation and spurious data in the messages blocks received in the Mission Center. By observing format analysis of variable length messages, such as DCP messages in SDID format, the modeling using automaton approach allowed the system to detect, in the message data field, a character sequence like that of the pattern which defines the message header. The ability of modeling such situation enabled the system to re-synchronize the next messages in the messages block whenever a truncated message occurs in the middle of the block. Therefore, re-synchronization yields efficiency in the format analysis since it avoids misunderstanding of correct messages which might be considered wrong just because they follow a truncated message in the message block. Another advantage of the SCMCD software is to report statistics about the quality of the received DCP messages since correct messages are stored in a different way, other than truncated.

By now, all the DCP transmission is done through the Cuiabá Ground Station. Other Receiving Stations, those with different ground communication message standards, shall be used in the near future.

Although the acquisition software module in the Mission Center has not yet implemented the reception of DCP messages other than ARGOS system standard, the aggregation of new formats and also the standard used by the DCPs in the SCD3 will be our next work. With that intelligence, the Mission Center shall be able to store and integrate the different environmental data in a data base, meeting user's needs.

6. REFERENCES

Hopcroft, J.E. & Ullman, J.D., 1979. Introduction to Automata Theory, Languages and Computation. Addison-Wesley, Reading, pp. 13-18.

Francisco, M.F.M. & Gatto, R.C.,1995. Environmental data system for Brazilian Data Collection Satellite. In: Advanced and Next-Generation Satellites, 25-28 September 1995, Paris, France, Proc. SPIE 2583, pp. 515-525.

INPE, 1992. Ground segment communication protocol specification. Report MECB-A-EIF-0004, National Institute for Space Research, São José dos Campos, SP-Brazil.