

Cyberinfrastructure for Integrated Computational Material Engineering

Tomasz Haupt

Mississippi State University; CAVS, Mississippi State, MS 39762, USA

Keywords: cyberinfrastructure, repository of data, material models, computational tools, web portals

Abstract

The goal of the cyberinfrastructure for ICME is to exploit the recent transformative research in material science involving multiscale physics-based predictive modeling, multiscale experiments and design. More specifically, the creation of the cyberinfrastructure will result in the development of a “community of practice” portal that allows development and integration of multiscale physics-based materials models for selected properties and processes. This paper demonstrates the use of the modern information infrastructure based on AJAX-based rich user interfaces, Service Oriented Architecture (SOA), Web Services and Grid computing streamlining of the process of gathering experimental results, and deriving the material properties (using online model calibration tools) for a particular material model (e.g., Damage Fit or Multistage Fatigue Fit) and employing the material model in finite element analysis in the process of building validated metamodels and design optimizations to support the Magnesium Front-End Three Nation Pilot Program.

Introduction

The objective of this effort is to design and develop a cyberinfrastructure to exploit the recent transformative research in material science involving multiscale physics-based predictive modeling, multiscale experiments, and design. More specifically, the creation of the cyberinfrastructure will result in the development of a “community of practice” portal that allows development and integration of multiscale physics-based materials models for selected properties and processes, in the context of the United States Automotive Materials Partnership (USAMP) three-nation Magnesium Front-End Research and Development pilot project (MFERD), in particular task 1.9: Cyberinfrastructure for Integrated Computational Material Engineering (ICME).

The development of the cyberinfrastructure for the computational material engineering will leverage tools, technologies, and software approaches developed by other large-scale scientific cyberinfrastructure projects supporting researchers and engineers in domains such as astronomy, medicine, biology, geophysics, earthquake engineering, and many more. Much of the foundation of the infrastructure is common across domains (security, grid integration, etc) and can serve to bootstrap this project to delivering an initial working solution in a relatively short time frame. Some initial customization was needed to support the unique aspects of this project. After the initial system had been developed, many of the advanced capabilities of the system required additional software development and customization and entirely new development to support additional features. Hence, we are proceeding with a two-stage development model: phase one was to leverage existing software and tools to bring up a working software infrastructure, and phase two was to evaluate and further develop missing or insufficient software, as needed, to support the project requirements.

The actual development and deployment of the cyberinfrastructure was driven by the requirements and early feedback generated by the engineering community, comprised of Mississippi State University (MSST) researchers performing other tasks of this project, as well as researchers and engineers involved in the MFERD/ICME project. Frequent interactions between the community (as defined above) and researchers developing the cyberinfrastructure with resulted in modifications of the initial plans, such as adding new elements to the cyberinfrastructure and changing the priorities of the development and deployment of the cyberinfrastructure components. The changes reflect a close collaboration between tasks and the desire to maximize the benefits of the cyberinfrastructure to the end user.

Community of Practice Web Portal

The interactions between researchers representing Material Sciences and Computer Science and Engineering resulted with charting the cyberinfrastructure effort as “the development of a community of practice web portal that allows development and integration of multiscale physics-based materials models for selected properties and processes.” The proposed architecture of the portal is shown in figure 1.

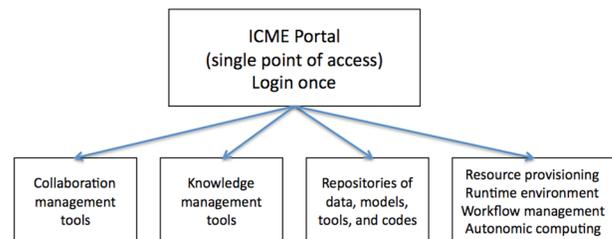


Figure 1: Architecture of the ICME community of practice portal

The collaboration management tools are a web space to gather project-related documents, such as tasks descriptions, annual reports, and presentations. In addition, this component supports broadcasting information about events to all members of the collaboration. We have chosen the Microsoft SharePoint server to implement the collaboration management. The server is online, and is available to registered members of the MFERD project.

