https://ntrs.nasa.gov/search.jsp?R=20180004469 2019-04-29T07:55:12+00:00Z

Integration of Information Management System, Workflow and Computational Tools Enabling Multiscale Modeling Within an ICME Paradigm

Steven M. Arnold* , Samuel C. Maphey†,
NASA Glenn Research Center, Cleveland, OH 44135

With the increased emphasis on reducing the cost and time to market of new materials, the need for analytical tools that enable the virtual design and optimization of materials throughout their processing - internal structure – property - performance envelope, along with the capturing and storing of the associated material and model information across its lifecycle, has become critical. This need is also fueled by the demands for higher efficiency in material testing; consistency, quality and traceability of data; product design; engineering analysis; as well as control of access to proprietary or sensitive information. Fortunately, material information management systems and physics-based multiscale modeling methods have kept pace with the growing user demands. Herein, recent efforts to develop a set of Python functions that exchange information between NASA GRC's Integrated multiscale Micromechanics Analysis Code (ImMAC) software toolset and its Integrated Computational Materials Engineering (ICME), Granta MI® database schema is presented. The goal is to enable seamless coupling between both test data *and simulation data*, which is captured and tracked automatically within Granta MI®, with full model pedigree information. These tools, and this type of linkage, are foundational to realizing the full potential of ICME, in which materials processing, microstructure, properties, and performance are coupled to enable application-driven design and optimization of materials and structures.

I. Introduction

With the increased emphasis on reducing cost and time to market of new materials, ICME (Integrated Computational Materials Engineering) has become a fast growing discipline within materials science and engineering. The vision of ICME is compelling in many respects, not only for the value added in reducing time to market for new products with advanced, tailored materials, but also for enhanced efficiency and performance of these materials. Although the challenges and barriers (both technical and cultural) are formidable, substantial cost, schedule, and technical benefits can result from broad development, implementation, and validation of ICME principles[1]. ICME is an *integrated* approach to the design of products, and the materials that comprise them, by linking material and structural models at multiple time and length scales.

A key ingredient is the linkage with manufacturing processes, which produce internal material structures, and in turn influence material properties and allowables, enabling tailoring (engineering) of materials to specific industrial applications. Figure 1 illustrates the interconnection of these scales and their cause/effect relationships, e.g., processing conditions produce a particular

*Technical Lead: Multiscale Modeling, Structures and Materials Division, 21000 Brookpark Road, AIAA member.
†Computer Engineering Intern, Clarkson University, Scientific Applications and Visualization Team.

microstructure from which properties are obtained, which then dictate a specific structural performance. Note that the evolution of elliptical line types (i.e., dotted to dashed to solid line) are purposely included to imply the level of maturity/understanding (from immature, to semi-mature, to mature, respectively) of modeling at each level of scale (both temporal and geometric). Furthermore, the figure illustrates the difference between two non-exclusive viewpoints; that is designing "with-the-material" (structural analyst viewpoint) versus designing "the material" (a materials scientist viewpoint). It is also apparent that the fundamental linkage between these two viewpoints is ultimately the associated constitutive model(s) for a particular material. It is critical to understand the input and output at each scale in order to determine the appropriate "handshaking" between scales and the meaningful properties that are ultimately required by a structural analyst.

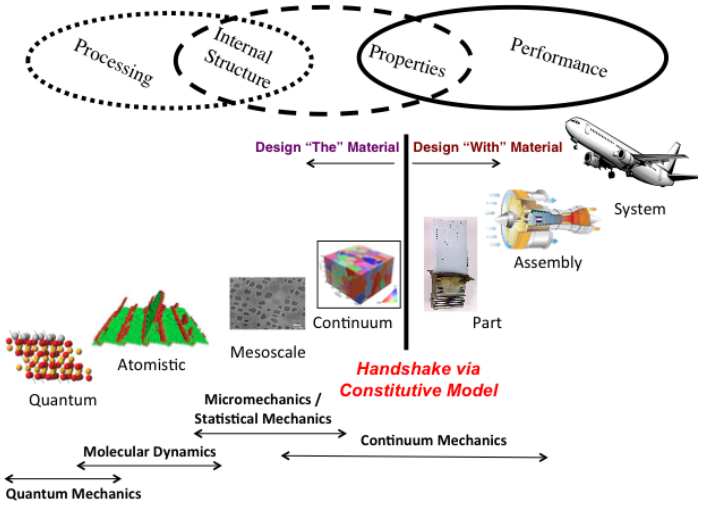


Figure 1 Description of associated length scale dependence and modeling methods in the context of ICME.

While there is a clear indication that ICME is growing, utilization of ICME in the daily work of researchers and engineers is still lacking. Two key contributing factors, since ICME is an inherently data intensive activity, are the lack of a robust information management system and the lack of a digital storage culture within most organizations. This stems from the fact that on the surface, a materials properties database may seem simply like a fancy means of storing, retrieving and distributing materials data; something akin to an electronic file cabinet. However, as discussed by Marsden et al.[2] and Arnold et al.[3], an effective ICME materials database (e.g., one in which experimental and computational mechanics are fully coupled) must allow the data inside the database to be easily accessible by analysis tools and allow the results from analyses to be read back into the database and stored with all of the associated metadata, while keeping track of associations across the full range of length scales.

This requirement greatly increases the need for data/metadata and contextual linkage so that knowledge can be both captured and reused. For example, the variety and complexity of modern materials, and their applications, necessitate complicated, and often extensive, materials testing. As for composite materials, large volumes of test data on various forms of the composites themselves, as well as individual constituents' thermal and mechanical behavior, are often required. Given a micromechanics based analysis approach, it is typical to require that data for each constituent be

reliably and conveniently traced back from the final products through their processing steps to the original raw materials. A second example is the need to provide adequate data to support increasingly sophisticated nonlinear, anisotropic and multi-scale engineering analyses. Here again, instead of storing a simple set of reduced, point-wise data, like elastic modulus and yield strength, the entire response (e.g., stress-strain, creep, relaxation, etc.) curves may be required. Collating, storing, processing, interacting with, and finally applying such data and metadata requires advanced dynamic information systems, enabling management of changing proprietary data alongside reference data collections, while ensuring consistency, quality, applicability and traceability.

Prior publications [3-6] discussed the data scheme, best practices, and informatics required to establish a robust, 21st century, information management system for capturing and analyzing material information. The goal of the information management system is to enable: 1) generalized constitutive modeling and 2) data-mining to establish microstructure/property/failure relationships for monolithic and composite materials. The proposed schema/ requirements for ICME were first demonstrated using a turbine disk Ni-based superalloy, in Arnold et al.[3]. Then, Arnold et al.[5,6] argued the importance of integrating both virtual (computationally based) and experimental data, over the entire material data life cycle and at various length scales, in the same information management system was essential for ICME to become a reality. The proposed ICME schema, which has been adopted by the MDMC[‡], is given in Figure 2. The specifics of the schema (i.e., required attributes) and the format (e.g., attribute type and record layout) for best storing such information were discussed in detail in Arnold et al.[5,6] for storing monolithic and composite material information at the coupon level. In the case of monolithic materials (e.g., fiber and matrix), three tables and their associated attributes were defined to enable the complete data life cycle to be handled, these are the: Deformation Table, Damage-Life Table, and Software Tools Table, see Figure 2. Whereas, in the case of composite materials one must think more broadly as multiple length scales can be involved depending upon the approach taken (i.e., macromechanics or micromechanics) to define the material's “constitutive model”. Consequently, the additional meso or macro scale above the constituent scale (e.g., that associated with monolithic material) necessitated the introduction of a fourth table, the Composite Table. Clearly, extension to other scales (e.g., atomistic, processing, microstructure modeling, structural) may require either the addition of new tables with appropriate attributes to the Model pedigree group within Figure 2 (e.g., Process Model Table) or new scale specific attributes to represent each new scale considered.

Figure 3 illustrates the interaction between experimental data and virtual data (data resulting from simulation tools) in that some experimental processing data (A) serves as input to a process model which in turn outputs some microstructural feature (W), which is stored in the database. This virtual microstructure data is then combined with measured microstructural data (B) and provided as input to a micromechanics and/or statistical mechanics analysis package, which then generates material property data (X, Y), which again is stored in the database. This property data (X,Y) is then subject to experimental validation (E,F), and also used in some continuum level analysis package (e.g., finite element analysis (FEA)) to assess some performance criteria (e.g., fatigue life,

---

[‡] The Material Data Management Consortium (MDMC) is a group of aerospace and energy sector organizations (both industrial and governmental) that have joined forces to develop best practices and associated software tools to integrate material and structural information technology with the realities of practical product design and advanced research. This group was established in 2002 through collaboration between ASM International, NASA Glenn Research Center and Granta Design Limited[7], see www.mdmc.net[8].

creep rupture, buckling load) Z; which is again stored in the database. Clearly, the present schema (with its assembly of model pedigree tables) not only allows model information and model parameters to be stored in a location that is easily accessible by FEA or other analysis codes through some type of interface software (e.g., Granta Material Gateway®), but also stores any associated simulation data necessary to assist in the evaluation, verification and validation of model output and certification of toolsets at multiple length scales. Also, once all of the input/output protocols are established, it can enable the seamless integration of these toolsets with optimization (e.g., OpenMDAO[9]) algorithms that will provide the final linkage of processing to performance criteria—thus realizing true ICME.

