TIPS TELEMETRY COMPILER*

G. Billerbeck and R. Idsardi
System Development Corporation
Vandenberg, AFB, California

SUMMARY

The TIPS Telemetry Compiler is a keyworded-free format language processor used to generate run-time tables for controlling the real-time software and front-end hardware of the Telemetry Integrated Processing System (TIPS) at Vandenberg Air Force Base. The use of a compiler shortens response time to new requirements and improves analyst productivity.

The Telemetry Source Language (TSL) is the interface between the telemetry analyst and the compiler. For example, TSL statements are used to specify parameters for the telemetry stream, compression algorithms, data acquisition, display, and history recording. Considerable flexibility has been built into the internal structure of the compiler by the use of an Input Control Definition Language (ICDL) to define the construction of the TSL. The flexibility provided by the use of an ICDL to map the source language into the compiler data base is essential for adapting the compiler to requirements beyond the scope of the original construction.

The compiler data base is sufficiently large and complex to require the implementation of data base management and memory management techniques. The nature of these facilities is important for a modular architecture and for reasonable computational efficiency. These key features of the internal structure of the Telemetry Compiler are transparent to the Compiler user. The output of the compiler is a Run-Time File for use by the Real-Time Software in loading the programmable front-end hardware and in software process control.

The TIPS Telemetry Compiler is written in structured FORTRAN on a CDC CYBER 173. The real-time software executes on a network of SEL 32/55 processors. At the time of publication, the framework of the compiler was completed as well as major portions of

* This paper is based upon work performed under contract to the Space and Missile Test Center (SAMTEC) of the U. S. Air Force at Vandenberg Air Force Base, California. Views in this paper are not necessarily those of the U. S. Air Force.

the run-time file generation phase, and the compiler had been used to build run-time files for major development milestones.

OVERVIEW

The purpose of the TTC is to provide the configuration and control data needed by the Real-Time software and hardware of TIPS. For a description of the TIPS hardware and software elements reference papers presented at this conference. To provide the setup data to the TTC the Telemetry Source Language was created to allow the user to input the data in a natural language.

Figure 1 depicts a functional overview of the TTC. The Input Control Definition Language is input to the TTC Data Base Dictionary Generator to produce the TSL language defintion in the form of a dictionary file. The dictionary is used to define TSL input and to define the storage structure of the TSL in the data base.

The TIPS Telemetry Compiler (TTC) user creates a data base through TSL input that describes all telemetry input formats processing requirements, and display descriptions for a vehicle in a batch or interactive mode. From this data base reports and a Run Time File (RTF) are created. The RTF contains tables and program loads which control the processing of the real-time telemetry subsystems of TIPS.

The TTC provides the user the capability to enter, save and recall TSL data, copy saved data from one CDB file to another, and edit or change individual TSL parameters in the CDB without a requirement for re-entering unaffected values.

TELEMETRY SOURCE LANGUAGE

Compiler source statements are written in the Telemetry Source Language (TSL). The TSL is a keyword oriented, free format input language which allows the user to specify his vehicle, processing requirements in a concise, unambiguous meaningful language. The hierarchical data structure implicit in the language forms relates data in a meaningful way for easy access by the user and the TTC. In order to reduce data base preparation and user learning time, the interactive mode of input provides a prompting display which only requires the user to fill in the blanks. The interactive processor allows the user to request additional information from the data base to help him fill in the blanks. Any errors of commission or omission are promptly noted with messages designed to assist the user to correct the error.

