CCFE
CULHAM CENTRE FOR FUSION ENERGY

UK Atomic Energy Authority

UKAEA-CCFE-PR(20)97

F. Schoofs, M. Gorley, J. Shimwell

A human and machine readable materials data capture framework

Enquiries about copyright and reproduction should in the first instance be addressed to the UKAEA Publications Officer, Culham Science Centre, Building K1/0/83 Abingdon, Oxfordshire, OX14 3DB, UK. The United Kingdom Atomic Energy Authority is the copyright holder.

The contents of this document and all other UKAEA Preprints, Reports and Conference Papers are available to view online free at scientific-publications.ukaea.uk/

A human and machine readable materials data capture framework

F. Schoofs, M. Gorley, J. Shimwell

This is a preprint of a paper submitted for publication in arXiv.org

A human- and machine-readable materials data capture framework

F. Schoofs*, M. Gorley, J. Shimwell

United Kingdom Atomic Energy Authority, Culham Centre for Fusion Energy, Culham Science Centre, Abingdon, Oxon, OX14 3DB, UK

*Corresponding author: frank.schoofs@ukaea.uk

Abstract: Materials data are an essential aspect of any engineering design and simulation effort. Here we describe a method to capture and log materials data in a human- and machine-readable framework, based on the JSON format. The data is captured according to a set schema with set keywords, in an open, scalable format that can accept any data and with support in most programming languages.

Keywords: materials data; materials database; materials data capture

1. Introduction

Any engineering design or simulation requires the input of reliable materials data and material models, at all relevant conditions considered in the design. Typically, this tends to be different temperatures, but also includes environmental conditions (e.g. humidity) or operational conditions (e.g. neutron irradiation). In the nuclear fusion community, there has been and is significant effort to collate materials handbooks for relevant materials. The EU-DEMO program [1,2] is collating information on Eurofer97 [3], tungsten [4] and Cu-Cr-Zr [5]. There are two dominant sources of data: new experiments within the program and historic data in reports, publications, standards. Whenever "raw", tabulated datapoints are not available, they need to be extracted, e.g. graphically [6].

All the data need to be collated in a materials database, based on which a materials handbook with design curves or minimum property curves can be formulated. MatDB [7] is one such database framework which collates open access data submitted by experts. In its present state, it is heavily biased towards the mechanical properties of steels. EUROfusion are investigating MatDB and developing other long-term storage mechanisms but needed workable interim solutions for progression of the works.

In this paper we describe a framework that allows flexible, expandable materials property data capture in a way that is both human- and machine-readable and can handle any type of property. It is not intended to capture full experimental datasets, although it theoretically could be extended to do so.

2. General framework structure

We adopt a hierarchical data structure, as schematically shown in Figure 1, using the JSON format [8], which has the benefit of being platform agnostic with native readers/writers and validator tools available in most programming languages, including but not limited to C++, Python and Javascript. It is also possible to define a schema to enforce consistency. Each file represents a unique experimental condition, which can be given a unique identifier (UID) number. This file captures all the data collected on a specific sample with a set of fixed conditions, which can include composition, heat treatment or environmental conditions (e.g. irradiation dose & temperature, gas test atmosphere). This implies that a single bibliographic reference can yield multiple UID files.

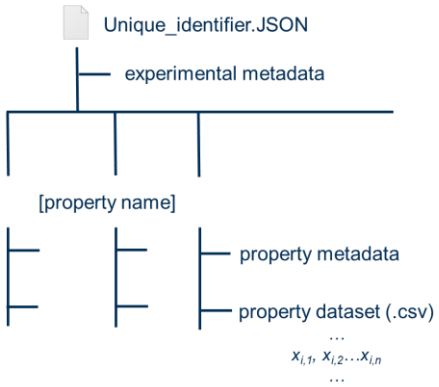

Figure 1: Hierarchical data structure framework.

Each UID file contains an experimental metadata block ("`meta`") which describes the conditions of the sample (see section 3) and then a block for each property measured on that sample (see section 4). Each property block contains a metadata block ("`property_meta`"), which describes how to interpret the data block, and the actual data block ("`property_data`"). Optionally, a model description block can be added ("`property_model`"), if there is a fitted model available (see section 5).

3. Reference metadata structure

The experimental metadata file should contain sufficient information to uniquely identify and categorise the sample, such as a bibliographic reference (e.g. DOI), the sample name or ID given in that reference and the composition of that sample. They are written as "`key:value`" on each line, as per the JSON format. All keys are case sensitive and need to be consistent across all UIDs.