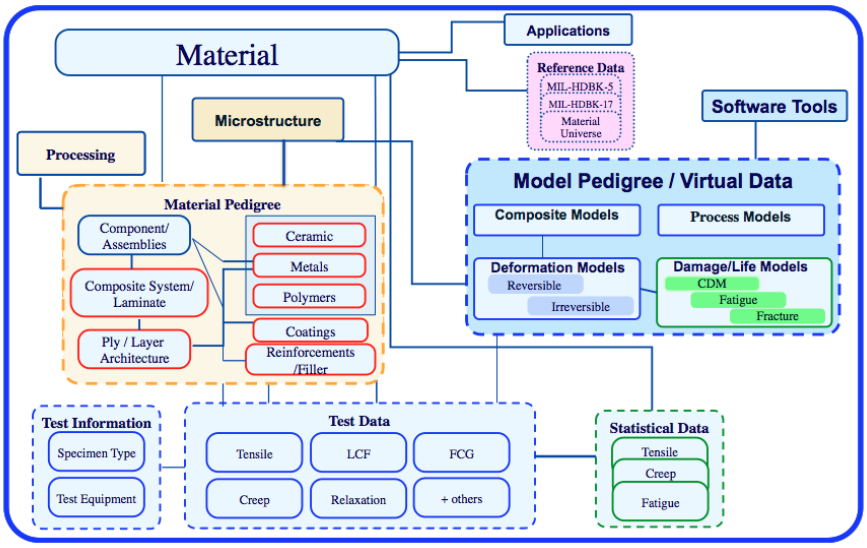


Figure 2 NASA Glenn's Schema Modified to Incorporate Virtual data to enable ICME.

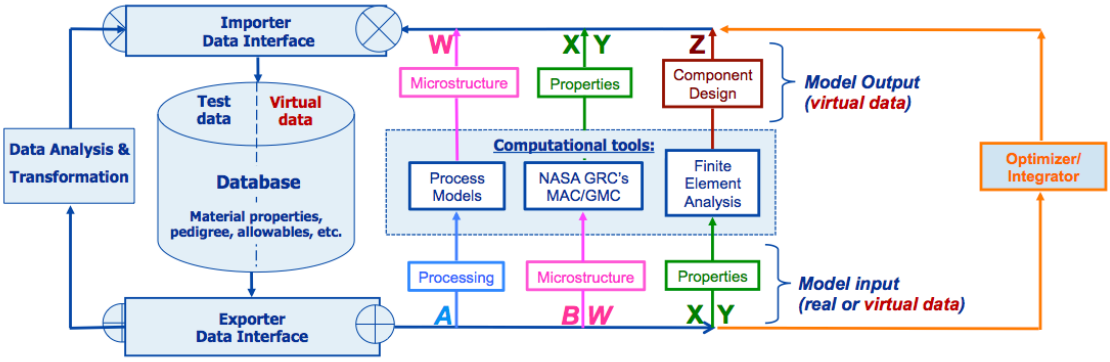


Figure 3 ICME infrastructure for housing modeling and testing information. Private communications with Dr. David Cebon, Cambridge University and Granta Design Ltd., 2013.

In a prior publication[10] it was demonstrated how the exchange of input/output information between Granta MI and NASA GRC's MAC/GMC code (illustrated in Figure 3) could be accomplished via web application and workflow engine. The web applications execute the requested MAC/GMC simulations, and the results (representing the composite at the macro scale), along with all metadata and model pedigree information, are then captured and stored in the database. On the micro scale, composite constituent materials from the Granta MI database are provided as input to MAC/GMC. Thus, the linkage exercise validated the schema on two levels of scale. In this paper, an alternative workflow engine (i.e. MATLAB) and associated Python application programing interface (API), which enable linkage of third party model simulation software (with full model pedigree storage) to the information management system, will be discussed. NASA GRC's MAC/GMC micromechanics analysis software will once again serve as the third-party modeling software, with the linkage to the Granta MI information management system being accomplished via the establishment of a number of specialized Python input/output applications. The micromechanics model, along with the newly developed workflow and Python, are described below.

II. Micromechanics of Composite Materials and Structures

In its broadest context, a composite is anything comprised of two or more entities with a recognizable interface (i.e., distinct internal boundaries) between them. If these internal boundaries are ignored, continuum mechanics can be used to model composite materials as pseudo-homogenous, anisotropic materials, with directionally dependent "effective," "homogenized," or "smeared" material properties. Micromechanics, on the other hand, attempts to account for the internal boundaries within a composite material and capture the effects of the composite's internal arrangement. In micromechanics, the individual materials (typically referred to as constituents or phases) that make up a composite are each treated as continua via continuum mechanics models, with their individual properties and arrangement dictating the overall behavior of the composite material. Over the past two decades NASA Glenn Research Center has been developing the ImMAC suite of tools for analyzing continuous, discontinuous, woven, and smart (piezo-electro-magnetic) composite materials and/or structures composed of such materials. MAC/GMC (a comprehensive and versatile stand-alone micromechanics analysis computer code), HyperMAC (the coupling of MAC/GMC micromechanics with the commercial structural sizing software known as HyperSizer[11]), MSGMC (the recursive coupling of micromechanics with micromechanics, for woven composites), and FEAMAC (the coupling of MAC/GMC micromechanics with the commercial finite element code, Abaqus[12]) make up this suite. At the core of these various tools is the well-known method of cells family of micromechanics theories (e.g., method of cells, Generalized Method of Cells, and High-Fidelity Generalized Method of Cells) developed by Aboudi and co-workers [13].

These methods provide semi-closed form solutions for determining global anisotropic composite properties in terms of the constituent material properties and arrangement, while also providing the full three dimensional stresses and strains in each of the constituent subcells. For a detailed, comprehensive discussion on modeling of composite materials, the reader is referred to the book entitled *Micromechanics of Composite Materials: A Generalized Multiscale Analysis Approach*[13]. Micromechanics based analysis lends itself to ICME in that it can link the processing and microstructure of the material directly to the resulting properties and performance of the material/structure, thereby enabling the engineer to not only "design-with-the" material but also concurrently "design-the" material. Consequently, by developing a database schema capable

of handling a micromechanics approach enables demonstration of an ICME capable (multiscale) database framework for composite materials.

A. The Generalized Method of Cells (GMC)

The Generalized Method of Cells (GMC) micromechanics theory assumes that a continuously reinforced composite microstructure can be represented as a collection of periodic repeating unit cells (RUCs) containing an arbitrary number of constituents, as shown in Figure 4. The RUC (indicated by a dashed line in Figure 4) is then discretized into N_β x N_γ rectangular subcells (in the case of doubly periodic GMC), as exhibited in Figure 5. Note, triply-periodic microstructures (e.g., particulate reinforced or 3D woven composites) although not discussed here, can also be represented by GMC. Each of these subcells is occupied by one of the constituent materials (e.g., SiC Fiber, BN coating, SiC matrix, and Free Si in the case of SiC/SiC composites). The number of subcells and the number of materials are completely general. In GMC, a first-order displacement field within the subcells is assumed, and displacement and traction continuity conditions are enforced in an average, integral sense at the subcell interfaces of the discretized RUC. These continuity conditions are used to formulate a set of semi-analytical linear algebraic equations that are solved for the local strains in each subcell ($\beta\gamma$) in terms of globally applied strains or stresses. Then, local constitutive laws can be utilized to obtain the local stresses in subcell ($\beta\gamma$),

$$\boldsymbol{\varepsilon}^{(\beta\gamma)} = \mathbf{A}^{(\beta\gamma)}\overline{\boldsymbol{\varepsilon}} + \mathbf{D}^{(\beta\gamma)}\left(\boldsymbol{\varepsilon}_s^I + \boldsymbol{\varepsilon}_s^T\right) \tag{1}$$

$$\boldsymbol{\sigma}^{(\beta\gamma)} = \mathbf{C}^{(\beta\gamma)}\left\{\mathbf{A}^{(\beta\gamma)}\overline{\boldsymbol{\varepsilon}} + \mathbf{D}^{(\beta\gamma)}\left(\boldsymbol{\varepsilon}_s^I + \boldsymbol{\varepsilon}_s^T\right) - \left(\boldsymbol{\varepsilon}^{I(\beta\gamma)} + \boldsymbol{\varepsilon}^{T(\beta\gamma)}\right)\right\} \tag{2}$$

