Title: An Implementation of ICME in Materials Information Exchanging Interfaces

Authors: Lianshan Lin, Weiju Ren

PII: S2352-9245(18)30006-1
DOI: https://doi.org/10.1016/j.md.2018.09.001
Reference: MD 52

To appear in:

Received date: 9-2-2018
Revised date: 3-7-2018
Accepted date: 6-9-2018

Please cite this article as: Lin L, Ren W, An Implementation of ICME in Materials Information Exchanging Interfaces, *Materials Discovery* (2018), https://doi.org/10.1016/j.md.2018.09.001

An Implementation of ICME in Materials Information Exchanging Interfaces

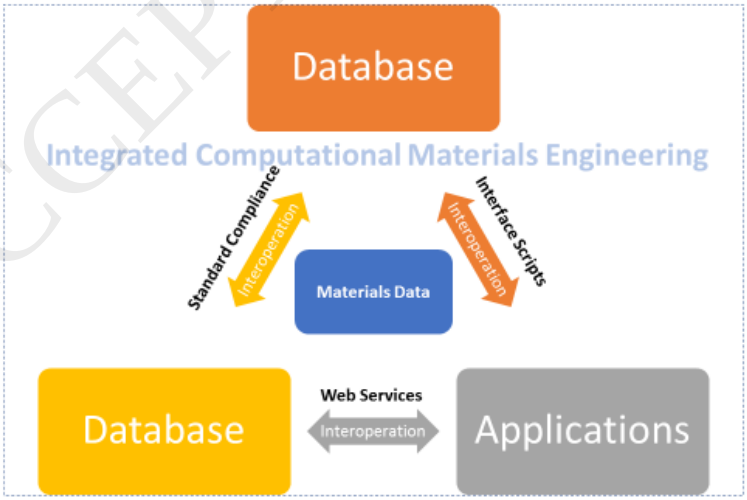
Lianshan Lin, Weiju Ren

Materials Science & Technology Division

Oak Ridge National Laboratory,

P.O. 2008, MS-6069, Oak Ridge, TN 37831, United States

Graphical abstract



Abstract: The Integrated Computational Materials Engineering (ICME) approach provides a new paradigm for improving the performance of existing materials or discovering and developing new materials. It focuses on developing effective connections between isolated engineering fields, to bring quantitative processing-structure-property relationships and abundant validated data that populate the knowledge base for accelerating the research of new materials while reducing the cost of development. The data exchanging interfaces play a key role in building such ICME connections among different materials models, simulation tools and individual organizations. With implementations of information exchanging interface between different materials database, and interface between database and applications, this article concludes that in building effective ICME applications, 1) standards-compliant interface can improve the exchange efficiency; 2) popular web service enables automated online materials data transfer; 3) customized data interoperation scripts can provide flexibility and productivity.

Keywords: ICME, interface, interoperation, materials database, implementation

1. Introduction

From raw materials to final products, the materials are involved in a complicated processing-structure-property relationship. After decades of development, commercial finite element analysis (FEA) and computational fluid dynamics (CFD) tools have enabled quantitative simulation on processing and structure. In contrast, nominal materials properties obtained from traditional mechanical tests or product applications are still the primary material indicators for FEA and CFD software. The integration of material properties, mostly with the simulation softwares only rather than within the complete cycle of material research, material processing and product manufacturing, will definitely lead to conservative designs, longer processing period and costly production [1, 2]. To improve the efficiency in manufacturing and shift the culture in materials development, the Integrated Computational Materials Engineering (ICME) was announced by National Academy of Engineering in 2008 [2]. The goal of ICME is to enable the optimization of the materials, manufacturing processes, and component design long before components are fabricated, by integrating the computational processes involved into a holistic system. ICME can be defined as the integration of materials information, captured in computational tools, with engineering product performance analysis and manufacturing-process simulation [2]. It emphasizes on the "I" for integrated and "E" for engineering, and targets on computational materials modeling as its destination. In that report [2], a number of lessons learned from the early applications of ICME approaches were documented. One of them was described as: “Databases are the key to capturing, curating, and archiving the critical information required for development of ICME.” Furthermore, it was emphasized that “for ICME to succeed, it must be embraced as a discipline by the international materials science and engineering community, leading to changes in education, research and information sharing.”

