A web-based support for the management and evaluation of measurement data from Stress-Strain and Continuous-Cooling-Transformation experiments

Ronny Kramer
Department of Computer Science
Chemnitz University of Technology
09107 Chemnitz, Germany
kramr[at]cs.tu-chemnitz.de

Gudula Rünger
Department of Computer Science
Chemnitz University of Technology
09107 Chemnitz, Germany
ruenger[at]cs.tu-chemnitz.de

To cite this version:

Kramer R., Rünger G.: A Web-Based Support for the Management and Evaluation of Measurement Data from Stress-Strain and Continuous-Cooling-Transformation Experiments. In: Information Systems Architecture and Technology: Proceedings of 40th Anniversary International Conference on Information Systems Architecture and Technology - ISAT 2019. Advances in Intelligent Systems and Computing, 1050 (2020), ISBN: 978-3-030-30439-3, DOI: 10.1007/978-3-030-30440-9_15, Springer, Cham

Abstract

Mechanical engineers are often faced with the challenge to keep track of the results of experiments done within their own institutions or other research groups. The lack of this knowledge may cause a potential waste of resources and time since experiments might be repeated if the existence of previous experiments is unknown or the data cannot be reused. The main problem with managing scientific data is the huge amount of experiments, the lack of public access to the results, and often missing meta-data which would be needed to compare the results. This article considers experiments and data resulting from experiments of Stress-Strain and Continuous-Cooling-Transformation experiments. It is shown how a web-driven data management process starting from the planning of experiments through to the evaluation of the measurement data could be supported by an appropriate work-flow to make sure that all meta-data is preserved to find and reuse the results.

data management, data evaluation, web-based, material science

1 Introduction

Public databases providing knowledge for scientists in different fields exist for decades. An example for material science are crystallography databases which exist for over 70 Years [10]. These databases have been designed independently for the specific use cases and, thus, they are heterogeneous in the sense that they store their data in different formats or use different terminology, so that a combination of their datasets is not trivial. To accommodate this issue, new ontologies, such as MatOnto[6] have been created to support the storage of knowledge in a structured way using a defined terminology. The definition and acceptance of ontologies like MatOnto combined with new Internet Standards, such as the Resource Description Framework (RDF)[8] and the Web Ontology Language (OWL)[4], can be explored by search interfaces, such as MatSeek[7], which allow ontology-based searches in multiple databases. These inventions have improved the cross usability of material science databases as well as the overall search process. The information gained by combining public databases is often not enough for data science processes in material science, as they normally only provide results and do not include measurement data with extensive knowledge about the provenance. The measurement data in conjunction with the provenance is required to gain new information, e.g. by the detection of similarities or differences in the datasets, which would be needed to make predictions about material compositions.

The goal of this article is to propose an approach which improves the available measurement data by including enough provenance necessary for data science analysis while keeping the overhead for gathering the data as low as possible. A thorough digital archiving including the collection of as much provenance data as possible can become a time-consuming task so that there is a need for a sustainable support accomplishing this task implicitly while performing the current research experiment. The approach proposed in this article is to provide a software system that is used by a scientist during the research work and that is able to collect the provenance data automatically. The challenge for such a software system is to define a work-flow that is convincing enough for the scientists so that they adapt their habits and take the extra burden to use a new support system instead of their individual work-flow they are used to. This can be achieved by providing assistance or even automation for repeating and time-consuming tasks. In the context of material science, those tasks are: The planning of experiments, the organization, the scheduling of the testing machine usage, the preparation of specimens or the actual execution of experiments and their evaluation. The execution of all of these tasks can be assisted or even automated except the preparation of specimens and the actual execution of experiments, as they still require manual work to be performed.

The contribution of this article includes the design and implementation of a data management and evaluation web-based software platform for a real world application in mechanical engineering, which has been developed in cooperation with the application scientists, and a proof-of-concept for a data management and evaluation web-based software platform to be used in other applied sciences.

The rest of this article describes related work in Section 2 and introduces the software system and work-flow concept in Section 3. Section 4 presents the application from material science and Section 5 provides concluding remarks.