where $\boldsymbol{\sigma}$ is the stress tensor, $\mathbf{C}$ is the stiffness tensor, and $\boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon}^I$, and $\boldsymbol{\varepsilon}^T$ are the total, inelastic, and thermal strain tensors, respectively, $\boldsymbol{\varepsilon}_s^I$ and $\boldsymbol{\varepsilon}_s^T$ are 6 by $N_\beta N_\gamma$ matrices containing all components of the inelastic and thermal strains, respectively, of every subcell (appropriately ordered), $\mathbf{A}^{(\beta\gamma)}$ is the strain concentration tensor, and $\mathbf{D}^{(\beta\gamma)}$ is the thermo-inelastic strain concentration tensor. Then, the generalized constitutive law for the effective, homogenized composite can be formulated as,

$$\overline{\boldsymbol{\sigma}} = \mathbf{C}^*\left(\overline{\boldsymbol{\varepsilon}} - \overline{\boldsymbol{\varepsilon}}^I - \overline{\boldsymbol{\varepsilon}}^T\right) \tag{3}$$

where the effective stiffness tensor, $\mathbf{C}^*$, effective inelastic strains, $\overline{\boldsymbol{\varepsilon}}^I$, and effective thermal strains, $\overline{\boldsymbol{\varepsilon}}^T$, are given by,

$$\mathbf{C}^* = \frac{1}{hl}\sum_{\beta=1}^{N_\beta}\sum_{\gamma=1}^{N_\gamma} h_\beta l_\gamma \mathbf{C}^{(\beta\gamma)}\mathbf{A}^{(\beta\gamma)} \tag{4}$$

$$\overline{\boldsymbol{\varepsilon}}^I = -\frac{\mathbf{C}^{*-1}}{hl}\sum_{\beta=1}^{N_\beta}\sum_{\gamma=1}^{N_\gamma} h_\beta l_\gamma \mathbf{C}^{(\beta\gamma)}\left(\mathbf{D}^{(\beta\gamma)}\boldsymbol{\varepsilon}_s^I - \boldsymbol{\varepsilon}^{I(\beta\gamma)}\right) \tag{5}$$

$$\bar{\boldsymbol{\varepsilon}}^T = -\frac{\mathbf{C}^{*^{-1}}}{hl}\sum_{\beta=1}^{N_\beta}\sum_{\gamma=1}^{N_\gamma} h_\beta l_\gamma \mathbf{C}^{(\beta\gamma)}\left(\mathbf{D}^{(\beta\gamma)}\boldsymbol{\varepsilon}_s^T - \boldsymbol{\varepsilon}^{T(\beta\gamma)}\right) \tag{6}$$

h_β and l_γ are the dimensions of the subcells, h and l are the dimensions of the RUC, and $\bar{\boldsymbol{\sigma}}$ and $\bar{\boldsymbol{\varepsilon}}$ are the effective (homogenized) stress and strain tensors. Extensive details regarding this derivation can be found in Aboudi et al. (2013)[13].

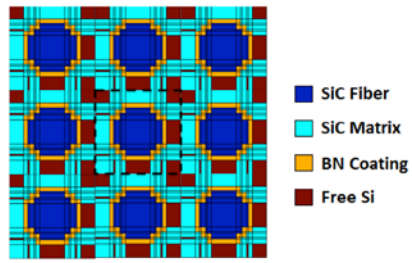


Figure 4 Representation of a doubly-periodic microstructure of a CMC composite material.

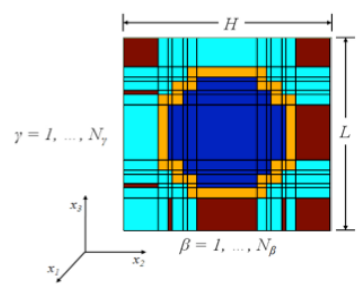


Figure 5 Discretization of a doubly-periodic RUC.

The developed ICME database schema is sufficiently general to admit all data associated with a micromechanics theory such as GMC. The present schema the various constitutent constitutive model parameters, be they reversible or irreversible, are stored in the individual folders/records associated with specific materials and constitutive models, for each material, in the Deformation Table. Whereas, the effective composite properties, **C***, and effective stress and strain responses, denoted by the overbar terms in equations (3, 5 and 6), and associated metadata (e.g., RUC details and simulation input file) are stored in the Composite Table. The specific attributes and associated data types, as well as the corresponding layout of information for the Deformation Table and Composite Table (see Table 1 below) are given in Arnold et. al. [5,6].

B. FEAMAC Multiscale Framework

A key ingredient of ICME is the linkage of the subscale effects to structural performance. As such, a synergistic multiscale framework (which executes concurrent multiscaling in time, but sequential multiscaling in space[14]) has been constructed to simulate the nonlinear response of composite structures by modeling the fiber-matrix architecture as an RUC at the microscale using GMC and coupling the microscale to a lamina/laminate level (macroscale) finite element model (FEM). The commercial finite element software, Abaqus [12], is used as the FEM platform, and the

MAC/GMC core micromechanics software[15,16] is used to perform microscale calculations. The scales are linked using the FEAMAC software implementation[17], which utilizes various Abaqus/Standard user subroutines. A schematic displaying a typical multiscale model using FEAMAC is displayed in Figure 6. The strains at the FEM integration point are applied to the RUC

Table 1 Layout and Attributes for Composite Model Table

Attributes	Type
Project Information	
Performing Organization	STXT
Project Name/Funding Source	STXT
Point of Contact (POC)	STXT
Material Description	
Material	STXT
Material Class	DCT
Commercial Name	STXT
Specific Name	STXT
Material Pedigree Record	Link
Batch Number	STXT
Material Notes	LTXT
General Modeling Information	
Model ID	STXT
Characterization/Analysis Date	DAT
Temperature	PNT
Temperature Range	RNG
Assumptions	LTXT
Micromechanics Modeling Information	
Micromechanics Method	DCT
Micromechanics Analysis Tool	STXT
Micromechanics Tool Information	Link
Micromechanics Input File	FIL
No. of Constiutents	INT
RUC/RVE Constituitve Description	TABL
RUC/RVE Image	PIC
Fiber Packing Arrangement	DCT
Effective Thermo-Elastic Composite Properties	TABL
Micromechanics Notes	LTXT
Laminate Level Modeling Information	
Laminate Name	STXT
Laminate Specification	STXT
Architecture Type	DCT
Laminate Pattern	DCT
Laminate Thickness	PNT
Ply Thickness (avg)	PNT
No. of Plies	INT
Laminate Definition	TABL
Laminate Analysis Tool	STXT
Laminate Analysis Tool Information	Link
Composite Laminate Analysis Input File	FIL
Laminate Notes	LTXT
Laminate Extensional Stiffness Matrix (A)	TABL
Laminate Coupling Stiffness Matrix (B)	TABL
Laminate Bending Stiffness Matrix (D)	TABL
Volume Fractions	
Total Matrix Volume Fraction	PNT
Total Reinforcement Volume Fraction	PNT
Total Void/Porosity Voume Fraction	PNT

Attributes	Type
Multiscale Modeling Information	
Multiscale Analysis Tool	DCT
Multiscale Analysis Tool Information	Links
Multiscale Analysis Input File	FIL
Multiscale Modeling Notes	LTXT
Composite Test Data Used for Characterization/Vaildation	
Tensile Test Data (Linked Records location in layout)	Links
Creep Test Data (Linked Records location in layout)	Links
RelaxationTest Data (Linked Records location in layout)	Links
Cyclic Test Data (Linked Records location in layout)	Links
Generic Test Data (Linked Records location in layout)	Links
Simulation Responses	
Stress vs. Strain Response (11 axis)	FDA
Stress vs. Strain Response (22 axis)	FDA
Stress vs. Strain Response (33 axis)	FDA
Stress (11 axis) vs. Time	FDA
Stress (22 axis) vs. Time	FDA
Stress (33 axis) vs. Time	FDA
Total Strain (11 axis) vs. Time	FDA
Total Strain (22 axis) vs. Time	FDA
Total Strain (33 axis) vs. Time	FDA
Shear Stress vs. Shear Strain Response (12 axis)	FDA
Shear Stress vs. Shear Strain Response (13 axis)	FDA
Shear Stress vs. Shear Strain Response (23 axis)	FDA
Shear Stress (12 axis) vs. Time	FDA
Shear Stress (13 axis) vs. Time	FDA
Shear Stress (23 axis) vs. Time	FDA
Total Shear Strain (12 axis) vs. Time	FDA
Total Shear Strain (13 axis) vs. Time	FDA
Total Shear Strain (23 axis) vs. Time	FDA
Force Resultant vs. Midplane normal strain (xx-axis)	FDA
Force Resultant vs. Midplane normal strain (yy-axis)	FDA
Force Resultant vs. Midplane normal strain (xy-axis)	FDA
Moment Resultant vs. Midplane curvature (xx-axis)	FDA
Moment Resultant vs. Midplane curvature (yy-axis)	FDA
Moment Resultant vs. Midplane curvature (xy-axis)	FDA
References	
General Modeling Notes	LTXT
Model References	LTXT