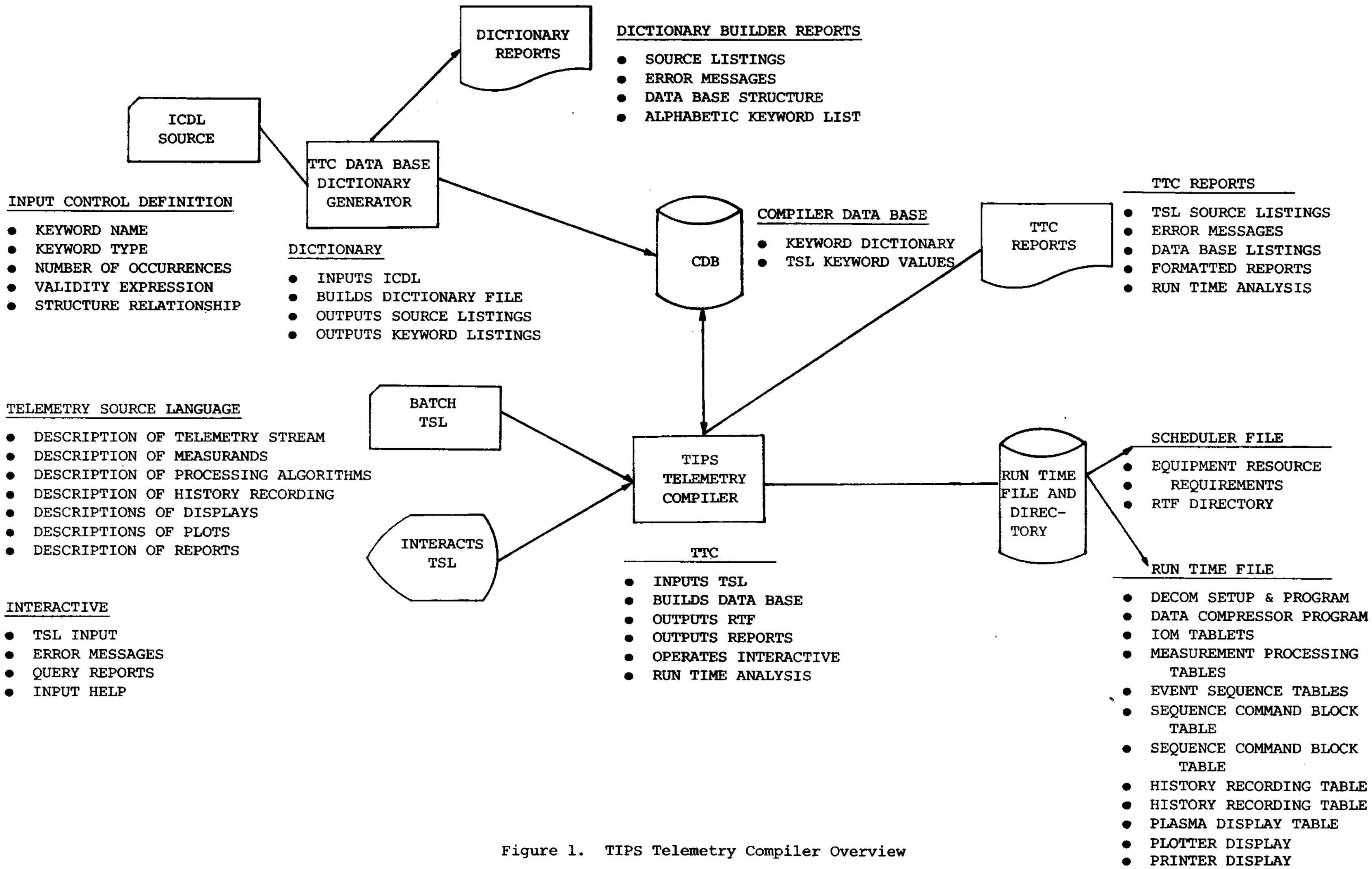

Figure 1. TIPS Telemetry Compiler Overview

The data base and TSL are hierarchically structured as shown in figure 2. When read from top to bottom and from left to right the figure shows the sequence in which the user describes the vehicle. The sequence of input is important because, if the sequence is known, then each source statement input can be used as a basis for validity checking of subsequent dependent data.

