AMERICAN SCIENTIFIC PUBLISHERS

Call for standardization in material behavior assessment systems output formats

Vencel Biró, Dorel Banabic
CERTETA - Research Centre in Sheet Metal Forming, Technical University of Cluj-Napoca, Cluj-Napoca 400114, Romania

During the development of a central platform that connects measuring systems, simulators and fabrication machinery through internet, helping the metal forming process chain, the biggest challenge turned out to be something surprising. The biggest challenge was not finding the correct technology to communicate a high load of data through the internet or circumvent firewalls or security. The biggest challenge turned out to be implementing the extraction of data from measurement results to be used in simulations. This data needs to be prepared by lab assistants. Because of the different result formats using pre-defined patterns is impossible so the researchers are forced to define their own extraction pattern which is extremely complicated. A common data format standard should be followed by all the measuring system manufacturers so the generated data can be easily processed and shared between different systems without demanding extensive programming knowledge from researchers.

Keywords: communication, web, process chain, non-real-time systems, measurement, simulation, fabrication.

1. INTRODUCTION

The paper will present the DaCoTraP project in development that contains the critical part that requires the standardization support. Then a few experiment types will be detailed and we try to offer a solution to the problem.

To define a generic mechanism that extracts specific parts of differently structured data, in our case experiment data, requires the invention of a new markup language. Fortunately such a language exists since 1950, it is very powerful and well documented [1] and it is used to manipulate text and data. Unfortunately this language is extremely complex and we cannot expect the users of the system, who are mostly lab assistants with training only in the field of metallurgy and with no programming experience at all, to define these patterns by themselves.

If the experiment results were defined in files with strict and standard formatting rules, a very rigid and simple toolset could be defined and put to the users' disposal.

2. DACOTRAP PLATFORM

A standard and simplified manufacturing process includes machines in different rooms, buildings, even continents. Testing devices determine the parameters of a work piece; simulation equipment processes the data from the simulation equipment. All these results are used by the cold presses that use simulation data and the measuring data to test the manufacturing of the part by creating the actual item. The data between the testing, simulation and manufacturing devices is usually transmitted with optical discs or other types of storage devices.

The purpose of the project is to eliminate the need of these storage devices by allowing the different machines to synchronize between them using the internet. Using the platform offered by the project machines will be able to communicate with each other in almost real-time independently from their location. The solution to the distance problem is a platform capable of collecting data, processing it and serving it back upon request. The platform needs to be generic enough to be used with

multiple machines (by offering interfaces and data formats that are standard within the system). It also needs to be easily accessible by the authorized personnel and needs to be able to scale well. Besides the communication the system will be able to keep track of many other parameters of the devices like status, error messages and statistics. The project can be applied in manufacturing environments where the measuring, simulation and manufacturing equipment is separated by distance. Every step of the manufacturing process chain has to have access to the internet[3].

The project is built on the mechanism of remote devices/clients communicating with the system by sending special packets through HTTP to a web service and accepting the response packages. The received data is saved into the database. A Windows service organizes the database and executes time consuming tasks. The users can see their devices, statistics and manipulate data using a web application. The bundle has an SQL Server database in the center. The database is the only connection between the different components because those components can be located on separate servers. It stores every project-related data.