Following that National Academy of Engineering report, in 2011, the 1st World Congress on Integrated Computational Materials Engineering (ICME) was successfully held in Seven Springs, Pennsylvania, USA. Three major topics, modeling processing-microstructure and microstructure-property relationships, and ICME in education were discussed among global professionals. The 2nd World Congress on ICME was

held in Salt Lake City, Utah, USA in 2013. Progress in several challenging areas, such as multi-scale modeling, process optimization, model validation, experimental tools and ICME education were reported. Concerning materials data, challenges and approaches in materials data management, the fundamental databases, the ecosystem, and the applicable tools to support the flow of traceable materials information for ICME were discussed [3, 4, 5, 6]. A more recent review article authored by G. Xiong and G. B. Olson [47] has reported further progress in the tools developments, simultaneously D. L. McDowell and S. R. Kalidindi summarized several key elements to develop ICME ecosystem [48]. Although these efforts are absolutely and fundamentally necessary in building the ICME ecosystem, development of materials information standardization and automated data exchange techniques are indispensable as well because ICME cannot be effectively constructed on isolated materials information islands. After decades of accumulation of contents and technology in materials engineering, independent institutes and organizations around the world have developed various individual materials databases, such as the NIMS Materials Database (MatNavi) [7] developed by the Japan National Institute for Material Science (NIMS), the MatDB [8] online materials database developed and supported by the Joint Research Centre (JRC) of the European Commission, and the web-based materials database Gen IV Materials Handbook [9] hosted by Oak Ridge National Laboratory (ORNL), in different scales and formats.

In addition to the traditional materials databases that focus on bulk materials properties, more and more materials data repositories are devoted to materials properties down to the atomic scale, such as MATIN [10], MDF [11], the PRISMS Materials Commons [12], the Citrine database [13], NIST's MDCS and NMRR [14]. Meanwhile, a few collaborative ICME data frameworks have formed and evolved and matured, including the Integrated Collaborative Environment (ICE) developed by AFRL [15] which is based on HUBzero software [16] and its RESTful API, the science and engineering gateway NanoHUB.org [17], the Materials Commons platform [12] from the University of Michigan, and several cloud-based systems like Sumatra [18], the Citrine Informatics [13], and the materials data facility (MDF) [11]. International materials data efforts, such as efforts from the RDA/CODATA Materials Data, Infrastructure & Interoperability Interest Group [19], have never stopped encouraging exchange of computational and experimental materials data through shared online repositories, standardized formats and terminologies and open programming interfaces that are essential to accelerating the development of ICME. The privilege restrictions of data accessing in ICME infrastructure was an obstacle initially because most of the traditional materials data have been isolated and protected for a long time. Recently, along with the development of ICME ecosystem, more and more materials repositories have completely or partially opened their data content to the public or the community for free accessing, such as the MatNavi [7], the MatDB [8], the MATIN [10], MDF [11], etc., which might gradually reduce the difficulty of accessing and sharing materials data. Despite these initial successes, however, the terrain across the ICME frontier still remains largely uncharted, with many hurdles and challenges to overcome for maturation of ICME methodologies [20-27]. Instead of creating new materials database systems, bridging the existing isolated materials repositories might be a cost and time-saving solution with maximum degree of information shared for the development of ICME. The practical approach for consideration would be to promote standards-compliant schemas and ontologies that enable database interoperability through development of data exchange mechanisms, thus allowing participating

database systems and applications to largely keep their original terminologies, schemas, and formats, as long as their data can be transferred to each other through a viable means. Therefore, it's not surprise to see that the scope of the Second Workshop on Software Solutions for ICME held in Barcelona in April 2016 was summarized in one word: Interoperability [28].

As case studies in materials information standardization and communication, three data exchanging interface examples are introduced in Section 2, 3 and 4 respectively. Section 5 concludes the accomplishments of this article. These examples illustrate the data interoperation between different databases, and data interoperation between database and application as well. Challenges and technical obstacles encountered in these cases, typical or not in ICME infrastructure, along with final resolutions will be described in the following sections. The key contributions of our work in this article include: i) develop information exchanging interface for tensile test in compliance with ISO-6892; ii) implementation of automated online materials data transfer; iii) demonstrate the flexibility and productivity of customized data interoperation scripts that could potentially benefit development of ICME. Through the implementation instances, this article concludes that in building effective ICME applications, 1) standards-compliant interface can improve the exchange efficiency; 2) popular web service enables automated online materials data transfer; 3) customized data interoperation scripts can provide flexibility and productivity. The successful implementation of these interfaces would evolve useful technologies and experiences, and shed light on the data interoperability for ICME infrastructure development.

