Data perennial description : the syntactic aspect.

Authors : Arnaud LUCAS (Arnaud.Lucas@cnes.fr)
Denis MINGUILLON (Denis.Minguillon@cnes.fr)

1 - The problem

Data needs to be described. A set of bits on a medium is of no use without the right explanations concerning its coding and its meaning.
A good description must remain understandable despite the evolution of the technology. A sharp description in an obsolete format is the same as no description.

There are at least two ways to describe data :
- The first one is to freeze a format. Each instance of data will then be compliant with this known format. This can be convenient for a particular domain where data is very stable.
- The second one is to let the data format be totally free and to provide a normalized way to describe any format.

The current article presents the EAST language that is a solution of the second type to describe in a normalized way any kind of data.

2 - Brief history

The EAST language is a CCSDS (Consultative Committee for Space Data Systems) recommendation which became an ISO standard (ISO 15889) in 1997. EAST stands for Enhanced Ada SubseT. In fact the declarative part of the Ada programming language covered 95% of the needs in data description. It was already a model in terms of normalization and the CCSDS just had to add a few missing features to write a complete and accurate recommendation.
As a standard without tools as no chance to be widely accepted, the time since its release has been used to build many tools. The current paper presents them and the gains they brought in operational projects.

3 - The principles

The main assumption is that data can always be seen as an ordered set of bits. This set of bit is to be considered as a data tree which leaves are the data fields carrying the information.
So, to describe such a tree, EAST proposes records and arrays for the branches, and final types (as integer, float, characters and enumeration) for the leaves.
This has to be completed by means to indicate if data is ascii or binary and which encoding are used for numeric fields (weight of the bits and formulas).
Finally some features to deal with conditional fields depending on the values of other fields are proposed by EAST.
The combination of these characteristics is sufficient to describe most of existing data as proved by the usage made since 1997 on several projects.

4 - The tools

4.1 - OASIS

Users were not expecting for a new language to learn. So, the decision was made to provide them with a tool dedicated to data description that would produce automatically an EAST description as an output (no need to master the syntax).
This tool is named OASIS (French acronym) and can roughly be described by the two following windows.