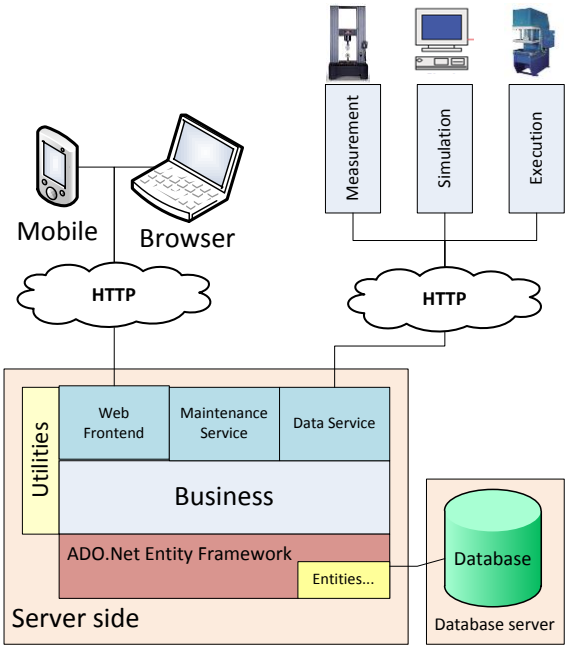

Fig. 1 - The DaCoTraP project architecture

The next important component is the web service called Data Service. This web service is collecting data from the devices, storing it in the database and serving back data to the connected device as a response to its HTTP request.

The Maintenance Service Windows service performs maintenance tasks on the database. It archives old data, deletes items marked for deletion, sends email as part of the common functionality for both projects.

The Web Application allows users to manage their devices, look at statistics and issue simulations. It avoids handling tasks that take long to execute and could lower the user experience. Even the sending of an email is delegated to the Maintenance Service.

The clients connecting to the Data Service are of many types: Gumstick PCs, Windows PCs, Linux devices, phones, and different types of machines that have an operating system capable of running 3rd party software. The software differs for every type of device and it is usually developed outside the range of the projects.

An important quality of the system is that the area affected by the software does not have to be limited to a single factory; multiple machines from factories, laboratories, solar plants from all over the world can work together by communicating, building a central knowledgebase and generating information.

To circumvent the different firewall-, protocol- and port restrictions the data communication uses HTTP and port 80 because these are the most likely enabled protocol and port. Besides only using the port 80, HTTP and REST-full communication the platform tries to utilize all the latest standards[6][7][8] to build a durable, maintainable application.

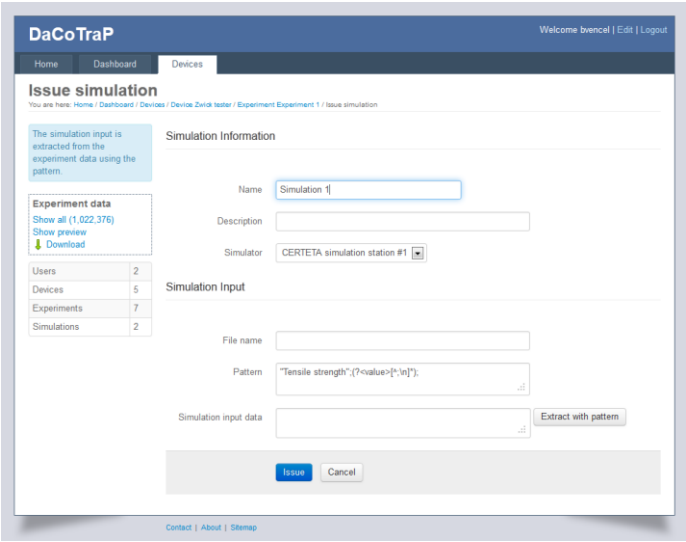

Fig. 2 - The clean interface of DaCoTraP made modern by Twitter bootstrap[1]

The application also takes the head-on opposite of the approaches described in most modern science books[4][5] which surprisingly still advocate old school ugly approach to interfaces. So instead of using ugly Java interfaces that try to resemble the actual equipment's dashboard DaCoTraP opts for a modern clean HTML interface.

3. DATA STRUCTURES

Each type of experimental data differs from manufacturer to manufacturer. The guideline the application tries to follow is to keep the data generic so

[1] http://twitter.github.com/bootstrap/

there won't be any manufacturer-specific database tables. The intention is to keep everything generic above the Service line. Looking at the flow direction of the experimental data on **Fig. 3** the arrow highlighted with red indicates the crucial point where the standardization is important.