DCT Discrete Text (specified choices)
FDA Functional Data Attribute (with associated parameters)
FIL Allows the association of any file type to a given record
INT Integer Value
LOG Logical
LTXT Long Text Field
PIC Allows association of any image format to a given record
PNT Point value
RNG Range variable
STXT Short Text Field
TABL Tabular Attribute (multiple columns of data - PNT, STXT, DCT, INT, link)

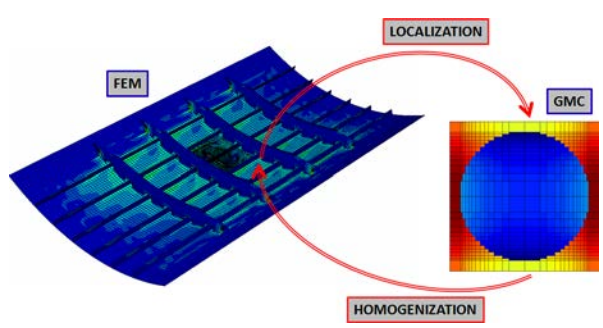


Figure 6 Diagram showing coupling of macroscale FEM and microscale GMC models.

and the local subcell fields are determined using GMC (this process is referred to as localization). If the subcell material behavior is nonlinear, the local stresses and strains are used to calculate the local stiffnesses, inelastic strains, thermal strains, and/or state variables via the local constituent constitutive laws. Homogenization of the RUC is then performed to obtain the global stiffnesses, inelastic strains, thermal strains, and/or state variables. The global stresses at the integration point are then calculated using these global, homogenized fields, and the effects of any nonlinear subscale phenomena are introduced into the macroscale through changes in the integration point stress state and stiffness. The global stresses, material Jacobian, and updated state variables at each FEM integration point are then supplied to Abaqus through the user material UMAT subroutine. For complete details on the FEAMAC implementation, the reader is referred to Bednarcyk and Arnold[16] and Aboudi et al.[12]

With the ability to link the GMC micromechanics model, which accounts for processing and microstructure while predicting properties (as discussed in the previous section), with a structural FEM, which simulates performance, the full range of ICME-related scales depicted in Figure 3 has been captured. Note the exact location of the simulation results of the structural analysis (i.e., stiffened panel in Figure 6) has yet to be finalized as they could go in a model pedigree table, the application table, or in a product life management (PLM) system external to the Granta MI database. More specific details regarding layout and associated attributes within the composite and software tables are given in the next section.

III. Workflow and Tools

In Figure 3, the high-level integration of the Granta MI information management system, extended to include both real and virtual data, was described. In this section, we will focus on the connection components (those represented by the Exporter Data Interface and Importer Data Interface in Figure 3) and their interaction with the associated database and computational tools. These components are foundational to a new workflow that allows for a user to interact with the information management system and other simulation tools associated with various length scales. Figure 7 illustrates this new workflow wherein a computational infrastructure (comprised of input/output files, workflow engine and various analysis codes operating at different length scales) is directly linked to the Granta MI information management system via export and import data interfaces (herein written in Python). In Figure 7, the exporter data interface is represented as Export Script, the importer data interface as Import Script and the computational tools highlighted within the grey box. The specific computational tools, as shown in Figure 3, can involve process models (which reflect process-microstructure relationships), micromechanics tools such as MAC/GMC as described in section 2 (which reflect microstructure-property relationships), and finite element analysis (which reflect property-performance relationships). The developed Python scripts were designed and implemented to enable interaction with or insertion within any of the referenced computational methods or workflow engines (e.g. MATLAB, Isight, etc.). The separation of workflow engine and analysis code, in Figure 7, is important as it enables integration of third party analysis software tools, in that specific input/output requirements can be isolated to substitutable modules of code, while not duplicating common functions between toolsets. The workflow engine drives the execution of a given problem, that is, the interaction between the executable third-party software package, in this case, MAC/GMC and Granta MI. With the information flow between the computational infrastructure and Granta MI occurring through its already established Service Layer via the Python input/output scripts, see Figure 8.

Figure 8, illustrates the application programing interface (API) developed by Granta Design, wherein multiple SDKs (software development kits) have been developed. Specifically, a linkage between the Granta MI database using C++ libraries to create Python classes that allow direct access to the data stored in Granta MI (MI Scripting Toolkit). This is represented as the Granta MI service Layer and Python MI Scripting Toolkit and will be the foundation to all of the following tools discussed herein, wherein the workflow shown in Figure 7 will be described in greater detail subsequently.

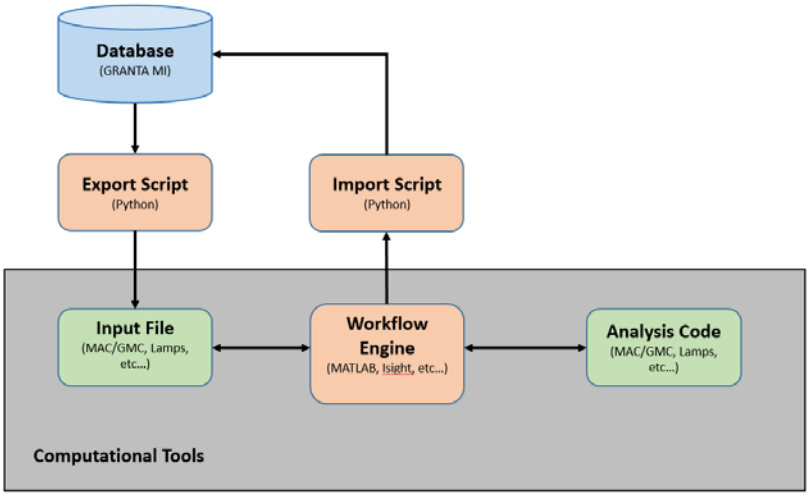


Figure 7 Schematic describing the information flow between a user, workflow engine, and the MI database.

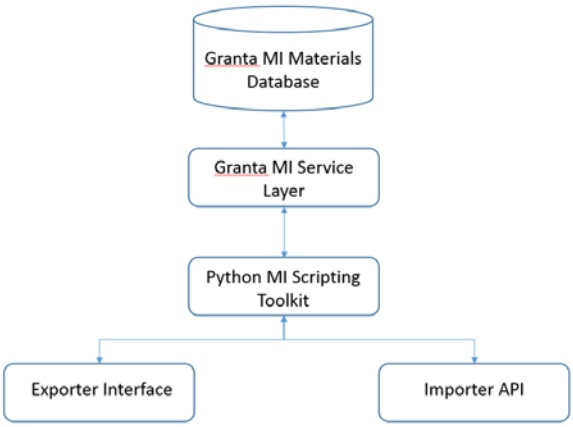


Figure 8 Diagram relating the developed code to the backend developed by Granta Design

The export script, discussed more in part IV, was developed as a graphical user interface (GUI) and API function calls, whereas the import script, discussed in more detail in part V, as merely an API. The decision to separate these in terms of their fundamental nature harkens back to the nature of the workflow in Figure 3. Wherein for a micromechanics based analysis, a user must be able to access/extract constituent materials property data from the Granta database with ease and simplicity. This typically occurs at the beginning of a given workflow in order to create an input

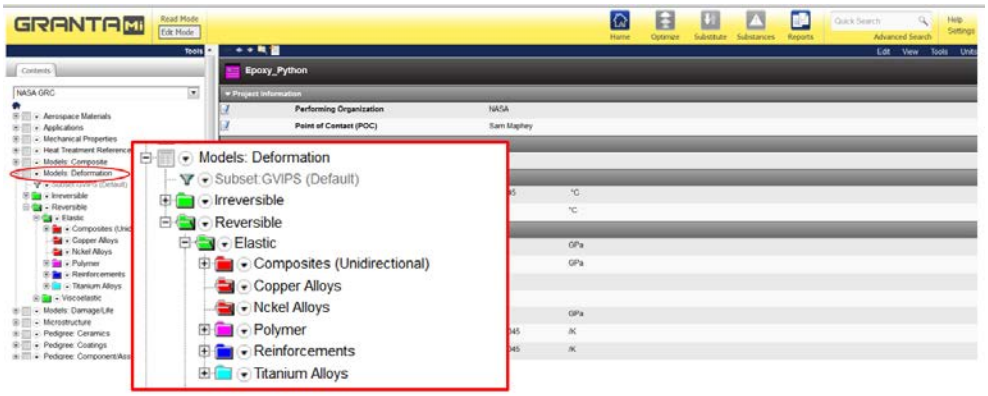
file that can be read by a selected analysis code. This input file will be created (based on a reference input file) with the appropriate model parameters and addition of unique MI identifiers, associated with the selected materials from the database itself. These identifiers are fundamental to the ICME infrastructure as they allow for full traceability and linking of materials at various length scales. Consequently, the use of a graphical interface eases the interaction between a user and MI database as they can simply search for a given material and retrieve its associated model parameters (e.g., thermoelastic properties). Furthermore, the decision to build numerous specific API function calls to enable importing of simulation results occurring at different points in the workflow (depending upon the computational tools being employed) was made to assist the analyst. By utilizing a workflow engine to drive the process it is much easier to create functions that facilitate the importing of results (e.g., effective composite properties and/or response curves) which arise from a given set of simulations (e.g., Monte Carlo study involving the influence of random microstructures).