2 Related Work

The book "Building the Data Warehouse"[2] from 1992 already presents some general patterns for the design of a system called data warehouse which is suitable to extract information from legacy systems and makes them usable for further analysis. The idea of data warehouses was applied to material databases by Y. Li, for the usage in large data collections with the goal to identify similarities and handle differences [5]. The problem with the data warehouses as described in these books is that they are designed to be compatible to specific databases, which makes the adaption to new systems more complicated [7]. The new idea in this article is to take the idea of data warehouses and place them in the center of a software system and build a work-flow about data generation and computer aided evaluation around it. While the disadvantage of this approach is, that the usage of existing data is still complicated, the advantage is that newly collected data can be stored with the focus on data science. This focus not only improves the applicability of data but also increases the portability as the storage can be built to support established ontologies.

A computer aided evaluation is already established in material science by software systems typically provided by the manufactures of the computer aided test machines used to perform the experiment. Since the main functionality of these software systems is to control the corresponding machine, the provided data evaluation is optimized for the specific use case. As a result of this optimization the software system is able to perform almost any computation that the user of this machine might want to perform, but its capabilities are limited to the specific use case.

In the area of mechanical material testing, two companies that provide their own software solution are ZwickRoell with the software testXpert[15] and MetLogix with L3[12]. Besides the Stress-Strain evaluation via tensile testing, they also support three-point flexural, peel, holding, creep and creep rupture testing. They do not support any evaluations for which the data required cannot be determined using their machines. Thus, a thermal analysis is not possible. In the area of thermal analysis, two companies which provide their own software solution are NETZSCH with Proteus[14], and LINSEIS thermo analytical software[13]. As specialized software solutions they are able to evaluate additional experiments to CCT, such as Continuous-Heating-Transformation (CHT) and Time-Temperature-Transformation (TTT). These software tools allow the export of the measurement data and results in various file formats, such as ASCII CSV or Microsoft Excel files. The usage of these evaluation software tools would be an applicable solution to reduce the complexity of a data management system. However, this is not viable for the experiments considered here, since parts of the provenance meta-data is lost during the post processing and the evaluation in external tools. Thus, none of the existing software tools are capable to provide a consistent software support as needed for the specific experiments considered in this article, which leads to the software solution.

3 Work-flow for implicit meta-data collection

Data collection is an essential part for scientific experiments, but is very time-consuming. To support the scientist, we propose an implicit meta-data collection which accompanies all phases of the experiment and collects data during the planning phase, execution phase and evaluation phase of an experiment. The implicit data collection is modelled as a work-flow containing all steps in the experiment, including manual work in the lab as well as software-supported steps. Figure 1 illustrates such a work-flow. To guide the scientist through the work-flow, an appropriate software system is to be provided. From the point of view of the scientist, the advantage of the software system is a support in the background during the planning, execution and evaluation phases of an experiment.

The work-flow proposed contains steps consisting of multiple actions. The steps in Figure 1 are presented in two different colors, white and grey, depending on their connection to the software system. The steps presented in white boxes can be supported or even automatized by the software system, while a grey box denotes actions that belong to the experiment and require manual labor executed outside of the software system. However, as the actions executed during these manual labor steps are intended to follow the created plan, the provenance is still guaranteed. Any incident needs to be logged to keep the provenance consistent.

The **planning phase** of an experiment is based on four pillars:

1. The selection of the context for the experiment planned. This decision influences the templates that are going to be used for all other steps.
2. The preparation planning for the specimen based on materials and preparation steps.
3. The selection of machines to be used from a list of compatible machines based on pillar 1 and 2. This connects the final results to the available information about the machines.
4. The processing of the data, such as predictions or suggestions based on larger data sets and data analysis.

Based on those four pillars, the experimental plan is created containing a check-list, a task plan for the technical stuff, a material list and a time schedule.

As already stated, the **execution phase** is contained in the work-flow but only loosely connected to the software system. Thus, the interaction with the software system is limited to the logging of unexpected incidents.