Knowledge management can be achieved by applying “architecture of participation” advocated and implemented by Web 2.0 concepts and technologies. The most important aspect of Web 2.0 is a focus on user-generated content, as opposed to centrally managed information. Tools like wiki, in conjunction with content ratings, lead to creation of a collective (read: peer-reviewed) knowledge that is always up-to-date and with spontaneously evolving structure reflecting the current state of the art. Commercial and community-based implementation of this concept, including Wikipedia, Amazon.com, and Facebook.com prove that this approach is very effective, and the process of knowledge accumulation is convergent.

Most of the Web 2.0 technologies are Open Source and can be relatively easily integrated with our portal. We are in an early stage of this activity trying to accumulate enough information to launch “ICME Wikipedia” that would be interesting enough to attract the community to participate.

Currently, we are concentrating on the development of the repositories to collect and disseminate through the Web: data, models, and codes pertaining to computational material engineering. The remaining chapters of this paper provide the details of the design and implementation of these repositories.

Finally, the cyberinfrastructure should provide the runtime environment for running complex multistep design optimizations that involve running multiscale simulations, as schematically shown in Figure A1 in the Appendix. The technical challenge of running these complex workflows comprised of high-performance applications running in a distributed, heterogeneous environment that are necessary to fulfill the promise of ICME are intimidating for a material scientist. An intelligent, autonomic, that is, self-protecting, self-healing, self-configuring, and self-optimizing runtime environment that hides the IT complexity from the user is necessary. Unlike the Web 2.0 environment needed to implementing the other components of the portal, the autonomous computing remains a hot research and development topic, and we are addressing it as a separate thread and it will be described elsewhere.

The repository of material properties

The repository of material properties (RMP) integrates three independent web applications:

- Repository of experimental data
- Repository of material constants
- Online model calibration tools

Experimental data

The experimental data come from physical measurements of materials’ properties. Currently, the repository supports stress-strain data, images of the microstructure, and strain-life (fatigue) data. The repository of experimental data is a database of the results of measurements (often requiring a transformation of raw data, such as deriving true stress-strain from force-displacement), or data taken from the literature. The user can upload the data to the repository, search for a particular data set, and retrieve the data for further analysis – typically to derive material constants.

Material constants

For the purpose of numerical simulations, a material is represented by mathematical models, such as Damage Model, that provides a prediction of the material behavior subjected to a certain conditions. The models are parameterized by a model-specific number of constants, referred to as material constants. The repository of material constants is a database of these constants. The constants are derived from the experimental data. The user can upload the constants to the repository, search for the constants of a particular model of a particular material, and retrieve the constants for further analysis – typically to use them in numerical simulations, such as finite element analysis using Abacus, LS-Dyna, or other software.

Model calibration

The model calibration is the process of deriving the material constants from the experimental data, usually by performing a fit of a model-specific function(s) to the experimental data. RMP provides online tools to perform the models’ calibrations.

Integration

Although each of the aforementioned web applications can be used independently, the advantage of RMP is that it integrates all three application into one (c.f. Figure2), thus allowing the complete cycle of analysis: upload of experimental data, apply the calibration tools to extract the material constants, save the constant to the database, and retrieve them in a form suitable to perform numerical simulations.

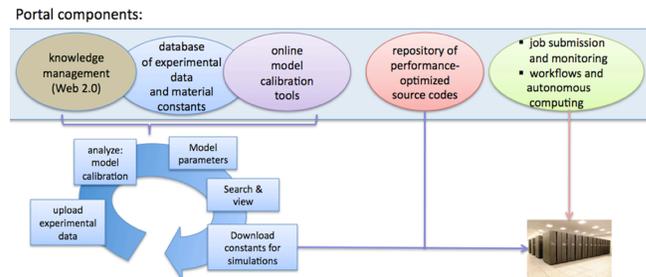


Figure 2: Integration of the portal component into a single application.

Architecture of RMP

RMP is implemented on top of two web services: Data Service and Compute Service.

Data service

Data Service aggregates three independent sub-services: metadata service, storage service, and replica locator.

Each experimental data set in the repository is stored in a file system. The storage service manages the part of the file system designated to store the data sets. When a file is submitted to the storage service, the service determines the location at which the file is to be stored, and returns its URI to the caller. GridFTP is used as the transport mechanism for moving the files to and from the storage.