To illustrate the above process, we will examine the effect of random microstructure on the response of a graphite-epoxy composite system. For instance, 5 graphite-epoxy composites samples are analyzed by changing their RUC (to simulate various random microstructures), simulating their response and then comparing the resulting stress-strain response including failure of each composite. To accomplish this example, the workflow engine used will be MATLAB, the computational tool MAC/GMC, and the two input/output interfaces based on the Python API. One of the key advantages to developing the exporter and importer using Python is its expansive compatibility with other scientific programming languages. Meaning that almost any workflow engine chosen will be able to call the designed API (specific Python functions) without the need for development of multiple APIs in multiple languages. It is also important to note that this workflow is not limited by the chosen workflow engine or computational tools. In the present example, MATLAB is taken to represent our workflow engine, but as stated previously any language/code that can interact with Python could be similarly supported. For example, in the future, extension to Simulia's Isight workflow engine will be explored. In addition, while the chosen computational tool herein will focus on MAC/GMC, this is not necessary as any other computational tool could be interfaced as well, with minimal effort. To conclude, the essence of this workflow is to link length scales to create a pedigree of data that can be traversed by examining the relationships between a composite and its constituent material properties.

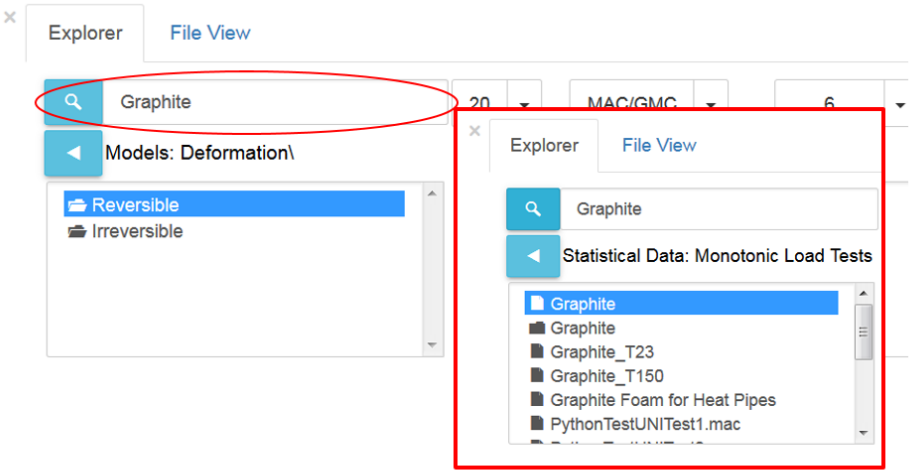
IV. Exporter Data Interface

To begin, the first aspect of our new workflow involves retrieving materials from Granta MI using a graphical user interface. This user interface was developed by building on top of the Python Scripting Toolkit provided by Granta MI and was built around the notion that a user's goal is to obtain constituent constitutive model parameters. This will involve the use of the deformation models table shown in Figure 9a; see Table 2 in Arnold et al. (2014)[5]. Given our focus on extracting materials data, the user is provided with two options that are shown in Figure 9b. They can either search through the database manually by navigating down the folders and subfolders to find a material or search by providing names of materials to the search bar. In our case, given a graphite-epoxy composite system, we will search for Graphite Fiber and Epoxy Matrix constituent properties. Granta's SDK gives the necessary functionality to provide the developed user interface with the necessary materials so that a user can quickly extract their material data. Along these lines the GUI, as designed, provides a user with a brief overview of the material they have selected,

describing the material class along with the deformation or damage model where applicable (see Figure 10). This is all in the name of retrieving real (experimentally obtained) material property data to build up a set of constituent representations that will be used to simulate the composite material. As the goal of the system is to create an information management scheme that provides full traceability, the constituent data extracted from Granta MI is also marked with a unique identifier that will reference the specific material record that it was extracted from. Denoted in Figure 11 as Record GUID, this data will provide the system with the ability to trace back from a composite system to its constituent materials. Figure 11 also illustrates how the model parameter data has been formatted to be consistent with the format required by the MAC/GMC software, since the goal is to perform micromechanics-based simulations. So, while this data has been extracted and formatted to fit this particular model, it can easily be changed to fit a different analysis code.



a)



b)

Figure 9 The Models: Deformation table here shown in Granta MI (a) and the developed User Interface (b).

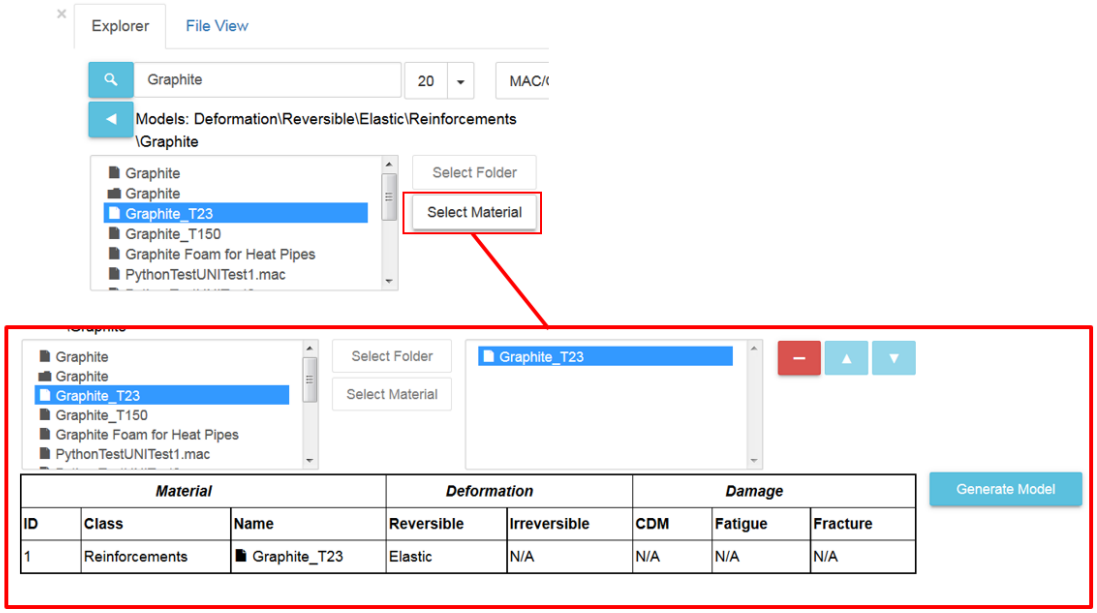


Figure 10 Selecting the material gives a brief overview of the material type and allows a user to generate a model.

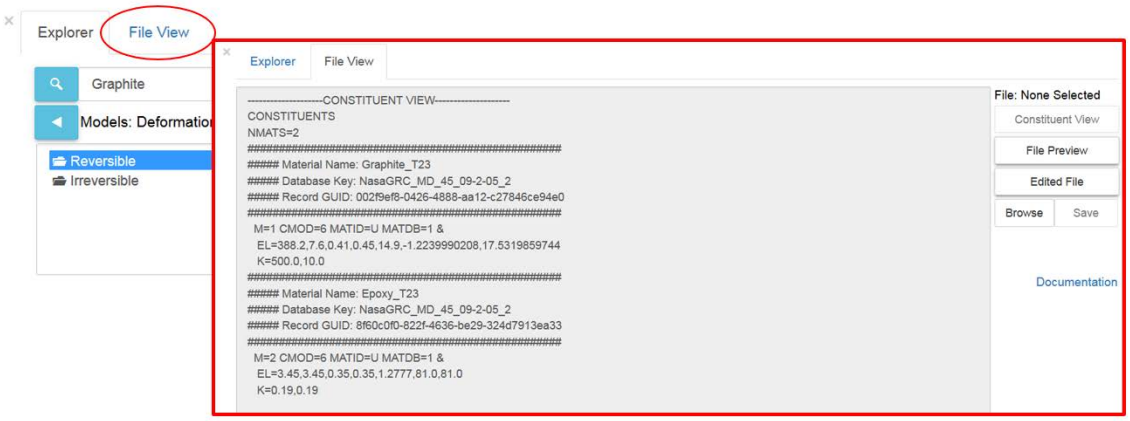


Figure 11 Material property data formatted in such a way as to fit MAC/GMC software.

To further our process and complete the simulation input data retrieval, a user will select a file that they want to use as their reference for running simulations. In MAC/GMC this is referred to as an input deck and is represented by the file we select, shown in Figure 12. This file (RefFile.mac) represents our original file which contains the necessary information to run a simulation using the computational tool (MAC/GMC). Here, all of the necessary information for our example problem regarding constituent behavior, the geometry of an RUC, failure criteria, and requested output (e.g., stress strain plot data) are defined. This file should, for the most part, retain its original structure except for the addition of the new constituent model parameters that the user has just created. These model parameters will replace the necessary keyword subsection in the MAC/GMC reference file

termed *CONSTITUENT with the new constituent data. Figure 13 shows how a user has the ability to look at these two files to confirm their accuracy and determine if the MAC/GMC edited file is correct. If it is, they simply save the file to the correct directory and if not, they can remove the constituent materials and select new materials to build their model.