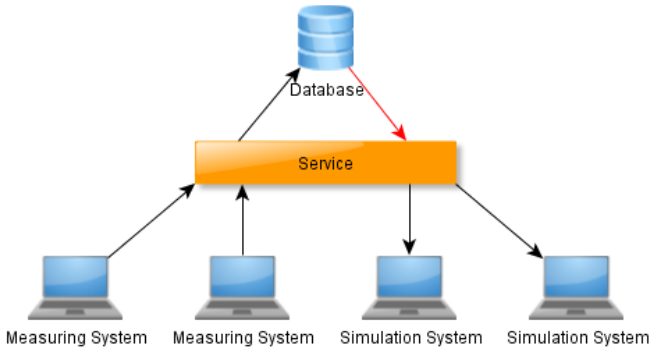

Fig. 3 - Direction of experimental data

We are only going to present three types of file structures but these examples will well illustrate the problem that needs to be solved. The full length of the packages cannot be shown in this paper because of lack of space, instead the important parts of the experiment data will be presented and some example simulation inputs derived from them.

The Zwick testing machine testXpert II software[2] produces the following ASCII/text-based output:

```
1  "Specimen thickness";1;"mm"
2  …
3  "Uniform elongation";25.4087;"%"
4  "Specimen width";20;"mm"
5  "Tensile strength";261.323;"MPa"
6  "E-Modulus end";80;"MPa"
7  "r-Value iteration depth";11;""
8  "Initial cross section";20;"mm²"
9  …
10 "Change in width";"ln true strain";"ln true
   stress";"Logarithmical width shape
   change";"Specimen width";"Standard force";"Standard
   travel";"Strain";"Strain (Plastic)";"True
   stress";"True strain";"vertical anisotropy";"Work
   hardening exponent";"Nominal strain";"Work"
11 "mm";"";"";"";"mm";"N";"mm";"%";"%";"N/mm²";"";"";"
   ";"%";"J"
12 0;0.0;0.0;0;20;3.43206;9.9659e-
   005;0.000124574;0.0;0.0;0.0;0.0;0.0;0;3.44994e-007
13 9.9659e-005;0.0;0.0;-4.06779e-
   006;19.9999;3.80028;0;0;0.0;0.0;0.0;0.0;0.0;-
   8.20365e-006;-3.04135e-012
```

The blue section (lines 1-9) denotes the part of the file that contains title/value/measurement unit format information. The green part (lines 10-13) contains the CSV format tabular data used to describe the chart plot. The values highlighted with yellow (lines 3, 4, 5, 6) are used when building the input data for one of the simulators:

```
// Zwick simulation data
 1  25.4087
 2  20
 3  261.323
 4  …
 5  4
 6  507
 7  0.0001
 8  0.282
 9  1.0
10  0.996
11  0.00015
```

The issue is that the differently formatted data is contained in the same file. Another issue is that the measurement unit is included as a separate piece of information. If for example the specimen width is needed in cm (line 4), the value has to be transformed cm after determining what measurement unit was it defined in the first place.

A single value can be extracted from the experiment data using the following regular expression:

"Tensile strength";(?<value>[^;\n\r]);*

(=Take the value after the part '"Tensile strength"' until a ";" character or new line is reached and save it as a result).

The above formula only defines a single value; each value uses a different pattern.

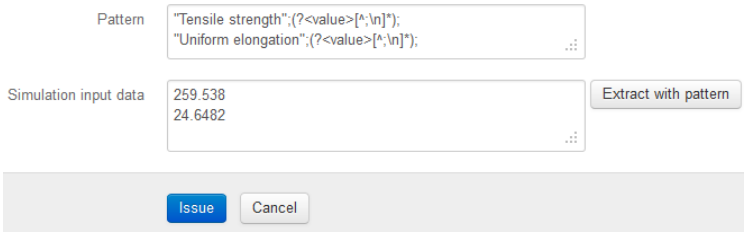

Fig. 4 - Screenshot of the 'Issue simulation' form of the DacoTraP project illustrating the current complexity of the formula that extracts two simple values from the experiment data