The metadata service collects the information about data sets maintained by the storage service. The information is comprised of the file identification (a name assigned by the user, project, material, etc.), the data provenance (owner, date submitted, etc.), tags enabling querying the metadata repository to find particular data sets matching search criteria, and some additional information necessary to process the data (such as transformation from raw force-displacement measurement to stress-strain relationship). When a new metadata record is created, the service returns its URI, so that it can be referred to at a later time. The metadata repository is implemented as a DBMS application.

The replica locator provides mappings between the metadata records and data files.

Typically, entering a new data set to the RMP is performed in three steps. The metadata record is created (URI_{metadata}), the file is uploaded through storage services (URI_{data}), and finally, both URI are sent to the replica locator, where the mapping between them is

established. Similarly, retrieving a data set from the RMP is done in three steps. Firstly, the user queries the metadata repository to retrieve the URI_{metadata} of the desired entry. Secondly, the URI_{data} is retrieved from the replica locator. Finally, the file is retrieved by submitting the URI_{data} to the storage service. (Caching of the URI_{data} does not work, as during each step authorization is made).

Because of this complexity, the Data Services is implemented as a façade. A facade is service that provides a simplified interface to a larger body of code, such as a class library, or a collection of independent services. The use of the facade pattern has several advantages:

- It makes it easier to use and understand underlying services, since the facade has convenient methods for common tasks;
- It makes the code that uses the services more readable, for the same reason;
- It reduces the dependencies of outside code on the inner workings of the collection of services, since most code uses the facade, thus allowing more flexibility in developing the system;
- It wraps a complicated collection of APIs with a single well-designed API

Other data items are stored only as metadata.

Compute Service

The model calibration tools are implemented as MATLAB applications. In order to utilize them in the Web environment, a web service referred to as Compute Service has been developed. The service allows interactive remote access to the pool of MATLAB instances running on the back-end. A pool of MATLAB instances is necessary to serve multiple users concurrently. Similarly to Data Service, the complexity of the Compute Service is hidden by a façade. Currently, three MATLAB applications (model calibration tools) are implemented:

- Damage Model
- Image Analyzer
- Multistep Fatigue Fit

Damage Model: The Mississippi State University internal state variable (ISV) plasticity-damage model (DMG) production version 1.0 is based on the ISV plasticity formulation of Bammann [1] with the addition of porosity [2] and the void nucleation, growth, and coalescence rate equations that admit heterogeneous microstructures [3]. The model is implemented as an ABAQUS user material subroutine (UMAT).

The model calibration routine DMGfit was developed by Carino [4], which updated the original BFIT routine by Lathrop [5]. The calibrated model constants can be directly merged into the "USER MATERIAL, CONSTANTS" section of an existing ABAQUS input deck.

Image Analyzer: ImageAnalyzer is a utility for calculating some model constants from an optical image of a material. Groups of pixels in the image that satisfy user-specified criteria are interpreted to be objects of interest (particles, grains, voids, etc.). Associated with each object are the following quantities: area, centroid, first nearest neighbor distance, major axis length, minor axis length, and orientation.

The area is the number of pixels in the object. The centroid's x-coordinate is calculated as the average of the x-coordinates of the pixels in the object. The y-coordinate for the centroid is found similarly. The first nearest neighbor distance for an object is the distance of its centroid to the centroid of the nearest object.

The major axis length is the length of the major axis of the ellipse that encapsulates the object. The minor axis length is the length of the minor axis of the ellipse. The orientation is the degree angle between the x-axis and the major axis of the ellipse. The aspect ratio is the ratio of the major axis length over the minor axis length. Thus the aspect ratio is bounded below by 1, in which case the object would be a circle. Area and length are calculated in pixels; these are converted to microns using the scale (microns per pixel) of the image. The following are examples of constants derived from the above quantities:

- particle size
- particle volume (or area) fraction
- coalescence factor
- grain size or dendrite cell size
- initial void volume (or area) fraction

Multistage Fatigue Fit: MMF is a high fidelity multistage fatigue (MSF) model to predict the amount of fatigue cycling required to cause the appearance of a measurable crack, the crack size as a function of loading cycles. The model incorporates microstructural features of the fatigue life predictions for incubation, microstructurally small crack growth, and long crack growth stages in both high cycle and low cycle regimes.