2. MatDB and Gen IV Materials Handbook

The Online Data and Information Network (ODIN) is hosted by European Commission Joint Research Centre Institute for Energy and Transport (EC-JRC-IET), providing the full cycle of data entry, retrieval and analysis over the Internet, to support European Union projects and network partners in energy and transport research. The facility consists of a collection of online databases organized into four main categories: documents, engineering, nuclear, and product information. The Materials Database (MatDB) [8] is one of the ODIN online databases specialized in the engineering category for materials test data that has a robust data model, comprehensive test support, and an intuitive user interface [30]. The MatDB database covers mechanical properties, thermo-physical properties, and corrosion data of engineering alloys generated in accordance with international material testing standards and recommendations. It contains over 20,000 test results accumulated after decades of development. Eventually, it aims at the direct web-enabled data entry from test machine into database for European Union projects and network partners by using XML (eXtensive Mark-up Language) [31].

Following the early work of ASTM E49 [32], five main entities, namely, Source, Specimen, Condition, Materials and Test Results, form the basis of the MatDB data model [30]. For the purposes of data exchange, the MatDB data model is also manifested as an XML schema definition (XSD). Each entity has sub-schemas with more specifications in different levels, building the whole framework of the database. The hierarchical schemas end with attribute elements, which are the metadata to be assigned with different types of values. In the tree-like MatDB system, the schemas and their sub-schemas build up

the branches, while the detailed attributes are leaves at the end. Fig. 1 depicts part of the MatDB structure.

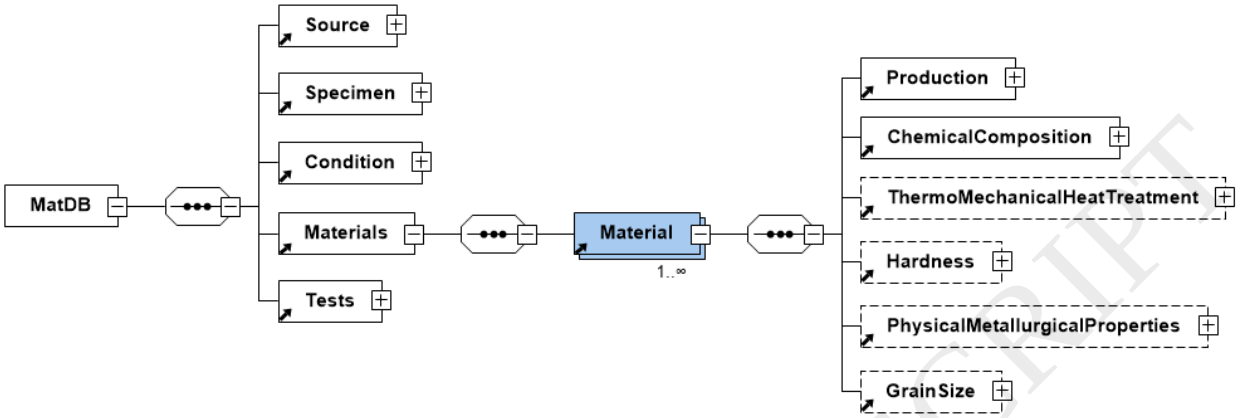


Figure 1 Hierarchical tree of MatDB

Since the MatDB XML schemas are a representation of data structure of the database application, experimental data can be imported into database efficiently [33]. Reversely, the binary data stored in the system can be exported to external XML data files that are compliant with the MatDB schemas.

The beginning of Gen IV Materials Handbook database system can be traced to 2005, and when its first beta version was released in 2006 [9]. Evolved with the development of Gen IV Nuclear Reactor Systems, the Gen IV Materials Handbook system has been upgraded to Version 5. More details about the Gen IV Materials Handbook system can be found in reference [29].