Figure 1 shows the data tree being built, with the current node shown in blue

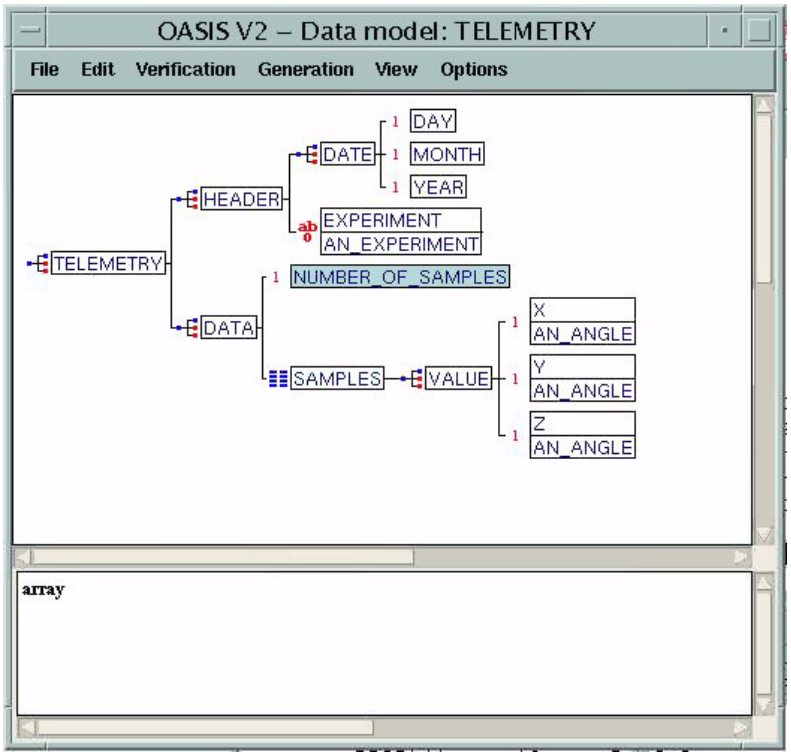

Figure 1 : the data tree window

From this example we can see the described telemetry begins with a header providing the experiment which produced the data and the date of this production.
This header is followed by an array containing the values of the measurements of the experiment.
The number of elements of this variable size array is given by the field "number_of_samples" and each of its elements is a record grouping 3 angles (X,Y an Z).
The leaves of the tree (in red in the following list) are the effective fields on the medium :

TELEMETRY.HEADER.DATE.DAY
TELEMETRY.HEADER.DATE.MONTH
TELEMETRY.HEADER.DATE.YEAR
TELEMETRY.HEADER.EXPERIMENT
TELEMETRY.DATA.NUMBER_OF_SAMPLE
TELEMETRY.DATA.SAMPLE(i).X
etc...

To know what are the precise characteristics of each of the nodes of the tree we can refer to figure 2.

Figure 2 shows the characteristics of the current node

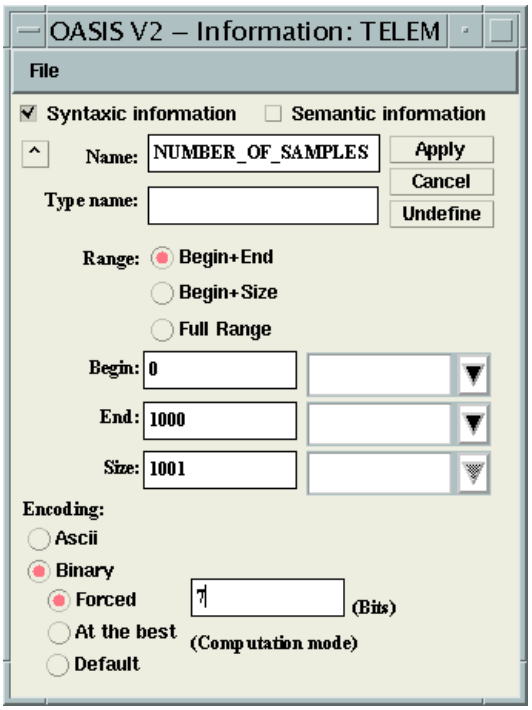

Figure 2 : the node characteristics window

From this second window we can learn that the field named "number_of_samples" is an integer for which the range of possible values is from 0 to 1000, coded in binary format on 7 bits.

The best way to know all the possibilities provided by OASIS is to ask for a demo at the e-mail address : east@cnes.fr, or/and to download the software from http://logiciels.cnes.fr.

Once the data has been described through the OASIS MMI it is possible, just by a click, to obtain its complete EAST description. Once saved (e.g. : in the file telemetry.eas), this description can be read and used by the following tools.

4.2 - The INTERPRETER

Continuing with the telemetry example we can imagine that a user software has to read the telemetry values. As this telemetry is very sharply described by the EAST description, a software library, named the interpreter can deliver to this application any of the fields on request.
This makes the user application independent from the coding of the fields on the medium.

Example : To get the value of the field "number_of_samples" the user application just has to provide a service named "get_data_entity" with the path string :
"TELEMETRY.DATA.NUMBER_OF_SAMPLES",
and it will get the value of this field found on the medium.

In case of evolution of the data (e.g. : new fields added in the header), as long as the path string identifying the wished field is not modified, the user application will not be impacted by the

changes.
In case of heterogeneous machines (e.g. PC data read on a Unix machine) the conversion is automatically done by the interpreter.

4.3 - The GENERATOR

A user application may also have to write, on a medium, values conforming to the telemetry format. In that case values have to be given to the sofware library named the generator. As soon as each of the telemetry fields has been given a value, the user application can ask to the generator to write the data on the medium. Here again, the user application has not to deal with the coding of the information on the medium.