Mashing-up Data Service and Compute Service

In web development, a mashup is a web page or application that combines data or functionality from two or more external sources to create a new service. The term mashup implies easy, fast integration, frequently using open APIs and data sources to produce results that were not the original reason for producing the raw source data. An example of a mashup is the use of cartographic data to add location information to real estate data, thereby creating a new and distinct Web service that was not originally provided by either source.

The RMP is a mashup of two independent services: data service and compute service, allowing a user to launch the model calibration tools in the context of selected data sets in the repository, and to automatically store the martial contexts back to the database.

Front End of RMP

The front-end of the repository of experimental data and the repository of material constants is implemented as a single web page. In addition, the front end of each tool is a separate single page. Each of those pages serves as a Graphical User Interface (GUI). The GUIs allow the user to interactively perform all actions supported by RMP by utilizing web widgets on the pages. GUIs are implemented using Asynchronous JavaScript And XML (AJAX). The processing of the actions associated with GUI widgets is performed either locally using JavaScript, or it is delegated to the server side through AJAX calls. Regardless whether the processing is local or remote, only a relevant fragment of the web page hosting GUI is updated; indeed, there is never a need to reload the whole page, unless a model calibration tool is invoked. By employing AJAX, the GUI has the look and

feel of a stand-alone application with a very rich interface. Figures A2-A5 show examples of the repository GUI.

Summary

This paper reports on the concept, design, and preliminary implementation of the cyberinfrastructure for the Integrated Computational Material Engineering. The repository of experimental data and material constants is available online and is being tested by a selected group of users, adding to collaborative research between Mississippi State University and the University of Virginia. The first production release (that will incorporate the feedback from the early users) is expected by the end of 2009.

References

- [1] Bammann, D. J., "Modeling Temperature and Strain Rate Dependent Large of Metals," Applied Mechanics Reviews, Vol. 43, No. 5, Part 2, May, 1990.
- [2] Bammann, D. J., Chiesa, M. L., Horstemeyer, M. F.,

Weingarten, L. I., "Failure in Ductile Materials Using Finite Element Methods," Structural Crashworthiness and Failure, eds. T. Wierzbicki and N. Jones, Elsevier Applied Science, The Universities Press (Belfast) Ltd, 1993.

[3] Horstemeyer, M.F., Lathrop, J., Gokhale, A.M., and Dighe, M., "Modeling Stress State Dependent Damage Evolution in a Cast Al-Si-Mg Aluminum Alloy", Theoretical and Applied Fracture Mechanics, Vol. 33, pp. 31-47, 2000.

[4] Carino, R., Horstemeyer, M., & Burton, C. (Dec 2007). Re-engineering DMGFIT[™] Fitting Material Constants to Internal State Variable Models. MSU.CAVS.CMD.2007-R0040.

[5] Lathrop, J.F., (Dec 1996). BFIT, A Program to Analyze and Fit the BCJ Model Parameters to Experimental Data Tutorial and User's Guide. SANDIA REPORT SAND97-8218 . UC-405, Unlimited Release, Printed December 1996.