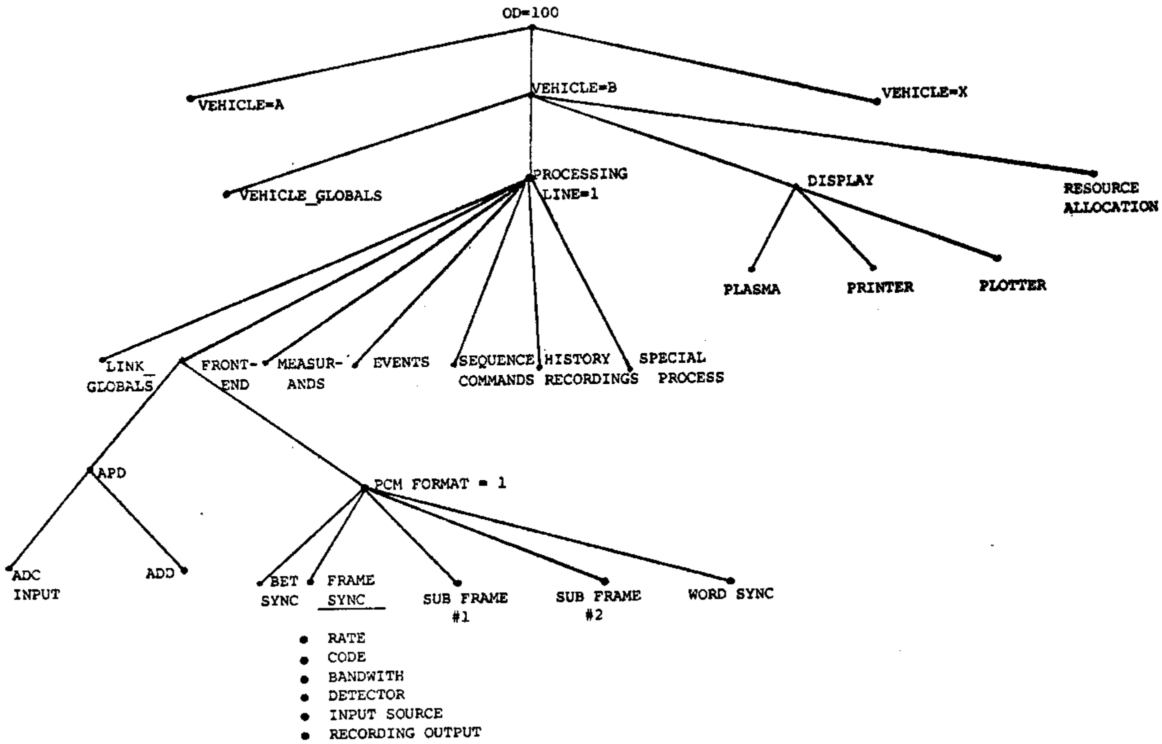

Figure 2. TSL Hierarchy

For example, a measurement is to be displayed on a plasma format. The measurement name to be displayed is checked against the measurements defined in the measurand section to verify that the display name has been defined.

The hierarchical structure of the language is used as a means for prompting the interactive user.

The entries in figure 2 are described below.

OD (OPERATION_DIRECTIVE) assigns all descriptions below as belonging to named OD. Each project on the range is assigned an OD, therefore the keyword separates the data base into projects.

VEHICLE assigns a unique name to the data base description of a unique vehicle. Each project usually has more than one launch vehicle.

VEHICLE_GLOBALS define identification data unique to the vehicle.

PROCESSING_LINK defines all processing links for the vehicle.

LINK_GLOBALS define link unique global parameters to the processing link.

FRONT_END defines synchronization setup parameters such as time code translation setup.

MEASURANDS define measurement attributes such as limits, engineering units, descriptions type, decommutation source, and data compressor algorithms.

EVENT_PROCESSING defines event sequences and command actions to be initiated as a result of event conditions.

SEQUENCE_COMMAND_BLOCK defines the commands for controlling and effecting processing.

SPECIAL_PROCESSING defines special processes to be executed by the RTA system.