4.4 - The standalone tools

The interpreter and the generator are available as APIs (Application Programming Interface) for C, C++, Ada and Fortran applications. On top of these low layer facilities, some standalone tools have been developed and can be used to check, display or simulate data.

4.4.1 - The data_checker

This is a tool to check the consistency of the data with regard to its EAST description. If a data file is entirely compliant with its description, the results of data_checker is an empty file. In case of inconsistency between the data and its description the result file contains the name of the inconsistent value(s) and the corresponding wrong value found on the medium.

4.4.2 - The ascii_dump

This tool is particularly interesting to display the values found on binary files. For each field, it prints its name and the corresponding value found on the medium. If the encountered value is not a valid one, it prints "Bad value".
Once the data_checker has stated that the data is consistent, ascii_dump can be use to check visually if it looks meaningful.

4.4.3 - The Data_Update_Wizzard

This tool allows its users to generate data conforming to an EAST description without having to write an application. The user can (through a graphical MMI) generate data at random and then modify the random values if he wishes.
This is particularly useful to generate test data for integration purpose when real data is not yet available.

5 - EAST and perenniality

The first projects that have used the EAST technology to handle their data have suffered from its youth defaults. In particular reliability and performance were not there.
Now, the performances have improved a lot (the most impressive benchmark showed a ratio of 1 to 1800 on some files). There are still improvements to make.
The reliability is now quite good and, when bugs still occur, the maintenance team is very reactive and generally, a corrected version is available within an average week.

So, now, the advantages look much more obvious.

- As said above, the main advantage is to make the application independent of the coding of the data, so a comfortable logical view (each data element being designated by its name) is sufficient to read and write the data.
- Another main advantage is to define strictly the interface between different sub-systems. This saves time during the software integration phases.
- Later in the data life-cycle, when data has to evolve, the consequences on the existing applications are very lowered.
- OASIS associates in a unique file the syntactic description and the semantic attributes. It can gather both kind of information in a unique document (cf. other presentation on the semantic aspect).

EAST also contributes to the perenniality of data by making them independent from the current software and hardware. Even if some fields are dependent from the machine on which they have been generated, their EAST description will give all the clues to interpret them after this machine will have disappeared.

When describing data with OASIS a user has to fill all the syntactic attributes of each of the fields. There is no possible lack in the description. So, the fact that people aware about the data will not be available for ever is no longer such an important problem.

As an example we can take the CDDP archive centre (in charge at CNES of Plasma Physics Data) which made the choice to request an EAST description for each data set it would archive. This guarantees the consistency of all the descriptions and their completeness.
Based on these required EAST descriptions, a data extraction service is offered to the users through the web. A generic facility (calling the interpreter) offers to select only a few fields.

The fact that, through the EAST descriptions, data is seen as a bit stream entirely described , allows a project to entrust its data to a generic storage service the interface of which is at the bit stream level.
That is the case for the CDDP that relies on the CNES STAF service to store its data. This guarantees to recover the same bit stream in many years from now.

The current list of user projects is available on http://east.cnes.fr

6 - Conclusions

The lesson learned from the current usage of EAST on several projects is that it is crucial to have a way to ensure the perenniality of data. It is very convenient to find this way equipped with tools to deal with data.
This assumes that the data is described at the earlier stage by people involved in its definition. The formal description can, afterwards, be delivered to anyone who has to deal with this data and who will take advantage of all the existing tools.
Note that, even if there is no guarantee on the perenniality of the tools (this is never achieved for ever) there is a guarantee on the perenniality of the description and so, the assurance to be able to interpret it in a far future.

7 - Links

To know a little more about EAST and user projects : http://east.cnes.fr
To download the tools : http://logiciels.cnes.fr
To get the standard documents : http://ccsds.org/document_access.html

For any question and/or need of support, please e-mail to : east@cnes.fr