Appendix

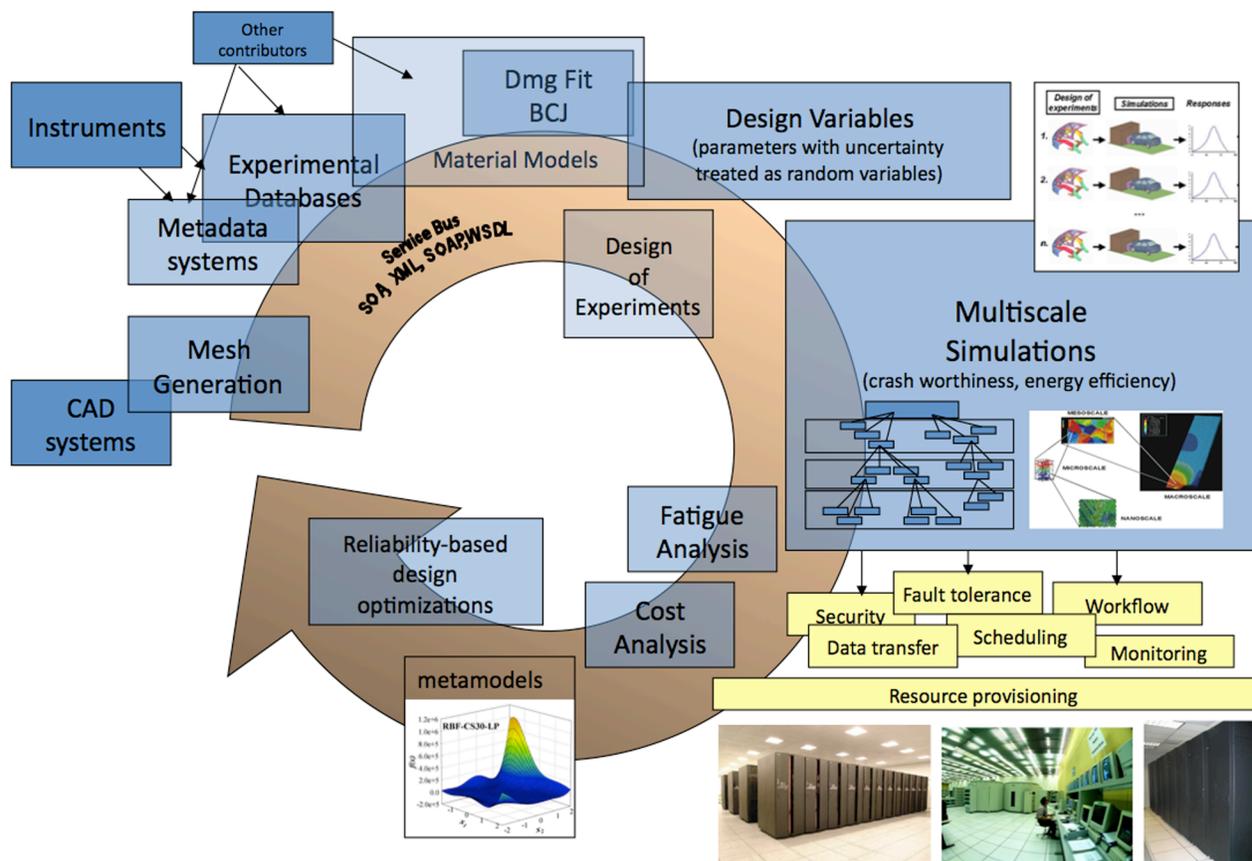


Figure A1: Long-term vision of the cyberinfrastructure for ICME. This envisioned system support performing multi-objective metamodel-based design optimizations involving multiscale simulations. The components of computational workflows (codes), models and data come from distributed repositories populated by the community. The workflows are managed by IT infrastructure acting autonomously, adapting to variable demand for resources by adaptive multi-scale simulations and to variable availability of resources.

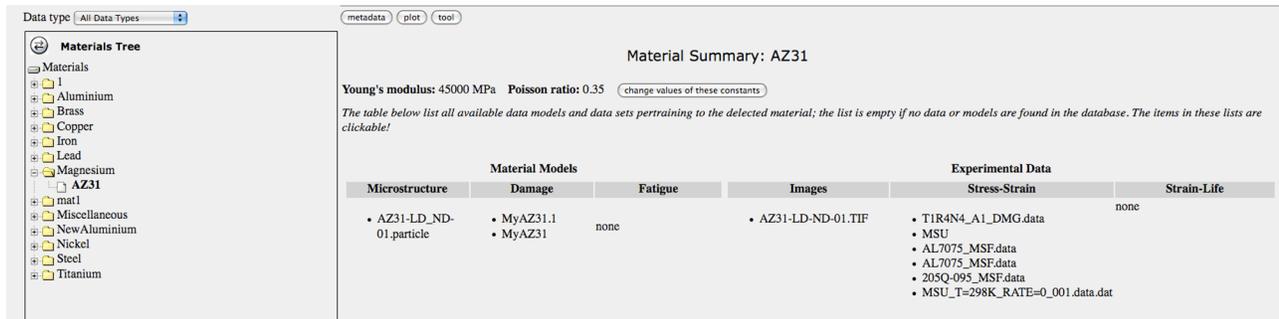


Figure A2: An example screenshot of the GUI of the repository: material summary: for a selected alloy of Magnesium (AZ31) the repository has several sets of stress-strain data and microstructure image, as well as two damage models and a result of the analysis of the image of the microstructure.