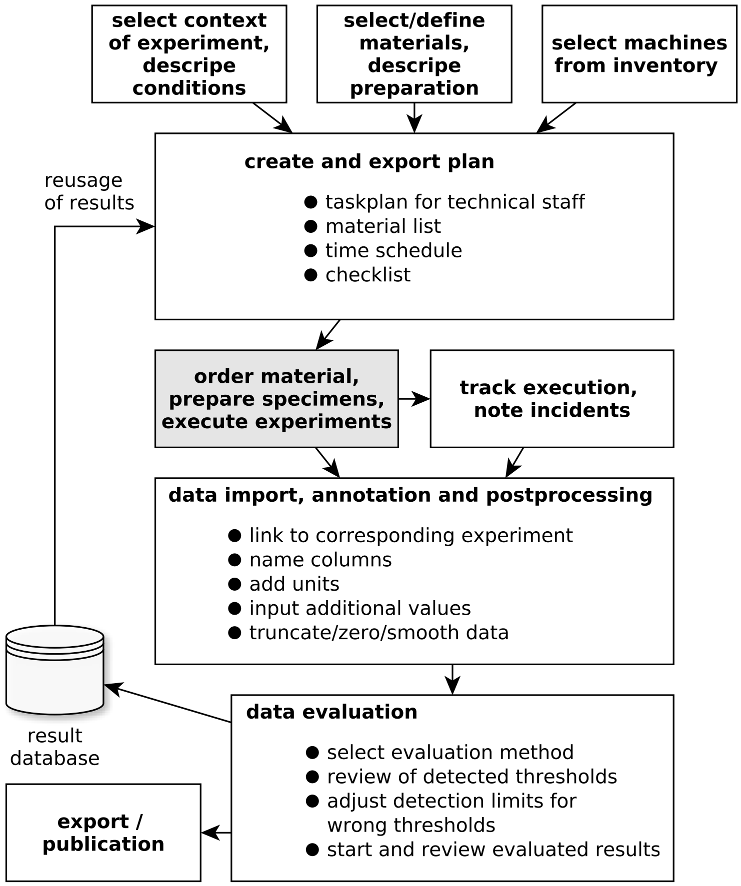

Figure 1: Work-flow to collect meta-data from an experiment, starting with the planning of an experiment up to the evaluation of the experiment. A mechanism for the reuse of knowledge gained by previous experiments is included.

The **evaluation phase** consists of two major parts, which are the preparation of the data and the final evaluation of the data. For the preparation, the raw measurement data gained from the experiments are imported, annotated and post processed so that data are in a form that is compatible to the software system. The annotation is done by linking the data sets to the experiment and by mapping the data columns to the contexts requirements. The physical units of the data columns are automatically assigned according to the mapping, but can be changed by the scientist. By a post processing of the imported data with methods, such as trimming, smoothing and zeroing, the compatibility to the software systems evaluation can be improved. For the final evaluation, the data is used to gain knowledge about the properties of the material, such as the stiffness, flexibility or the resistance to high temperatures. The material properties are presented in an appropriate form for exploration or exportation and are stored in the database for future data-mining. An example for the final evaluation based on Stress-Strain and Continuous-Cooling-Transformation experiments is given in the next section.

4 Stress-Strain and Continuous-Cooling-Transformation

The implicit collection of meta-data of the Stress-Strain, see Subsection 4.1, and Continuous-Cooling-Transformation, see Subsection 4.2, evaluation is captured in a work-flow as described in general in the previous section. To realize this work-flow, the software system provides templates for all steps and implements the specific equations required to calculate the material properties. As the evaluation of the experiments can be a very time consuming task, the automated evaluation is the main advantage of the software-system provided to the application scientist in material science. The automated evaluation is crucial to provide faster evaluations and visual support and, thus, supports to perform more experiments in shorter time frames supporting a faster knowledge acquisition.