The Erichsen experiment file format raises another issue: some values are given in floating point format (line 4) while others have decimal format.
The Erichsen simulation data has to be extracted from values that are again differently formatted from the other manufacturers: instead of using the title/value/ measurement unit combination separated by semicolons as in the case of the Zwick equipment the title is separated from the value with an equal sign (=) while the measurement unit follows the value enclosed in straight parenthesis (ex. [kN]).

```
1  System File=C:\Users\Erichsen\Desktop\Mes-
   Cluj\MES.sys
2  Comments=[1]
3  …
4  START_SHF_DS;0; 1.00000000000000E+0001;
   1.50000000000000E+0001t
5  …
6  Date=4/5/2012
7  Time=3:34:11 PM
8  Tester=Erichsen
9  Speed= 62.000000[mm/min]
10 Maximum amount of Sheet Drawing Force=
   35.659695[kN]
11 Maximum amount of Punch Stroke= 100.000000[mm]
12 Initial Value Of Sheet Holder Force= 100.000000[kN]
13 True Name=Sheet Holder Force;Drawing Force;Time;…
14 Channel Name=Sheet Holder Force;Drawing
   Force;Time;…
15 Channel Unit=kN;kN;s;mm
16 100.000000; 0.470919; 0.000000; 0.000000
17 100.000000; 0.464896; 0.100000; 0.000000
18 …
```

To make things more complex there are values without a defined measurement unit.

```
// Erichsen simulation data
 1  62.000000
 2  35.659695
 3  100.000000
```

A single value can be extracted using the following regular expression:

$$Speed=\s*(?<value>[^[\n\r]*)$$

(=Take the value after the part “Speed=” until a “[” character or new line is reached and save it as a result with any number of white-space characters between the “=” and the value).

The Aramis experiment data is even more complex: the simulation data is composed from the values of the third and fourth (highlighted) column (line 8, 11). This approach completely differs from the previously mentioned two. All the data is delimiter-separated but the column headings use a different separator than the data itself which again is using a different separator, tab, form the other two discussed cases.

```
1   value        mean sampl| mean geometry       | sample    | sample # | section
2   major      | major     | major    | minor    | geometry  |          |
3   strain     | strain    | strain   | strain   |           |          |
4        0.449                                        150         1  section0
5   …
6        0.463     0.459                              150         1  section4
7   …
8        0.423     0.426       0.44       0.41        150         3  section4
9   …
10        0.71                                         24         3  section3
11        0.66     0.723      0.694     -0.355         24         3  section4
12  …
```

Besides the simple values the Aramis simulation data also contains calculated fields (ex. Number of fields) but this is not the manufacturer’s fault.

```
// Aramis simulation data
 1  MATERIAL_DEFINITION
 2  Material: DC04
 3  Strength: 0.85
 4  FORMING LIMIT
 5  Number of points: 6
 6  Semantics: 1
 7  -0.355     0.694
 8  0.00274    0.298
 9  0.0748     0.32
11  0.201      0.398
12  0.292      0.426
13  0.41       0.44
```

4. PROPOSED STANDARDS

We have seen that the data formats are different but we also observed that they contain common parts so we need to identify all the different requirements but use the best of the common parts. The definition of a complete material measuring system output format standard would take more research and would not fit into this paper; instead we are focusing on the already mentioned cases.

The following components need to be defined by the standard:

- Sections
- Number formats
- Parameters (Field name/value/measurement unit data)
- Tabular data
- Comments

A typical experiment file could look something like:

```
 1  [Experiment meta information]
 2  # Project-related comment
 3  Experiment run by = John Doe
 4  Date = 4/13/2012
 5  Experiment name = Elongation test
 6
 7  [Data]
 8  Field1 [kN] = 12,132.4456
 9  Field2 = 12,132.4456
10  Field3 = String containing \= sign
11
12  [Chart data]
13  "Change in width";Col2;
14  3.43206;   9.9659
15  9.9659;    0.0
16  13,000.34; 0.1
```

The format of the file is fairly straight forward:

Files are split into sections, each section starting with a section declaration. A section declaration starts with a ‘[’ and ends with a ‘]’. Sections occur on one line only. The name of the section can be anything because we do not see the possibility of agreeing with preset section names. Duplicate sections are not allowed.