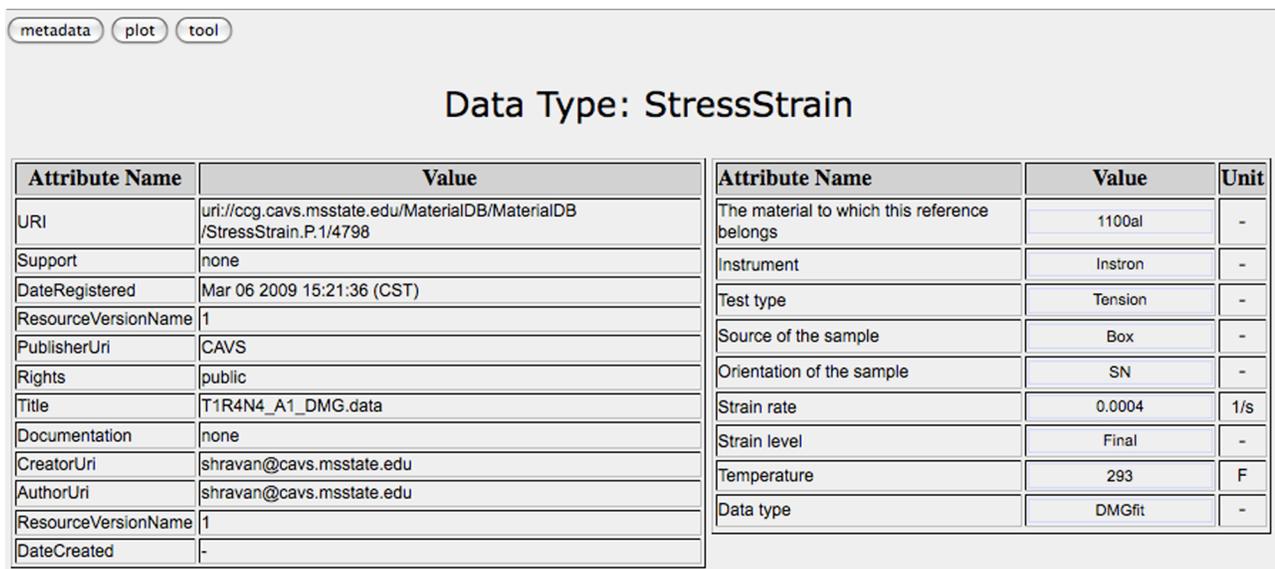


Figure A3: An example screenshot of the GUI of the repository: metadata viewer. The metadata for Aluminum 1100Al stress-strain data