4.1 Stress-Strain

The goal of Stress-Strain data is the calculation of material properties that, among others, describe how the specimen and therefore the materials or joins react when forces are applied [9]. The data is collected during tension testing where a force is applied with a given speed. A standardized specimen is strained until its failure while the change in length and the required force is being recorded [11].

The evaluation of Stress-Strain Data is defined in the DIN 6892-1 [11]. The base value for all calculations is the young modulus E in Pascal that represents the elastic value of the specimen and is calculated as $E = \frac{\sigma}{\epsilon}$ where σ is the stress and ϵ the proportional deformation. For metals and some other materials, the young modulus can be described by a law of physics, called Hooke's law. According to this law, the young modulus visualizes a linear slope at the beginning of the data region where $E = \frac{\sigma}{\epsilon} = const$ applies; this is called the Hookean slope [3]. Algorithm 1 is used to detect the young modulus and assumes that Hooke's law is valid for the specimen considered. In the ideal case, the Hookean slope is represented by a data range that starts with the first datum and ends at the beginning of a concave curve at the datum with maximum stress. However, real world data often shows a convex curve which changes into this concave curve. In such cases, an approximation is used to determine the inflection point which has the biggest slope [11]. As the measurement data may not be smoothed, the software system needs to cope with this problem described by calculating the average slope over a data window.

In all cases tested for this article, the automatic detection of the young modulus by the algorithm has been proven to be robust. Due to the error-prone nature of real world data, there is a chance that the algorithm makes a wrong

Algorithm 1 Detection of the young modulus

```
procedure findYoungModulus
    Let C be a list of stress and proportional deformation pairs
     with m index of max stress and w as window size;
    for (i=w, ... ,m-w) do
        calculate average slope over C(i-w, ... , i+w);
    end for
    find maximum of the calculated slopes;
end procedure
```

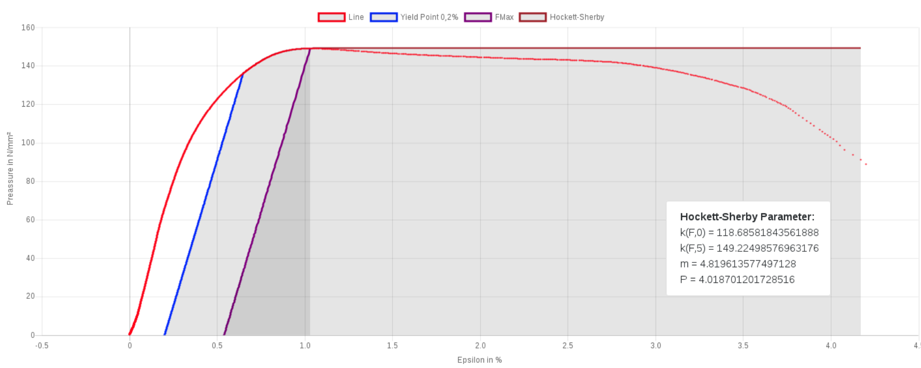

Figure 2: Example diagram of the Stress-Strain evaluation generated by the software system.

detection. Thus, the scientist may visually review the final result diagrams and may decide whether the algorithm has found a plausible young modulus or an adjustment of detection parameters is required in some circumstances.

The results of Stress-Strain experiments consist of three different components; an example diagram generated by the software-system is given in Figure 2. The first component is a plot of the Stress-Strain curve to provide a visualization. The second component consists of the evaluated material properties, which are split up again into two groups. There are values that can be determined by searching local extremes on the curve such as the point at which the maximum force was applied. Other values require the construction of the Hookean slope in predefined points, such as the yield point at 0.2% proportional deformation, and a subsequent calculation of their intersection with the curve or the zero-crossing. The third component provides predictions, such as the Hockett-Sherby equation [1], which are calculated by curve fitting.

