PREVIOUS DISCUSSIONS (NOT RELEVANT NOW)

On Fri, 3 Dec 2021 at 11:15, MOLINA Francisco Javier <Francisco-Javier.MOLINA@ec.europa.eu> wrote:

Yes, I think that it is safer to have always at least one row of one note, even if it empty. Anyway, the cost in memory is almost null.

Javier

From: Nacho ... <ignacio.lm@gmail.com>

Sent: 03 December 2021 11:01:52

To: MOLINA Francisco Javier (JRC-ISPRA)

Cc: POLJANSEK Martin (JRC-ISPRA); PETIT Patrick (JRC-ISPRA); PEGON Pierre (JRC-ISPRA); CAVESTRO Jessica (JRC-ISPRA-EXT); MARTORANA Emilio (JRC-ISPRA-EXT)

Subject: Re: Progress update and Next steps

Dear Javier,

This is fine. If I am not wrong, I think whay you describe is similar to the current behaviour, that the row in the CSV must always exist, which can be empty if there are no notes.

Although technically it is not a problem to infer that if the row is not there, then it is assumed that the notes are empty. We can do this for the signal upload and treat CSV files with empty or no note field equally. But the download is different, and if you think this can be confusing we can always add the blank row.

We will be in touch,

Nacho.

On Fri, 3 Dec 2021 at 09:38, MOLINA Francisco Javier <Francisco-Javier.MOLINA@ec.europa.eu> wrote:

Dear Nacho,

Yes, I think that the separator "---" is very positive for the computers and for the human eye.

Additionally, we could also oblige that the minimum number of notes is 1, in order to prevent some possible errors with empty fields that may happen to us, but also to external users when they have to process the csv files.

Thank you for your fast reaction and useful input.

Best,

Javier

From: Nacho ... <ignacio.lm@gmail.com>

Sent: 02 December 2021 18:00:55

To: MOLINA Francisco Javier (JRC-ISPRA)

Cc: POLJANSEK Martin (JRC-ISPRA); PETIT Patrick (JRC-ISPRA); PEGON Pierre (JRC-ISPRA); CAVESTRO Jessica (JRC-ISPRA-EXT); MARTORANA Emilio (JRC-ISPRA-EXT)

Subject: Re: Progress update and Next steps

Dear Javier,

This is a great request to get hands-on experience on the application, so I will code it together with Emilio. It might take us a bit longer to finish since we will have to do it over shared screens.

The discussion we had was about the semantics of the data, and I still think that they have to be maintaned as they currently are, meaning that the field is just a "notes" value that only has meaning within the context of an experiment. Actually, I wrote about this in the documentation I am doing, about two weeks ago. To refresh my memory, I reviewed my hand written notes, the ontology notes, the old ELSADB database and our exchanged emails. Feel free to check the draft below [1].

Keeping the semantics is not in contradiction with your request: we can store the data one way and present it in a more suitable way, and this is actually how the GUI works - the notes are presented each one in a different column, even though they are ingested from a single field in the CSV file and stored as a single string in the database. This changes only affects the CSV format, which is the format used to download and upload signals, but the database might remain unaffected as long as you still do not use notes containing "##". However, I will check if this symbol has to be modified for something more appropriate, and in this case the database will be slightly affected.

My only concern with your request is about the use of a field containing the number of notes ("n0: no notes"). This can be error prone and errors will be difficult to detect. My suggestion is to use a separator instead, like "----", before the values start. Example:

signal1, signal2, signal 3, sognal4

....

note N-1, note N-1, note N-1, note N-1

note N, note N, note N, note N

----,----,----,----

0, 0, 0, 0

1, 0.03, 0.01, 0.1

2, 0.06, 0.04. 0.7

...

This separator method has two advantages:

1- You do not need to keep track on how many notes you have, you simply write them.

2- Errors can be detected. If you put n0=5 and by any reason you write only 4 notes, it will be almost impossible for the computer to detect the error. However, if you forget to include the separatod symbol "----" before the values, then the error is immediately detected.

Note that some of the existing rules still apply: you will still have to leave blank spaces if a note exists for some signals but not for others:

signal1, signal2, signal 3, signal 4

....

South, North, , South

A,B,C,D

----,----,----,----

0, 0, 0, 0

In the example above, Signal3 does not have an "orientation" value, but it does have a "letter" value, so the "orientation" value is simply empty, having nothing between the two commas: ",,". In the current version, this same thing requires to have the value empty between the symbols, e.g.: "####".

As soon as Emilio has the development environment installed, we will start.

Thanks,

Nacho.

[1]

Signal Notes. The signal notes reason of existence goes back to the times of the ELSADB database. When conducting an experiment, there is some information that does not accommodate in the existing data model, but which is useful to keep. For example, during the experiment there might be a sensor that is located facing North, and this information is relevant or has some interest within the context of the experiment. It was a requirement that such information is stored, so in ELSADB this was modelled as a free-string attribute in the signals table (strictly speaking, the orientation is not an attribute of a signal, but of the device that generated the signal, but we will assume that it is correct to set it in the signal table as it is an information useful in the context of the signal, and not of the device itself). This attribute was insufficient by itself, so five attributes were created in the table to hold potential data, called “positions”, also known as: param(1), param(2), param(3), param(4) and param(5).

This solution has some problems:

· It is a wrong technical approach. A number of undefined attributes should be modelled not as attributes but with additional tables. Otherwise, the attributes are both limited and fixed in number (five), so if an element requires less than five attributes, the rest must be “nullified” and if it requires more than five, the database schema has to be modified.