This concludes the first step in our new workflow and has accomplished two objectives. A user has retrieved the model parameters for a constituent constitutive model representing the fiber (graphite) and matrix(epoxy) constituent materials that will be used to simulate the composite material behavior via MAC/GMC. As such, the original reference file's *CONSTITUENT section is updated with this new information along with the associated unique identifiers for each material that comprises the model. This is done to provide Granta with the necessary linkages that will allow for traceability in the input dataset and fully implement the ICME methodology by linking a composite material to its comprised constituents.

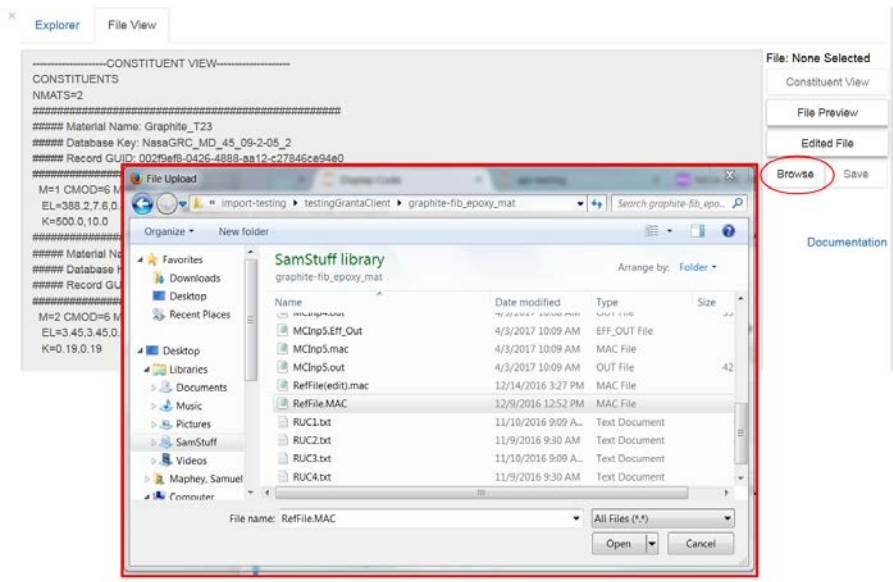


Figure 12 Illustrates how one would select a reference file.

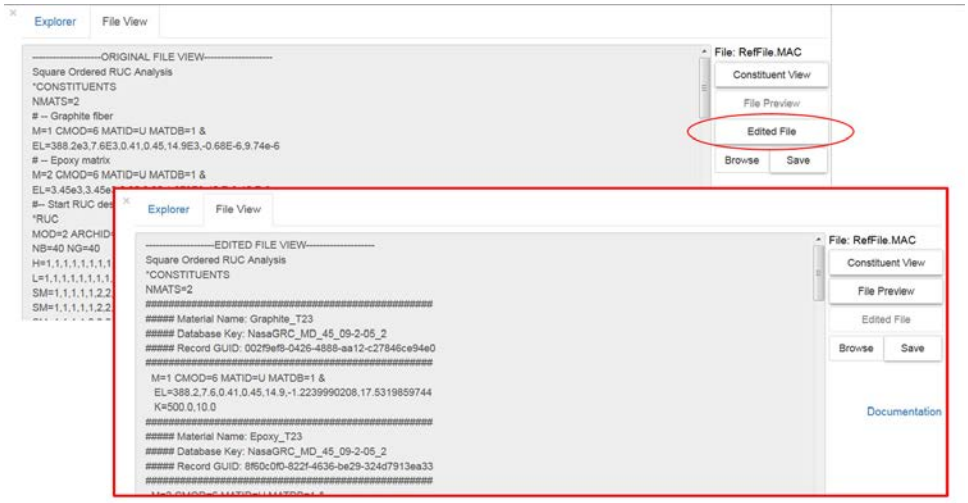


Figure 13 Original file and edited file views given to the user.

V. Importer Data Interface

The prior section outlined the first portion (exporting data) of our workflow for interacting with Granta. The following will describe the specific function calls needed to allow the import of simulation results back into Granta. This API has been written in Python to allow for use in many scientifically inclined programming languages. Given Python's open sourced nature and its popularity in scientific computing, the constructed API may be used in many workflows that employ other languages as the basis to their engine. So, despite these simulations being run in MATLAB, it is reiterated that the API calls can be used with any language compatible with Python.

Table 2a API functions for the user (a) and developer (b)

Function	*Purpose*
Backend/Utility Functions	**These control the addition of data into granta by file type**
update_discrete_types	To update any discrete attributes specified by attribute name
update_file_types	To update any file attributes specified by attribute name
update_hyper_link_types	To update any hyper link attributes specified by attribute name
update_integer_types	To update any integer attributes specified by attribute name
update_link_types	To update any link attributes specified by attribute name
update_logical_types	To update any logical attributes specified by attribute name
update_numeric_types	To update any numeric attributes specified by attribute name
update_range_types	To update any range attributes specified by attribute name
update_text_types	To update any text attributes specified by attribute name
update_tabular_types	To update an tabular attributes specified by attribute name
update_xy_plots	To update any XY Plot attributes specified by attribute name
import_records	Import any records to granta
import_folders	Import any folders to granta
import_generics	Import any generics to granta
Backend/Frontend Functions	**These update attributes based on their data type**
send_request_to_import	Bulk imports all data that has been built up
update_boiler_plate	To update boiler plate information relevant to any imported record
update_file_types	To update any file attributes
update_numeric_types	To update any numeric attributes
update_discrete_types	To update any discrete types
update_text_types	To update any text types
update_hyper_link_types	To update any hyperlink types
update_logical_types	To update any logical types
update_integer_types	To update any integer types
update_tabular_types	To update any tabular types
update_range_types	To update any range types
update_xy_plots	To update any xy plot types

Function	*Purpose*
Simulation Based Data Functions	**These import to multiple attributes based on the simulation run**
update_effective_output_from_file	To update the effective engineering moduli
update_laminate_results_from_file	To update effective laminate information
update_links_from_file	To update the links to materials used for the particular composite in the database
update_table_links	To update table links in a table
Get Functions	**These display data that is returned from Granta**
get_attribute_list	To retrieve a list of attribute names and their types from a given record

Table 2b API functions for the developer

Function	*Purpose*
Backend/Utility Functions	**These control the addition of data into granta by file type**
update_discrete_types	To update any discrete attributes specified by attribute name
update_file_types	To update any file attributes specified by attribute name
update_hyper_link_types	To update any hyper link attributes specified by attribute name
update_integer_types	To update any integer attributes specified by attribute name
update_link_types	To update any link attributes specified by attribute name
update_logical_types	To update any logical attributes specified by attribute name
update_numeric_types	To update any numeric attributes specified by attribute name
update_range_types	To update any range attributes specified by attribute name
update_text_types	To update any text attributes specified by attribute name
update_tabular_types	To update an tabular attributes specified by attribute name
update_xy_plots	To update any XY Plot attributes specified by attribute name
import_records	Import any records to granta
import_folders	Import any folders to granta
import_generics	Import any generics to granta

To start, Table 2 outlines the functions available to a user. The functions have subsequently been split into frontend (a) and backend (b). While both are available to an end user, the objective for creating the additional frontend functions was so that a given user need not remember the specifics of the specific Granta MI schema but rather focus on what specific data they want to import. These sets of functions will be used throughout the following example to illustrate how a user might run a Monte Carlo simulation and import the simulation results back into Granta. This simulation will create 5 composite materials and import the relevant data after changing the repeated unit cell (RUC) analysis portion of the MAC/GMC input deck. Figure 14 shows the two files (i.e., original and edited) that utilized/produced in the previous step (exporting). The relevant lines associated with the *RUC section of the input deck of interest (see Figure 14) will be replaced with the lines in the files named RUC1.txt, RUC2.txt, etc., respectively to define the five necessary MAC/GMC input decks.