Examples of keys include:

- "`DOI`": the digital object identifier of the reference (article or dataset)
- "`ref_str`": a string descriptor of the reference
- "`sampleID`": the descriptor of this sample as used in reference, e.g. "sample A"
- "`condition`" or "`heat_treatment`": important for many practical metals; the value needs to be consistent for all alloys
- "`composition`": the composition of the alloy or material under consideration, either as a list of elemental composition or as a standard designation (e.g. Al6061). Specific elements can also be singled out on separate lines, e.g. "`wtCr`"
- "`grain_size_um`": average grain size of the material in mm

Below is an example of a file created for collecting CuCrZr property handbook data, referring to sample data in Ref. [9]:

```
    "meta": {
            "doi": "10.1007/s10765-010-0857-y",
            "ref_str": "G. Pintsuk et al. Int. J. Thermophys. 31 (2010) 2147",
            "sampleID": "FZJ",
            "condition": "SAA",
            "composition": "0.8Cr0.08ZrbalCu",
            "wtZr": 0.08,
            "wtCr": 0.8
        }
```

4. Experimental property data structure

In order to make automated machine access easier, the names of the property blocks need to be consistent across all UIDs. Continuing with our example for CuCrZr, these block names are for example: *"*`CTE`*"*, *"*`density`*"*, *"*`heat_capacity`*"*, *"*`thermal_conductivity`*"*. Each property block contains the datapoints in a "`property_data`" block and a metadata block allowing the correct interpretation of the data block. Again, all keys need to be consistent across UIDs and properties. For a typical (*x,y*)-type dataset, the following would be a minimum:

- "`x_name`": descriptor for the *x*-axis data
- "`x_unit`": unit of the *x*-axis data
- "`x_err_unit`": indicator on the units of the *x*-error bars, can be "`pct`" or "`axis`"
- "`y_name`": descriptor for the *y*-axis data
- "`y_unit`": unit of the *y*-axis data
- "`y_err`": indicator on the units of the *y*-error bars, can be "`pct`" or "`axis`"

An example property block is shown below:

```
"thermal_conductivity":{

    "property_meta":{

                "x_name":"Temperature",

                "x_unit":"C",

                "y_name":"Thermal Conductivity",

                "y_unit":"W/m*K",

                "y_err_unit":"axis"

            },

    "property_data":{

    "x":[20.0, 100e, 200, 300, 400, 500],

    "y":[358.6, 369.9, 372.7, 373.4, 373.5, 356.4],

    "y_err":[12.10, 25.37, 25.26, 22.51, 21.20, 25.02]

    }

}
```

The `x_err` key is omitted, because there is no data available in this case.

5. Capturing property models

In addition to these essential blocks, a "model" block can be included for each property (e.g. "`property_model`"). This should include a model name, who it was fitted by (e.g. the reference itself), the units for input and output, the parameter validity range (e.g. temperature range of the model) and then the parameters themselves as a list.

For example, the fit to hydrogen diffusivity data in Ref. [10] can be captured as:

```
"property_model":{

    "model_name":"arrhenius-eV-kT",
    "fitted_by":"reference",
    "x_unit":"K",
    "y_unit":"m2/s",
    "x_lim":[573,1023],
    "p":[2.2e-4,0.84]

}
```

6. Conclusion

Figure 2 shows a screenshot of an example UID file based on data for a single sample and heat treatment condition in Ref. [9]. We have also created a basic interface in JavaScript/HTML to generate these files [11]. In summary, we have shown a flexible, scalable framework to capture materials property data, in a format that is both human- and machine-readable.


```
{
    "meta": {
        "doi": "10.1007/s10765-010-0857-y",
        "ref_str": "G. Pintsuk et al. Int. J. Thermophys. 31 (2010) 2147",
        "sampleID": "FZJ",
        "condition": "SAA",
        "composition": "0.8Cr0.08ZrbalCu",
        "wtZr": 0.08,
        "wtCr": 0.8
    },
    "density": {
        "property_meta":{
            "x_name":"Temperature",
            "x_unit":"C",
            "y_name":"Density",
            "y_unit":"g/cm3"
        },
        "property_data":{
            "x": [25.0],
            "y": [8.89]
        }
    },
    "heat_capacity": {
        "property_meta":{
            "x_name":"Temperature",
            "x_unit":"C",
            "y_name":"Heat capacity",
            "y_unit":"J/kg*K"
        },
        "property_data":{
            "x": [50, 100, 150, 200, 250, 300, 350, 400, 450, 500],
            "y": [0.34822, 0.39234, 0.40178, 0.40898, 0.4116, 0.41636, 0.41591, 0.42154, 0.42557, 0.42951]
        },
        "property_model": {
            "model_name": "poly",
            "fitted_by": "user",
            "x_unit":"C",
            "y_unit":"J/kg*K",
            "x_lim": [50,500],
            "p": [-4.66e-7,3.88e-4,3.45e-1]
        }
    },
    "thermal_conductivity": {
        "property_meta":{
            "x_name":"Temperature",
            "x_unit":"C",
            "y_name":"Thermal conductivity",
            "y_unit":"W/m*K",
            "y_err_uinit":"abs"
        },
        "property_data":{
            "x": [20, 100, 200, 300, 400, 500],
            "y": [358.60, 369.90, 372.70, 373.39, 373.50, 356.37],
            "y_err": [12.10, 25.37, 25.26, 22.51, 21.20, 25.02]
        }
    },
    "thermal_diffusivity": {
        "property_meta":{
            "x_name":"Temperature",
            "x_unit":"C",
            "y_name":"Thermal diffusivity",
            "y_unit":"mm2/s",
            "y_err_unit":"abs"
        },
        "property_data":{
            "x": [20, 100, 200, 300, 400, 500],
            "y": [107.664, 107.668, 105.371, 103.561, 102.588, 97.827],
            "y_err": [3.633, 7.383, 7.141,6.244,5.823,6.869]
        }
    }
}
```