The entire system of Gen IV Materials Handbook is constructed with several Parts [9]. The early design of the Parts and the interrelationships are illustrated in Fig. 2. The solid black line between two Parts indicates their internal connection, which means an internal hyperlink could be created for two records in different Parts. Each individual Part has at least one schema designed for specific data structure. This architecture design promises a maximum degree of traceable records with least abundant information, but keeps the possibility of any structure modification in further development. By comparing the system structures in Fig. 1 and Fig. 2, it's obvious that these two database systems, the MatDB and the Gen IV Materials Handbook, have been designed in different ways which share very few, if not none, data schemas in common. MatDB grows its materials content in a tree like way, which always has a root element and the layers of sub-element can be increased to any arbitrary number. This tree structure takes less efforts in constructing its elemental contents, however costs more time to locate a specific information node particularly when the layer number increases. The structure of Gen IV Materials Handbook is more or less in a flat but interlinked network. Designing the schemas of Handbook system might need more endeavors since all the information has to be well organized and categorized. The flat structure promotes the speed in finding any elemental detail and keeps the record as concise as possible, however too many leaps among the internal links might affect user's impression on the information entirety.

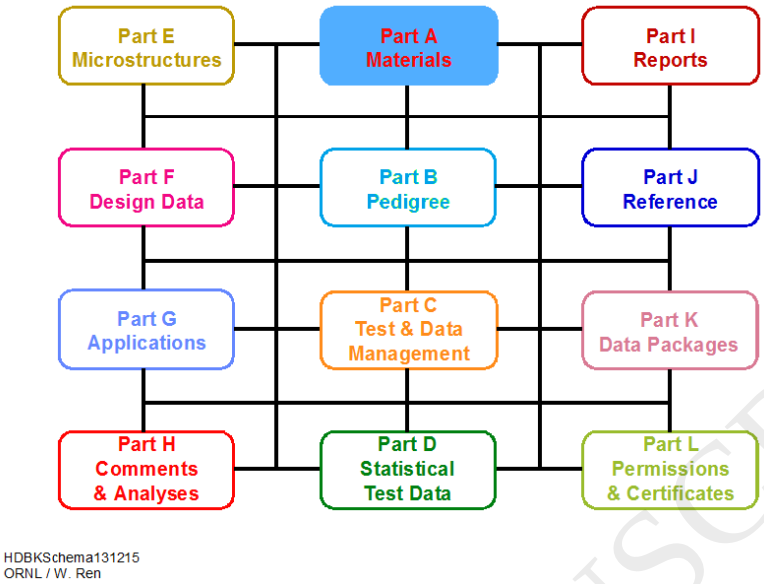


Figure 2 Overview of Gen IV Materials Handbook system

The Materials Handbook database has integrated data import/export functionalities as well. Existing database records can be exported into external files such as Microsoft Excel, plain text or XML files. With specific formats, external data in Excel files, plain text files or XML files can also be transferred into the database through its online service or client software tool.

As numerous components in Gen IV nuclear reactors are exposed to high temperatures, neutron fluence, and corrosive environments, safe and economic system operations necessitate extended materials qualification testing, which are extremely time and cost consuming. Interoperation of these precious testing data among databases and organizations will significantly reduce costs associated with redundant materials testing programs, enable improved auditing traceability and support the reuse of data. However, existing materials test databases currently present compatibility challenges: they differ in format and associated semantics; they are stored in both heterogeneous and distributed database repositories; these repositories have different working environments and software tools to access the data; and these environments and tools are in a constant state of change. Regarding the discussion above, MatDB and Gen IV Materials Handbook systems might be two of the typical databases born with these tremendous differences.

Facing the practical needs and challenges, data interoperation between MatDB and Gen IV Materials Handbook aims to extend their advantages by sharing the data of them, reducing the time and cost in testing and collecting properties data of the same materials. However, all the existing differences between them constitute the near-impossible obstacle on the way of the direct server-to-server data communication. Eventually, a solution using external XML files as intermediate data container was found to achieve the interoperation by taking advantage of the import/export functionality of both systems, and the standard compliant schemas (e.g. ISO-6892-1:2009 for tensile test). The utilization of a data schema that is compliant with an ISO testing standard has proved particularly effective in enabling systems integration because testing standard encapsulates accepted procedure, terminology, and

outputs, all of which define the form and format of generated data when the testing standard is followed. Utilization of ISO test standard in database interoperation also saves significant efforts of matching up the exported data elements from these two distinct systems. More technical details have been reported in reference [29]. Meanwhile, a XML interpreter, the EXtensible Stylesheet Language Transformations (XSLT) was employed to transform corresponding XML elements from MatDB-exported XML file to Handbook-exported XML file, or vice versa. Fig. 3 demonstrates the transformed XML document that is exported from Gen IV Materials Handbook but fits with the MatDB import requirement. The same transformation procedure can also be applied on the external XML documents exported from MatDB system, to generate XML files that can be imported directly into the Gen IV Materials Handbook . In addition, the export, transform and import procedures can always be achieved automatically through either online operations, or by client tools. Thus an efficient two-way data exchanging interface by using XML document as intermediate has been effectively realized.