Items in a section are known as parameters. Parameters have a typical key = value format. The ‘:’ character is also accepted in place of ‘=’ to separate keys and values in parameters, for example *var1: foo*. If the value has to contain a ‘=’ or ‘:’ character, they should be prefixed with ‘\’. Duplicate parameters are only allowed if they are in two different sections, thus they are local to sections; this configuration simply ignores the duplicates; if a section has a duplicate parameter only the first one is taken into consideration.

Table 1 - Common escape sequences

Sequence	Meaning
\\	\ (a single backslash, escaping the escape character)
\;	Semicolon
\#	Number sign
\=	Equal sign
\:	Colon

The measurement unit from the end of the value should be moved to the end of the key like in line 8. This would make the reading of the value easier and cleaner. Preferably the measurement unit should not be used. Base measurement units should be agreed upon and used while saving values.
Ex. if the agreed measurement unit for distance is *mm* then 1 micron can be expressed as 0.001 mm.

Lines starting with ‘#’ are assumed to be comments (line 2). The comment is usually not used in the simulation inputs but they are easily skipped if they are clearly marked.

Numbers should always be formatted in decimal format: the value 2×10^{-22} is perfectly legal and can be processed.

The localization of the file should always be English (UK) so the decimal separator is ‘.’ and the thousand separator is ‘,’. The English internationalization also assumes a 12/14/2012 13:24 date format. This of course cannot be asked because besides the application being in different languages and deployed on machines with different language options this would be in conflict with the other features of these applications that export data in other formats. The internationalization information could be added as a key-value pair into the first section.

Tabular data should have its own section (line 12), use semicolon (;) as separator and have the column names defined in the header (line 13). The semicolon is much more optimal to be used as separator than ‘,’ or tab because the tabular section almost only contains numbers; so the values should not be encapsulated in quotes because of thousand separators.

5. CONCLUSIONS

Defining the basic standards for the experiment results data generated by the material measuring systems will help development of a toolset that will allow users to define data extraction rules that can be applied to machines of any manufacturer. This would eliminate the skill gap stopping non-programmer users from using the tool.

The proposed solutions are not exhaustive at all; the standard proposals can be extended to handle special cases like missing values, empty values, continuing lines, binary data and many more.

ACKNOWLEDGMENTS

The paper has been elaborated as part of the projects: "PhD research in the field of engineering with the purpose of developing a science-based society – SIDOC”, Contract no. POSDRU/88/1.5/S/60078 and PCCE-100/2010.

REFERENCES

[1] J. Friedl, Mastering Regular Expressions - Powerful Techniques for Perl and Other Tools, O'Reilly Media, (1997)
[2] Zwick testXpert II software description (2010)
[3] V. Biro, D. Banabic, DaCoTraP – A web based platform for metal forming process chain, Computer Methods in Materials Science, 11 (2011), 265-270
[4] C. C. Ko, B. M. Chen, J. Chen, Creating Web-based Laboratories, Springer, Heidelberg-Berlin (2004).
[5] F. Davioli, N. Meyer, R. Pugliese, S. Zappatore, Remote Instrumentation and Virtual Laboratories, Springer, Heidelberg-Berlin (2010)
[6] Microsoft Patterns & Practices Team , Microsoft Application Architecture Guide, Microsoft Press (2009)
[7] S. D. Ritchie, Pro .Net Best Practices, Apress Media LLC, New York (2011)
[8] B. Bibeault, Y. Katz, jQuery in Action, Manning Publications, Shelter Island, NY (2010)