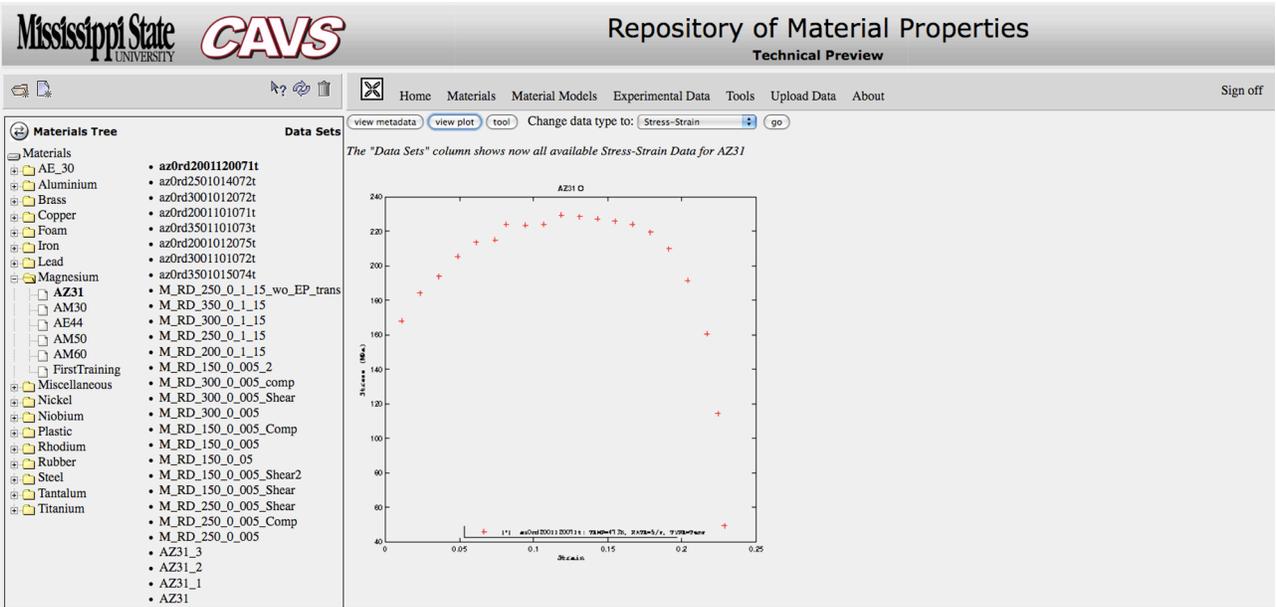


Figure A4: An example screenshot of the GUI of the repository: data viewer. The stress-strain curve example displayed in the GUI.

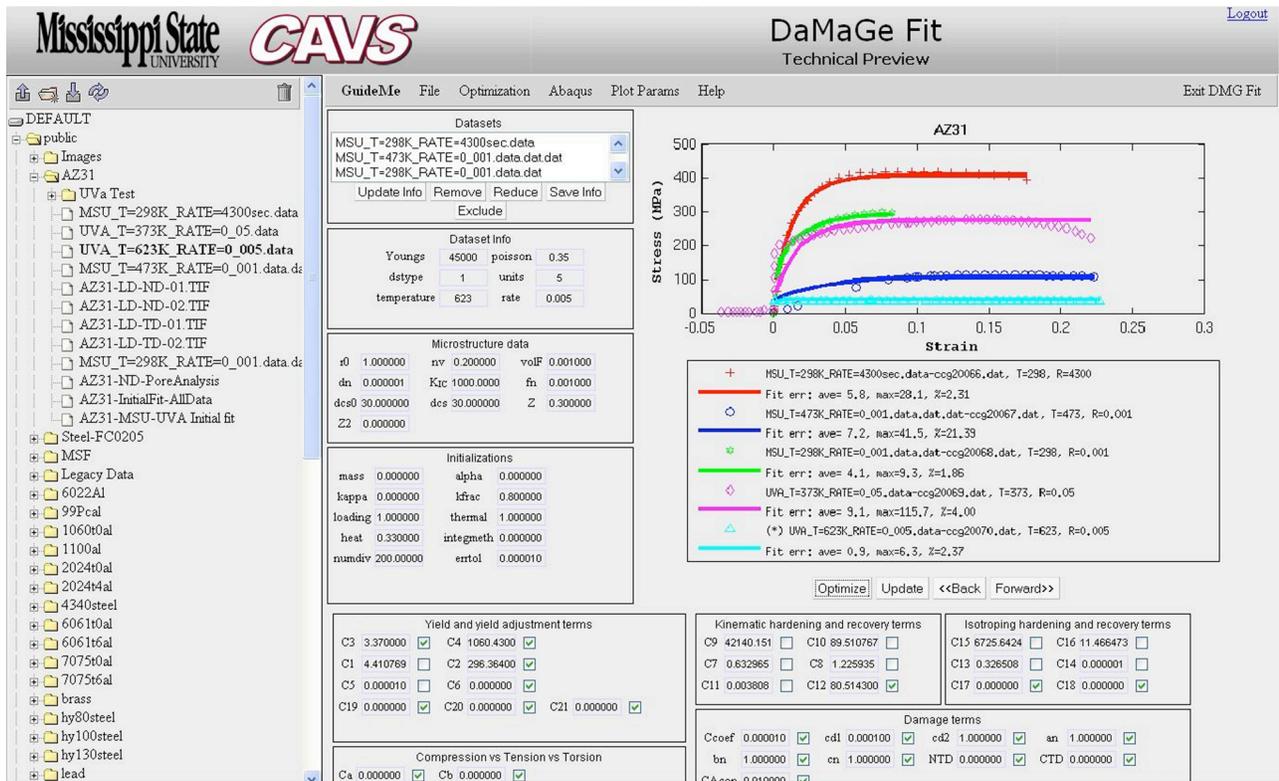


Figure A5: An example screenshot of the GUI of the repository: Damage fit interface