Figure 3 The transformed XML file from Gen IV Materials Handbook

3. Gen IV Materials Handbook and Cubic Perovskites Database

Fuel cells are clean energy-conversion devices that can achieve high efficiency. Proton conducting ceramic fuel cells (PCFC), being one of the promising fuel cell systems, can operate at relatively low temperatures (~400-700 °C), which significantly improves material reliability and operational efficiency. Perovskites have been studied as potential proton conducting materials for PCFC for years. However, pure perovskites such as $BaZrO_3$ (BZO) do not conduct protons. Doped with acceptor dopants such as yttrium, perovskites can be transformed into proton conductors. To identify combinations of dopants and host-materials that might have optimized proton conductivity, a defect-database that focuses on energetics involved in formation, interaction and migration of various types of point defects is needed. A recent effort at ORNL [34] built a computational framework to perform high throughput *ab initio* calculations to compute the genomics of various point defects in more than eighty different ABO_3 cubic perovskite structures. The following combinations of A ϵ {Mg, Zn, Cd, Ca, Hg, Sr, Ba} and B ϵ {Mn, Si, Ge, V, Rh, Ti, Ru, Ir, Os, Sn, Hf, Zr} have been constructed as pure perovskites. The point defects in pure perovskites correspond to oxygen vacancies, H interstitials, substitutional acceptor dopants in the B-site (restricted to yttrium atom only) and dopant-H defect complex. The correlations between defect energetics and material structure have been systematically investigated to gain meaningful insights.

This high-throughput computational framework is based on using the density-functional theory simulation package *VASP* and code of Materials Project [35], including modules of *Pymatgen* to create input data and parse output data, *Fireworks* to launch simulations and store the execution history and IO data into *MongoDB*, *Custodian* to manage computational jobs and handle *VASP* errors, along with an MPI bundler package developed by Oak Ridge Leadership Computing Facility (OLCF) called **wraprun**. The VASP simulation outputs are not only stored on the OLCF servers, but also collected in remote MongoDB server in JavaScript Object Notation (JSON) format. The hierarchical structure of this MongoDB database can be illustrated by the diagram in Fig. 4.