RECORDING defines history file processing.

DISPLAY defines a keyword group specifying the format and content of printer, plotter, and plasma display.

RESOURCE_ALLOCATION assigns processing and display requirements per scheduling functions.

INPUT CONTROL DEFINITIONS

The Input Control Definitions are an innovative approach to the solution of the problem of changing user requirements. Using the Input Control Definitions, the user can modify the existing TSL input forms and add new ones.

The Input Control Definitions contain information about the syntax of the input, the type of value, the range, limits, constraints such as it must be previously defined, whether it’s optional or required, etc. New keywords can be defined and they will automatically be displayed as prompts in the interactive mode of input.

The ICDL provides the capability to add new features to a language or system without modifying the TTC. This feature of TTC is designed partly as a requirement and partly to maximize the life cycle of the TTC at least cost. The user can modify existing input forms and add new ones through the Input Control Definitions. By adding special processes, he has added new real-time processing algorithms, all without modifying the TTC or the real-time software. New definitions or modifications can be accomplished through changes in Telemetry Compiler language statements.

TTC INTERACTIVE OPERATION

In the interactive mode of TTC operation, source statement input is from the interactive terminal in response to prompts generated by the TTC. Through the use of control statements and prompts the TTC provides the capability for the telemetry analyst to control the TTC and manipulate the compiler data base with a minimum of input.

The interactive SCREEN formats are generated from the dictionary in the CDB. Therefore a change in the TSL does not require changes in the TTC interactive form generation software.

In the interactive mode the user is provided three input capabilities. These modes of input are:

- TSL input, where the user is supplied keyword names and values if they exist. The user may enter new values or modify old values as he chooses. The TTC guides the user per the hierarchy described in figure 2. At the termination of the hierarchy are groups of keywords defined as data groups. These groups of keywords are displayed on the screen and value input is solicited by the TTC. After user input the TTC solicites the next value input.

- TTC data base, in this mode the user may copy, delete, query or simply browse through the data base if values exist.

- Interactive aids. The TTC provides tutorial help by displaying information about a keyword which is stored in the dictionary. Screen management controls are provided which allow the user to change modes or skip lines while inputting data.

COMPILER DATA BASE

The Compiler Data Base (CDB) provides the user with all the capabilities associated with a sophisticated data management system:

a. Interactive dialogue,

b. Query language to access any single item or group of items in the data base,

c. Hierarchical data structure for expressing complex data relationships and grouping data items according to usage,

d. Report generation tailored by user requests.

The CDB is the internal interface within the TTC for all processing functions. All functions within the TTC either input or output data to/from the CDB. The dictionary generator creates the TSL definition and storage dictionary and this dictionary is one logical part of the CDB. The batch or interactive TSL input functions of the TTC input source statements and store the resultant values in a linked list with the second logical part of the CDB. The CDB is the source of data for the run time file generation. The RTF code generator requests data from the CDB by logical name, therefore isolating the code generation from physical storage representation and the TSL input. By having the RTF created from the CDB, the user can build the CDB over a period of days before requesting the RTF be generated.

CONCLUSION

The TTC reduces range support lead time and vehicle preparation time by providing a self documenting language which easily configures the TIPS for new vehicle support. The TTC allows the user to configure the general purpose TIPS software to meet the specific needs of each user. This is possible because the real-time software is table driven and the TTC generates the tables from analyst TSL input.

Finally, the TTC is extensible because of the dictionary generation capability which allows the language to be expanded without changing the TTC software.

REFERENCES

1. Van Dolsen, L. L., “TIPSAn Integrated Solution for Multi-Mission Telemetry”, Proceedings of the International Telemetering Conference, Volume XIV, International Foundation for Telemetering, Los Angeles, 1978.

2. Straehley, E. H., “TIPS Real Time Acquisition, Processing and Display Subsystem”, Proceedings of the International Telemetering Conference, Volume XIV, International Foundation For Telemetering, Los Angeles, 1978.