· It has no defined semantics. The attributes work as a “store-anything-you-wish” box, a place to store miscellaneous data.

· It is not scalable. It is true that more than five attributes might not ever be needed, but the limitation exists if more data are necessary.

During the development of CDV, it was a user requirement to keep this information. The use of a property to store arbitrary data was not a very welcome idea from a data modeller perspective so it is one of these times where developers and users need to come into a balanced agreement.

The main issue was the lack of semantics, so an analysing of the sort of information that is stored in these attributes was started, in order to see if multiple data properties could be created in the ontology. It soon became clear that the information stored was useful for the researcher but meaningless outside the context of the experiment. For example, the property “orientation” could be created, but this only had value to compare one signal data to other signal data in the same group of experiments. A value of “north” in this property could be perfectly be “east” if the same exact experiment had been conducted in another building, with the same exact results, so it was meaningless as a global concept, yet it was very relevant for the user in the context of the experiment. The creation of an “orientation” property had no real value if we needed to compare data from two different laboratories or data sources. Likewise, there was not predefined way or number of useful information, so for one experiment the interesting data are the floor on which the sensor that recorded the signal was located, for another it could be the orientation, or any sort of arbitrary detail the researcher wanted to record a local classification of the obtained signal data. This made it clearer that an “additional notes” could be used to model it, since it was something that only got meaning within the context of the specific experiment and under the point of view of the test agent in charge of it. These additional notes can hold information like “A/B/C/D...”, “YES/NO”, “ON/OFF”, “level1/level2/level3...”, “North/South/East...”, “S1/X/Y...” and any other sort of information that might be meaningless out of context. Defining the semantics of the information as “notes” instead of a limitless amount of properties, that were barely reusable and not really meaningful externally, was an acceptable approach. So the first issue was resolved.

The second issue was the appropriate modelling of the number of notes. For these data, the elegant solution is to use an RDF collection or RDF sequence container. However, in the end it was more practical to use a single literal that uses a separator (##) to inform the visualisation tool that the information of the string is more useful to be presented in different groups that can be sorted and compared between them (e.g. C##level2##South). This actually looks like quite an ugly solution, but it works considering that the semantics of the data as just an internal note that the researcher is making, and the separator is only to specify a displaying indication, and not a storing indication. This solution can be found valid within the semantics and has greatly simplified the retrieval and creation of these data. However, it is open for change to, e.g. RDF collections, in future, once the data prove their actual use in ELSA and other laboratories

On Thu, 2 Dec 2021 at 16:55, MOLINA Francisco Javier <Francisco-Javier.MOLINA@ec.europa.eu> wrote:

Dear Nacho,

Now that a new version is being produced of the system, I want to change slightly the format of the csv files through which the signals are downloaded or uploaded to ELSADATA. The change concerns mostly the way the "notes" attributed to the signals are separated within the file, which until now was done within the same row with two # characters between them, but now it should be with a separate row for every note. This is then linked to a little addition in the file name that informs of the number of notes. The number of notes will be also explicit in the first row of the file.

I remember that you and me had discussions in the past about this separation of the notes and you convinced me, but after these years of practice and the exchange with some external users, I have realized that it is really important to separate different notes in different rows in order to make the file easier to read by the human eye.

I am attaching a text file with the explanation of the new format.

Please tell me what you think about this.

Thanks and regards,

Javier

---------Source

CTRL: controller acq

STD: standard acq

PHYS: physical DoFs

HYBR: hybrid system

---------Sampling

av: record average

ins: instantaneous

con: constant average

re: resampled

---------Elaboration

ORIG: measured (or controller computed)

DER: derived by posttreatment

IDEN: identified by posttreatment

---------Number of notes

n0: no notes

n1: 1 note

n2: 2 notes

...

nX: X notes

Examples of group names:

CTRLavORIGn5 - Source: controller acq | Sampling: record average | Elaboration: measured | 5 notes

CTRLinsORIGn5 - Source: controller acq | Sampling: instantaneous | Elaboration: measured | 5 notes

CTRLconORIGn5 - Source: controller acq | Sampling: constant average | Elaboration: measured | 5 notes

CTRLreORIGn5 - Source: controller acq | Sampling: resampled | Elaboration: measured | 5 notes

CTRLavDERn5 - Source: CTRLav0 | Sampling: record average | Elaboration: derived | 5 notes

STDavORIGn5 - Source: standard acq | Sampling: record average | Elaboration: measured | 5 notes

STDavDERn6 - Source: STDav0 | Sampling: record average | Elaboration: derived | 6 notes

PHYSavORIGn3 - Source: physical DoFs | Sampling: record average | Elaboration: equation | 3 notes

PHYSavDERn3 - Source: PHYSav0 | Sampling: record average | Elaboration: derived | 3 notes

HYBRreIDENn2 - Source: hybrid system | Sampling: resampled | Elaboration: identified | 2 notes

File name is composed as:

{experiment name}\_{group name}.csv

Example of signals file name for experiment a15:

a15\_CTRLavORIGn4.csv

Structure of the csv file:

- One column for every signal.

- For every column:

row 1: group name and description

row 2: signal name in the group

row 3: signal description

row 4: signal magnitude

row 5: signal unit

row 6: signal note 1 (if it exists)

row 7: signal note 2 (if it exists)

...

row 5+X: signal note X (if it exists)

rows 5+X+1 to 5+X+Nsamplings: signal samplings from 1 to Nsamplings

Note that all the signals in a group have the same number of notes and the same number of samplings

so that they use the same number of rows in the csv file.