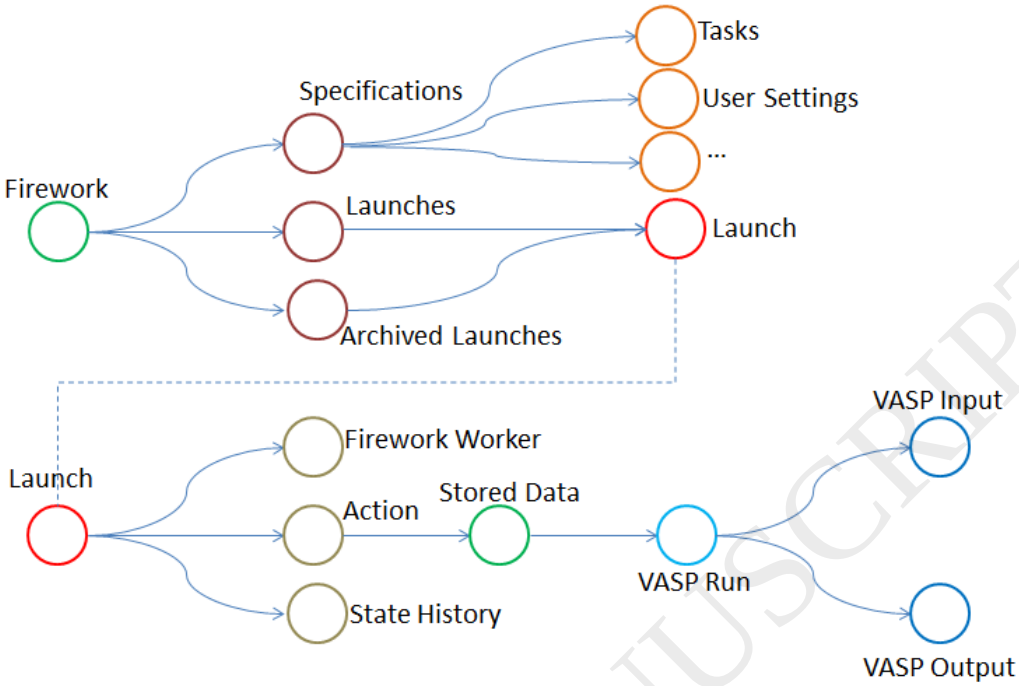


Figure 4 Hierarchical structure of stored data in the high-throughput DFT framework

The MongoDB database created in this high-throughput density functional theory (DFT) calculation framework provides full support for running all the scheduled calculation tasks, preserving their running status and archiving all the input and output files during the calculation. Two primary objects/tables, the Firework and the Launch, consist of simple related hierarchical structures in multiple layers. Firework stands for an assembly of calculation workflow. Its two major components, Specifications and Launches, store the workflow tasks to be execute, and the launched/run workflows correspondingly. The other component, Archived Launches in Firework, stores the launches (launch ID) that have been marked as archived by workflow user. Each individual execution of workflow, the Launch, is stored in a separated table that linked to Firework tree. Within each Launch, Firework Worker store the information of each physical computational node for the high-throughput DFT framework; Action element keeps the stored data of specific calculation, such as the input and output of every VASP simulation; State History element helps to identify the running state of the submitted workflow. The succinct database structure, along with the JSON formatted information block, significantly reduces the difficulty of data interoperation.

Implementation of ICME framework requires the integration of materials information in multiple length scales, including but not limited to the nano, micro, meso and macro scales. Technologies of data exchange among these different scales must be investigated to guarantee effective and efficient data interoperation. Compared with the traditional structural materials databases such as MatDB, Gen IV Materials Handbook, and MatNavi, this state-of-art Cubic Perovskites database focuses more on material properties at the atomic scale, particularly for this specific type of functional materials. Moreover, the Cubic Perovskites database is supported by MongoDB, a new generation of database and the most popular document store database in contemporary products. Regarding that, the traditional

materials databases have been mostly constructed with relational products, e.g., MS-SQL in Gen IV Materials Handbook and Oracle in MatDB. Sharp distinctions between the material scales, variations between the relational database and document-oriented database, and different data formats will unavoidably pose problems in the process of integration. The interoperation work between Gen IV Materials Handbook and Cubic Perovskites database in this section is to extend the materials database from traditional structural scale into micro scale. Meanwhile, using perovskites database as an example, this interoperation explores a different interface such as that between relational database and document-oriented database.

For the purpose of investigating data interoperation between different length scales, meanwhile extending the materials data from traditional structural scale into micro and atomic scales, developers of the Gen IV Materials Handbook system collaborated with the researchers of the cubic perovskites database to accommodate these calculated perovskites data in the Handbook system. Based on the perovskites data structure in Fig. 4, two tables, MP-Fireworks and MP-Launches, have been constructed in the Handbook system with more attributes details listed in Fig. 5. Design of these two tables in the Handbook system focuses primary on archiving the VASP running environment, calculation inputs and outputs, rather than a complete image copy of Firework and Launch tables in the cubic perovskites database. In other words, information about the calculation workflow has been maximally suppressed in the Handbook counterparts, while the remaining information not only preserves the useful calculation inputs and results, but also keeps enough original data for re-running the VASP calculations. A python algorithm has been developed as an interface to implement the data transportation. Using the MongoClient module, the algorithm connects to the cubic perovskites database in remote MongoDB server and extracts necessary datasets into formatted local files for Handbook uploading. As a demonstration of this process, Fig. 6 lists part of the interoperation python algorithm. Examples of Firework records and Launch records in Handbook system can be found in Fig. 7 and Fig. 8 respectively.