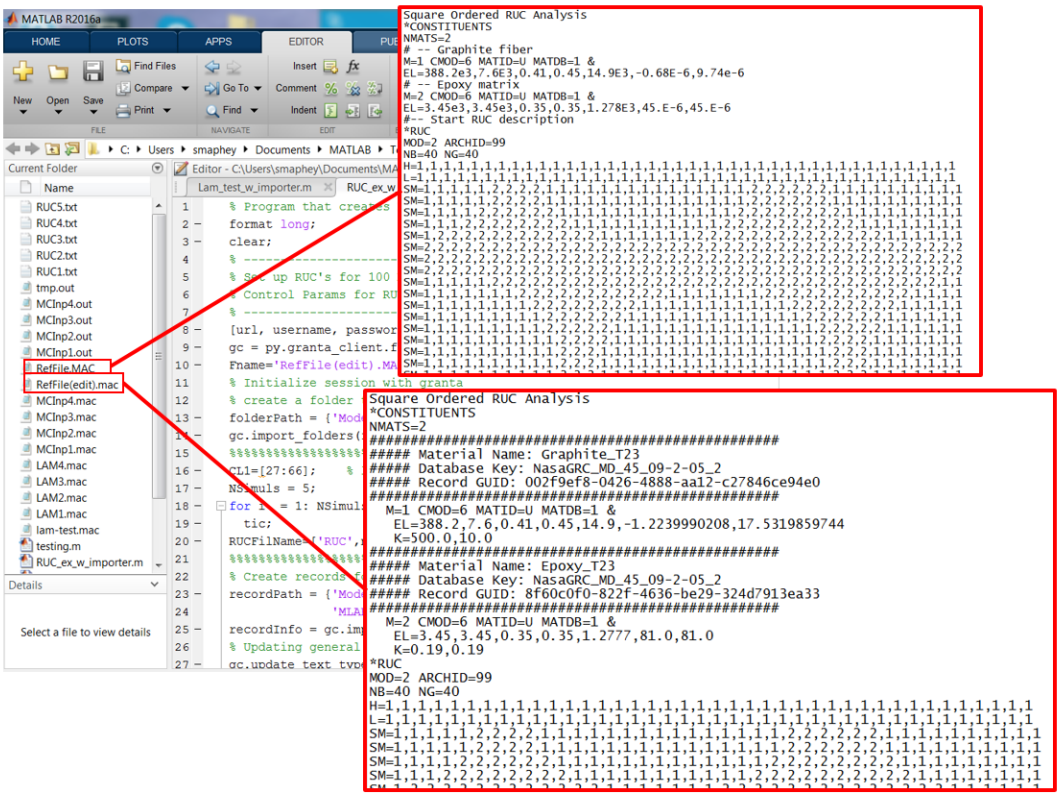


Figure 14 The original (RefFile) and edited (RefFile(edit)) files used by workflow engine (MATLAB script)

The example code shown in Figure 15 shows the beginning of our MATLAB script where we define the file of interest (RefFile(edit).mac). Here we will make our first call to the developed API. This call allows us to connect to the MI server and gain access to the Granta database. In addition, the tree structure (e.g., folder) that will hold the resulting simulation results (each run comprising a single record within the database) is specified according to the imposed database schema. Next, input decks describing five different random composite microstructures are sequentially created (via the workflow defined by the MATLAB script) by adjusting the RUC slightly using the RUC1.txt, RUC2.txt, etc. files. Boiler plate information indicating who, what, and why the analysis run is being conducted is imported within the given simulation record. This is outlined in Figure 16 which shows the relevant lines in the MATLAB script and the resulting data inside Granta MI. Finally, the simulations results, e.g., effective property data and stress-strain curves, are imported as shown in Figure 17, along with the model or simulation pedigree information such as the associated input and output mac files and links to the original material used to create the composite. By creating multiple virtual composites, it is possible to use Granta MI to make comparisons between multiple composites, as shown in Figure 18. Similarly, we can also run simulations to look at other composite features such as volume fraction instead of microstructure, see Figure 20 wherein results for a SCS-6/Ti-21S composite system are given. Thus, any composite material simulation workflow can now be created and executed so that the complete digital thread (including input, output, full traceability of both constituent and composite material behavior, software version, etc.) is captured.

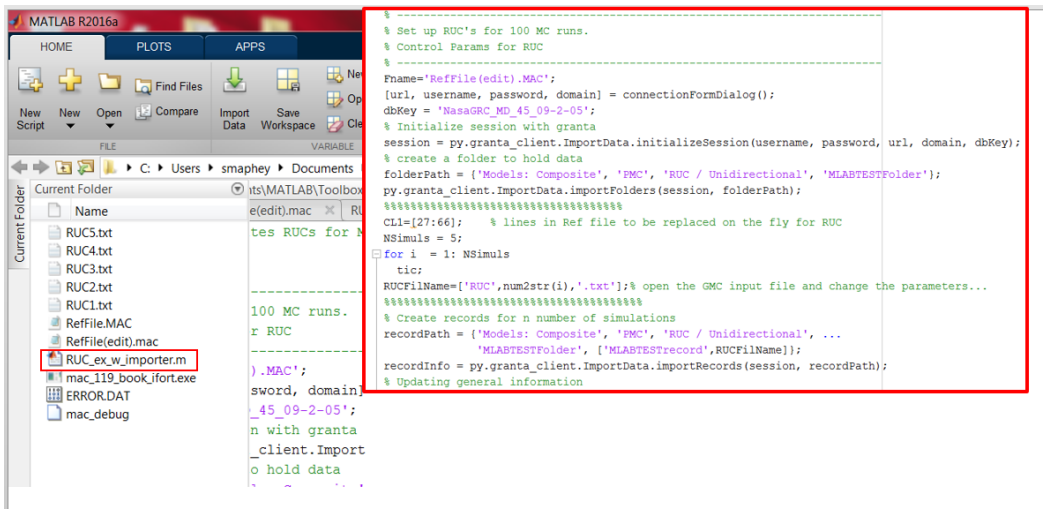


Figure 15 Beginning of the MATLAB script where a connection to Granta's server is made.

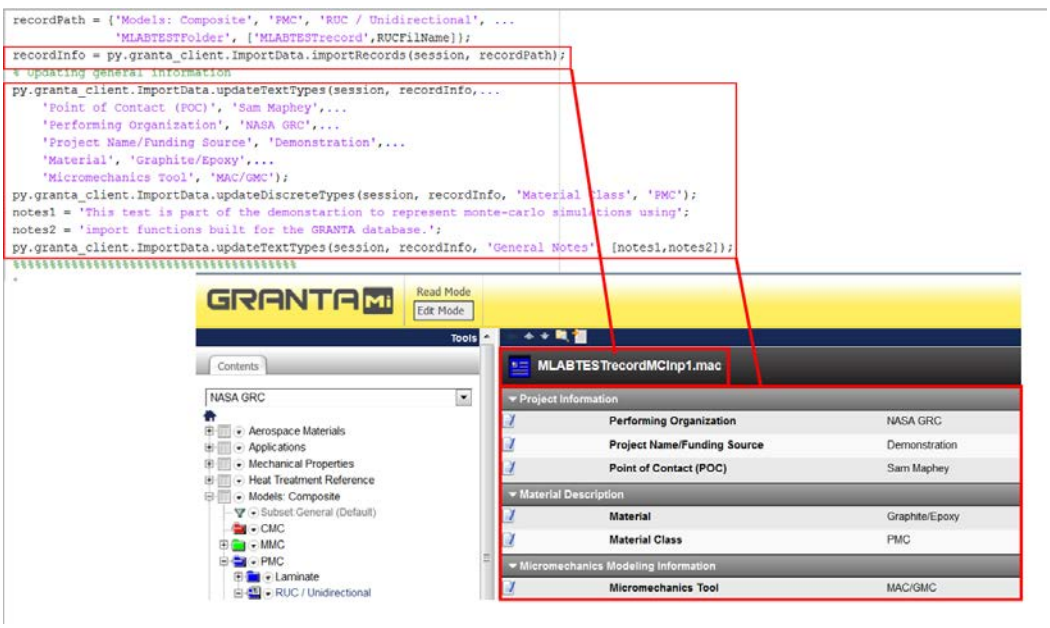


Figure 16 Boiler plate information that is contained in each simulation record.