4.2 Continuous-Cooling-Transformation

The goal of the Continuous-Cooling-Transformation (CCT) is to determine the phase transformation temperatures during the cool-down process. Phase transformations are structural changes within the material considered. They are

Algorithm 2 Find all linear sections with a temperature range of over $k\ Kelvin$

```
procedure findLinearSections
    Let C with |C|=m be a list of length variation over
    temperature;
    Let k be the minimal temperature range in Kelvin and w the
    window size;
    Let T be an empty list;
    Let trange be a tuple initialized with the temperature of
    the first element of C;

    for (i=w,... ,m -w) do
        calculate average slope over C(i-w,... , i+w);
        calculate average temperature over C(i-w,... , i+w);
        if (slope change > threshold or last window) then
            if (trange > k) then
                Add trange to T;
            end if
            set low temperature of trange;
        end if
        set high temperature of trange;
    end for
end procedure
```

Algorithm 3 Filter linear sections by largest curve coverage

```
procedure filterLinearSections
    Let T with |T|=m be a list of linear section temperature
    tuples;
    for T(i=1,.... ,m) do
        filter adjacent pair (i, i+1) of linear sections with
    largest range between temperatures;
    end for
    transform pair into linear functions t_h and t_l;
end procedure
```

visualized as a change of the length of the specimen that is not related to the linear shrinking based on the expansion coefficient. The temperatures of the phase transformation represent an important part of CCT diagrams, which are required to judge the compatibility to a specific use case [9]. While the recognition of the phase transformation within a diagram is trivial for a human with experience in this domain, the adaptation of this experience in a software system is a challenge. In the following, it will be discussed how it is possible to predict the theoretical shrink process by expanding the linear sections close to the phase transition. The extraction of the transformation temperatures is achieved by comparing the theoretical shrink with the real shrink.

The Algorithms 2 and 3 used for the detection of the phase transformation, are based on the assumption that the expansion and shrinking process

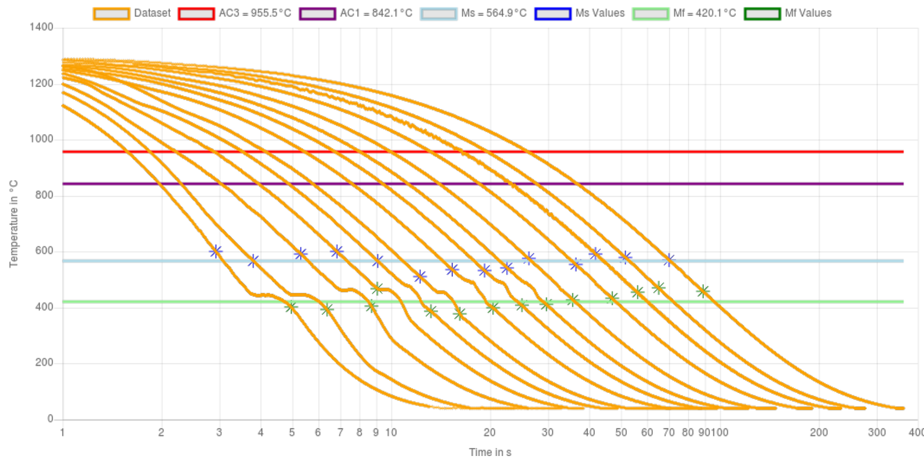

Figure 3: Example diagram of the Continuous-Cooling-Transformation evaluation generated by the software system.

during heat-up and cool-down is only influenced by the expansion coefficient. Algorithm 2 works by searching all linear sections of T, where the temperature range for the slope calculation is at least k $Kelvin$. The curve data is expected to consist of a heat-up and a cool-down stage, including a plateau p where the temperature has a peak for a short amount of time. By splitting the curve data C at the plateau p, Algorithm 2 can be used to find the linear sections for each stage independently. Algorithm 3 filters the linear sections close to the phase transformation by the largest curve coverage and converts them into the linear functions t_h and t_l. The percentage of transformation over the temperature t can be evaluated by calculating $\frac{|t_h(t)-C(t)|}{|t_h(t)-t_l(t)|}$ $\forall\, t \in Inputdata$. Figure 3 shows the evaluated results in form of a diagram created by the software system.