Results of this investigation have shown that data interoperation between these two distinct databases has accomplished the data interoperation at least from a document-oriented database to a relational database. Since external documents can be exported in Gen IV Materials Handbook system by using its tool, the data interoperation in the opposite way, i.e., from the relational Gen IV Materials Handbook to the document-oriented Cubic Perovskites database, can be achieved as well. One possible challenge in this opposite way of data transfer might come from the divergent document formats: a document of representing objects (JSON) and a document of markup language (XML). However, many free converters have been developed to enable smooth file transformation between these two popular formats, which will significantly alleviate the difficulty in future investigations.

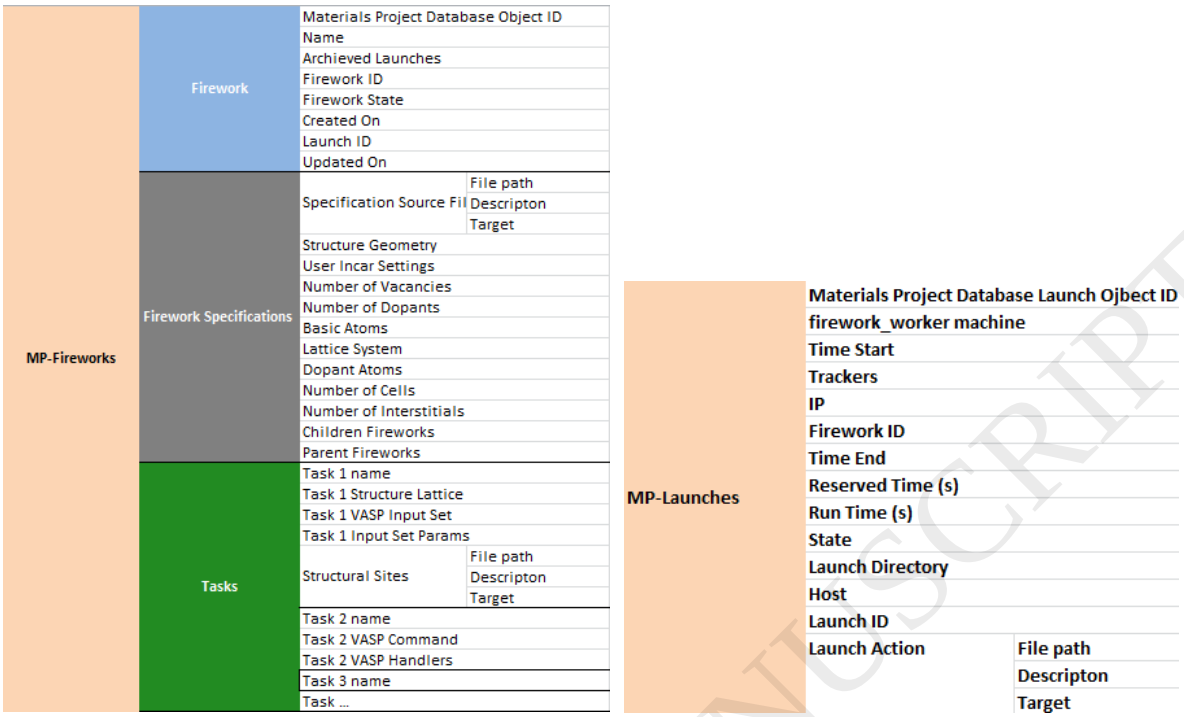


Figure 5, Fireworks and Launches schema design in Handbook system

```
from pymongo import MongoClient
import subprocess
import csv

def get_firework_details():
    conn = MongoClient("localhost", 2222)
    db = conn["MongoDB_instance"]
    db.authenticate("MongoDB_admin_user", "password")
    coll = db["fireworks"]

    for result in coll.find():
        rst_id=result["_id"]
        rst_name=result["name"]
        rst_fw_id=result["fw_id"]
        rst_state=result["state"]
        rst_archived_launches=result["archived_launches"]
        rst_created_on=result["created_on"]
        rst_updated_on=result["updated_on"]
        rst_launches=result["launches"]
        rst_spec=result["spec"]

        #spec section
        rst_spec_incar_settings=result["spec"]["_tasks"][0]["input_set_params"]["user_incar_settings"]
        rst_spec_number_vacancy = result["spec"]["noVacancies"]
        rst_spec_number_dopant = result["spec"]["noDopants"]
        rst_spec_basic_atoms = result["spec"]["basicAtoms"]
        rst_spec_lattice_system = result["spec"]["latticeSystem"]
        rst_spec_dopant_atoms = result["spec"]["dopantAtoms"]
        rst_spec_number_cells = result["spec"]["noCells"]
        rst_spec_number_interstitials = result["spec"]["noInterstitials"]
        rst_spec_parent_fws = ""; rst_spec_child_fws=""
        try:
            rst_spec_parent_fws = result["spec"]["parent_fw"]
        except Exception as ex:
            print "parent fireworks not found."
            pass
        try:
```