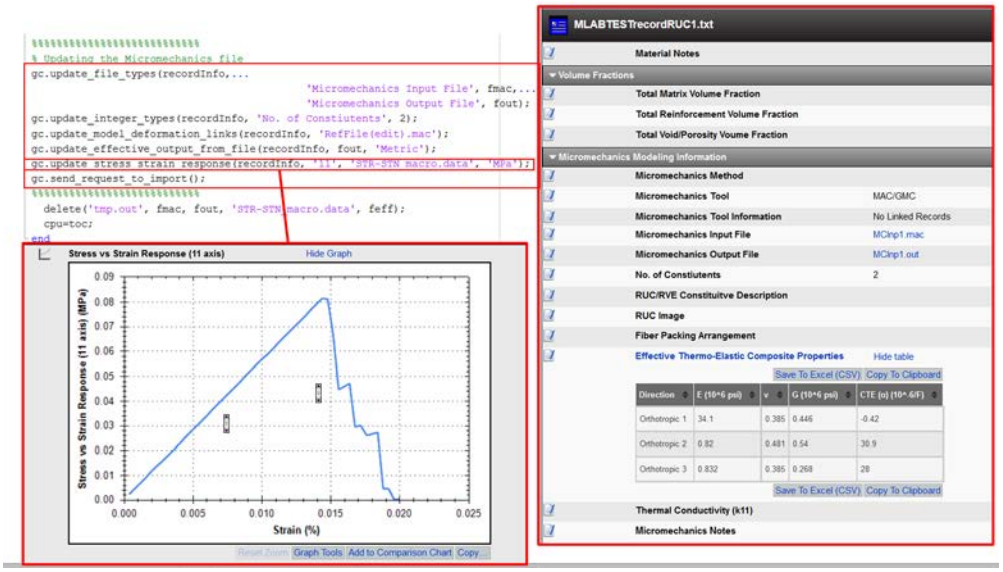


Figure 17 Illustrates model pedigree and composite results that can be included in simulation record

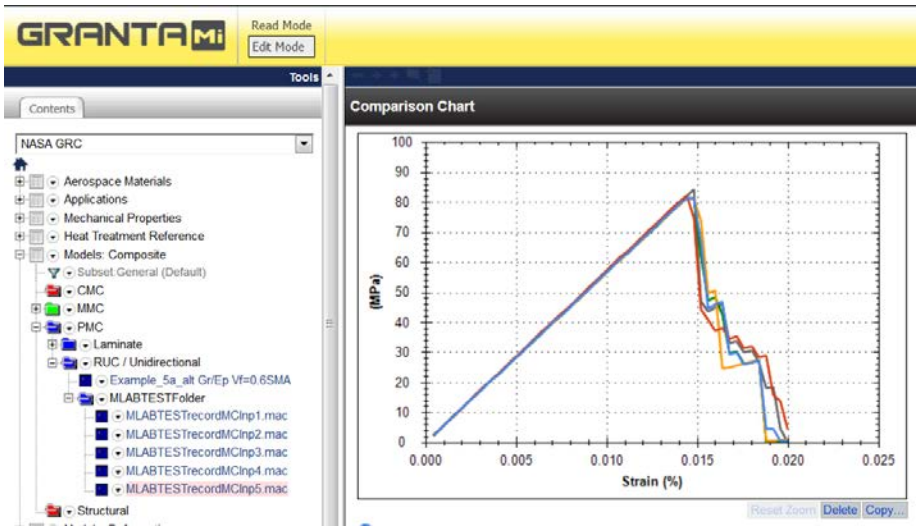


Figure 18 Comparison of five 60% volume fraction, Graphite/Epoxy, transverse stress-strain curves with varying microstructure.

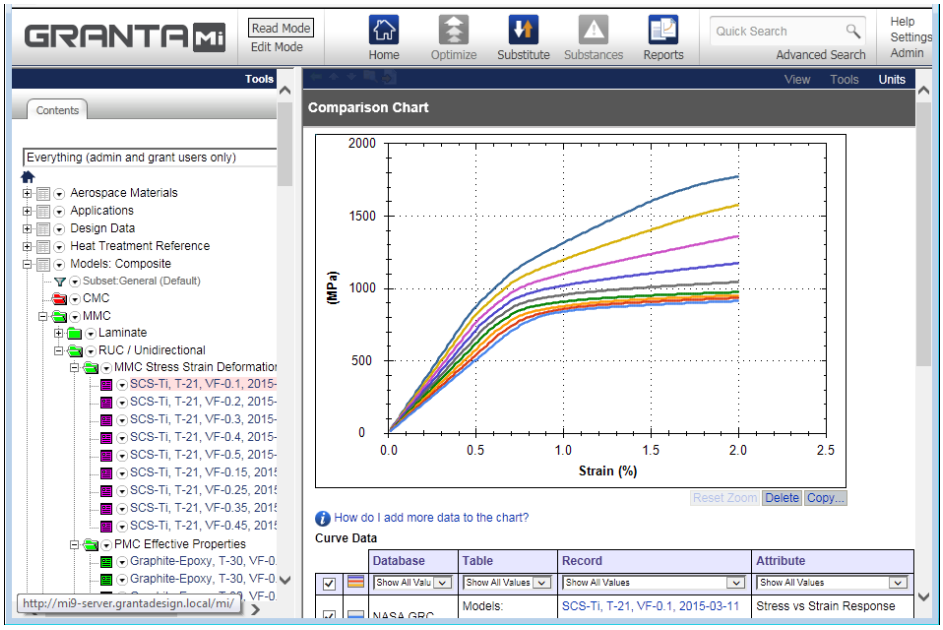


Figure 19 Comparison chart of fiber volume fraction effect for SCS-6/Ti-21S composite system

VI. Conclusions

As models become more accurate, their complexity tends to increase, as they rely less and less on simplifying assumptions. This complexity drives the need for more data to be measured, simulated, compared, stored, and tracked. The goal of ICME, to link mechanisms and models at multiple scales, drives these same needs and underscores the value of and need for a robust information management system. Often overlooked as a "mere database," this information management system should be viewed as a "necessary" or an "enabling" infrastructural aspect to ICME. In this paper, we have taken the next step in implementing a robust information management system by developing and demonstrating the usefulness of Python based API interface that enables linkage of third party model simulation software (with full model pedigree storage) to such an information management system (Granta MI) at two levels of scale. Such linkage opens up the design space and enables seamless and rapid connection of experimental data with virtual (simulation) data at various levels of scale, thereby enabling fit for purpose material tailoring.

References

[1.] National Research Council, Integrated Computational Materials Engineering: A Transformational Discipline for Improved Competitiveness and National Security, the National Academies Press (Washington, DC, 2008)

[2.] Will Marsden, David Cebon and Elizabeth Cope, "Managing Multi-Scale Material Data For Access Within ICME Environments," in Tools, Models, Databases, and Simulation Tools Developed and Needed to Realize the Vision of Integrated Computational Materials Engineering, S.M. Arnold and T. Wong (Eds.); ASM International, (Materials Park, OH, 2011).

[3.] S.M. Arnold, Holland, F.A., Gabb, T., Nathal, M. and Wong, T.; "The Coming ICME Data Tsunami and What Can Be Done", 54th AIAA/ASME/ASCE/AHS/ACS Structures, Structural Dynamics, Materials Conference, Boston, MA (April 23-27, 2013).

[4.] D. Cebon and M.F. Ashby, "Engineering materials informatics," MRS Bulletin, Vol. 31, No. 12, pp. 1004-1012, 2006.

[5.] Arnold, S.M., Holland, F. and Bednarcyk, B.A.; (2014). Robust Informatics Infrastructure Required For ICME: Combining Virtual and Experimental Data, 55th AIAA/ASMe/ASCE/AHS/SC Structures, Structural Dynamics, and Materials Conference, National Harbor, Maryland, 13 - 17 January 2014, AIAA-2014-0460.

[6.] Arnold, S.M., Holland, F.A., Bednarcyk, B.A., and Pineda, E.J.; "Combining Material and Model Pedigree is Foundational to Making ICME a Reality", IMMI, 4:4 DOI 10.1186/s40192-015-0031-2, 2015

[7.] Granta Design Limited, http://www.Grantadesign.com/, accessed on Dec 7, 2017

[8.] Official website of Materials Data Management Consortium, http://mdmc.net, accessed on Dec 7, 2017

[9.] Open MDAO, see http://openmdao.org, accessed on Dec 7, 2017

[10.] Arnold, S. M., Bednarcyk, B. A., Austin, N., Terentjev, I., Cebon, D. and Marsden, W.; "Information Management Workflow and Tools Enabling Multiscale Modeling Within ICME Paradigm", 57th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech 2016, ICME Special Session, Manchester Grand Hyatt, San Diego, California, Jan 4-8, 2016

[11.] Collier Research Corporation makers of HyperSizer; www.HyperSizer.com, accessed on Dec 7, 2017

[12.] Simulia Abaqus, a Dassault Systemes subsidiary; http://www.3ds.com/products-services/simulia/overview/, accessed on Dec 7, 2017

[13.] Aboudi, J., Arnold, S.M., and Bednarcyk, B.A. (2013) Micromechanics of Composite Materials: A Generalized Multiscale Analysis Approach, Elsevier, Oxford, UK.

[14.] Sullivan, R. W. and Arnold, S. M., "An Annotative Review of Multiscale Modeling and its Application to Scales Inherent in the Field of ICME," *Models, Databases, and Simulation Tools Needed for the Realization of Integrated Computational Material Engineering*, S. M. Arnold and T. T. Wong, Eds., ASTM International, Materials Park, OH, pp. 6-23, 2011.

[15.] Bednarcyk, B. A., and Arnold, S. M., "MAC/GMC 4.0 User's Manual – Keywords Manual," *NASA/TM-2002-212077/VOL2*, 2002a.

[16.] Bednarcyk, B. A., and Arnold, S. M., "MAC/GMC 4.0 User's Manual – Example Problems Manual," *NASA/TM-2002-212077/VOL3*, 2002b.

[17.] Bednarcyk, B. A., and Arnold, S. M., "A Framework for Performing Multiscale Stochastic Progressive Failure Analysis of Composite Structures," *Proceedings of the 2006 Abaqus User's Conference, May 23-25, Boston, MA*, 2006.