Figure 2: Example UID file for a sample from Ref. [9]

Acknowledgements

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 and 2019-2020 under grant agreement No 633053 and from the RCUK Energy Programme [grant number EP/T012250/1]. To obtain further information on the data and models underlying this paper please contact PublicationsManager@ccfe.ac.uk. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] G. Federici, R. Kemp, D. Ward, C. Bachmann, T. Franke, S. Gonzalez, C. Lowry, M. Gadomska, J. Harman, B. Meszaros, C. Morlock, F. Romanelli, R. Wenninger, Overview of EU DEMO design and R&D activities, Fusion Eng. Des. 89 (2014) 882–889. https://doi.org/10.1016/j.fusengdes.2014.01.070.

[2] M. Gorley, E. Diegele, E. Gaganidze, F. Gillemot, G. Pintsuk, F. Schoofs, I. Szenthe, The EUROfusion materials property handbook for DEMO in-vessel components—Status and the challenge to improve confidence level for engineering data, Fusion Eng. Des. 158 (2020) 111668. https://doi.org/10.1016/j.fusengdes.2020.111668.

[3] E. Gaganidze, F. Gillemot, I. Szenthe, M. Gorley, M. Rieth, E. Diegele, Development of EUROFER97 database and material property handbook, Fusion Eng. Des. 135 (2018) 9–14. https://doi.org/10.1016/j.fusengdes.2018.06.027.

[4] F. Schoofs, M. Gorley, A route to standardised high heat flux testing: An example for tungsten, Fusion Eng. Des. 139 (2019) 132–136. https://doi.org/10.1016/J.FUSENGDES.2018.12.060.

[5] K. Zhang, E. Gaganidze, M. Gorley, Development of the material property handbook and database of CuCrZr, Fusion Eng. Des. 144 (2019) 148–153. https://doi.org/10.1016/j.fusengdes.2019.04.094.

[6] A. Rohatgi, WebPlotDigitizer, (n.d.). https://github.com/ankitrohatgi/WebPlotDigitizer.

[7] MatDB, (n.d.). https://odin.jrc.ec.europa.eu/alcor/Main.jsp.

[8] International ECMA, The JavaScript Object Notation (JSON) Data Interchange Format (ECMA-404), 2017. https://doi.org/10.17487/rfc7159.

[9] G. Pintsuk, J. Blumm, R.C. Hula, T. Koppitz, S. Lindig, D. Pitzer, P. Rohde, M. Schoderböck, T. Schubert, F. Tietz, O. Wouters, Interlaboratory Test on Thermophysical Properties of the ITER Grade Heat Sink Material Copper – Chromium – Zirconium, Int. J. Thermophys. 31 (2010) 2147–2158. https://doi.org/10.1007/s10765-010-0857-y.

[10] T.V. Kulsartov, Y.V. Chikhray, I.L. Tazhibaeva, V.P. Shestakov, Investigation of Hydrogen Permeability through Copper Alloy CuCr1Zr0.1 and Duplex Structure Be-Cu, Fusion Technol. 34 (2018) 919–923. https://doi.org/10.13182/fst98-a11963730.

[11] F. Schoofs, Materials Property Data JSON generator, (2020). https://doi.org/10.6084/m9.figshare.12186774.