A user survey comparing the automated evaluation of these exemplary experiments to the traditional evaluation method using multiple disjoint software-tools showed a time saving potential. The scientists who have tested this software-system reported an individual time reduction from a couple of hours down to a few minutes per experiment, which is considered to be a success.

5 Conclusion

In this article, we have proposed a software system which is designed to support experiments in material sciences with the specific functionality to collect a large variety of experimental data and meta-data with the aim to provide the possibility for successful data-mining and data analysis processes in the future. The core of the software system proposed is a specific work-flow supporting this functionality. The goal is to provide a support of the scientists in material science which serves their needs by adding benefits that ease their experimental work evaluation. Our approach is to build a software system which makes it possible for the scientists to mainly concentrate on the experiments while the software and the data gathering is executed in the background and makes it possible to come back to the software system when needed or the next step is to be done. The software implementation is in the state of a proof-of-concept, which has successfully been applied to a specific experiment described in this article.

6 Acknowledgements

This work is supported by the Welding Engineering group of the TU Chemnitz and by the Federal Cluster of Excellence EXC 1075 *MERGE Technologies for Multifunctional Lightweight Structures* financed by the German Research Foundation (DFG).

References

[1] J.E. Hockett and O.D. Sherby. “Large strain deformation of polycrystalline metals at low homologous temperatures”. In: *Journal of the Mechanics and Physics of Solids* 23.2 (1975), pp. 87–98. ISSN: 0022-5096. DOI: 10.1016/0022-5096(75)90018-6.

[2] William H. Inmon. *Building the Data Warehouse*. New York, NY, USA: John Wiley & Sons, Inc., 1992. ISBN: 0471569607.

[3] A C Ugural and Saul K. Fenster. *Advanced Strength and Applied Elasticity, 4th Edition*. Prentice Hall, 2003.

[4] Sean Bechhofer et al. “OWL Web Ontology Language Reference”. In: (Feb. 2004).

[5] Y. Li. “Building The Data Warehouse For Materials Selection in Mechanical Design”. In: *Advanced Engineering Materials* 6.1-2 (2004), pp. 92–95. DOI: 10.1002/adem.200300522.

[6] Kwok Cheung, John Drennan, and Jane Hunter. “Towards an Ontology for Data-Driven Discovery of New Materials”. In: (Jan. 2008).

[7] Kwok Cheung, Jane Hunter, and John Drennan. “MatSeek: An Ontology-Based Federated Search Interface for Materials Scientists”. In: *IEEE Intelligent Systems* 24 (Jan. 2009). DOI: 10.1109/MIS.2009.13.

[8] David Wood, Markus Lanthaler, and Richard Cyganiak. *RDF 1.1 Concepts and Abstract Syntax*. W3C Recommendation. W3C, Feb. 2014. URL: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

[9] Frank Hahn. *Werkstofftechnik-Praktikum : Werkstoffe prüfen und verstehen; mit ... zahlreichen Tabellen*. München: Fachbuchverl. Leipzig im Carl Hanser Verl., 2015, 260 S. ISBN: 9783446432581.

[10] Ian Bruno et al. “Crystallography and Databases”. In: *Data Science Journal* 16 (Aug. 2017). DOI: 10.5334/dsj-2017-038.

[11] ISO. *Metallic materials - Tensile testing - Part 1: Method of test at room temperature*. Beuth Verlag, Berlin, 2017.

[12] *L3 - Kraft Prüfsoftware für die Materialprüfung*. Accessed: 2019-05-09. URL: http://www.metlogix.de/pf/l3.

[13] *LINSEIS Thermoanalyse Software*. Accessed: 2019-05-09. URL: https://www.linseis.com/produkte/dilatometer/l78-rita/#software.

[14] *NETZSCH Proteus® Software für die Thermische Analyse*. Accessed: 2019-05-09. URL: https://www.netzsch-thermal-analysis.com/de/produkte-loesungen/software/proteus/.

[15] *Prüfsoftware testXpert III*. Accessed: 2019-05-09. URL: https://www.zwickroell.com/de-de/pruefsoftware/testxpert-iii.