Figure 6, Python interoperation algorithm

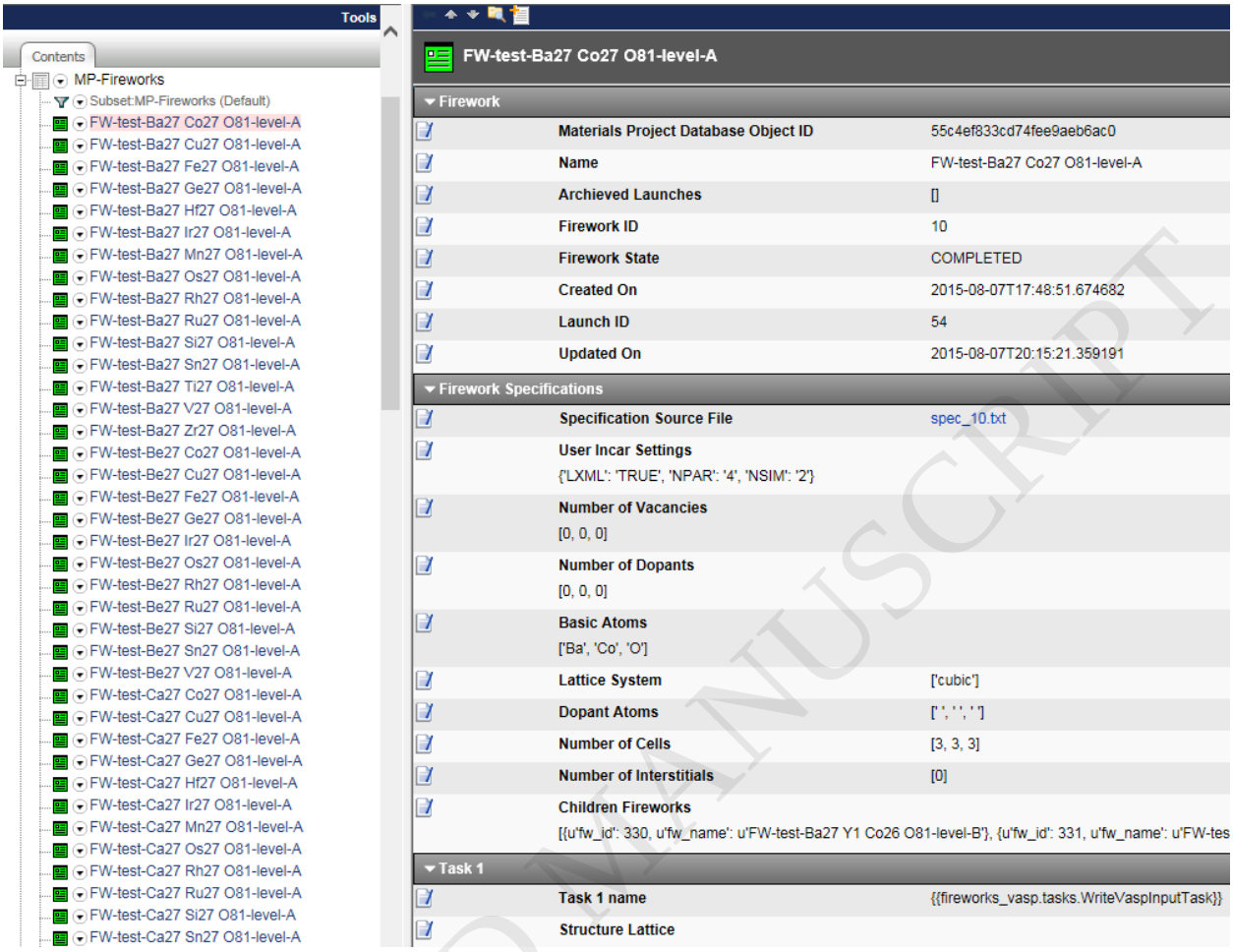


Figure 7, Firework records in Handbook system

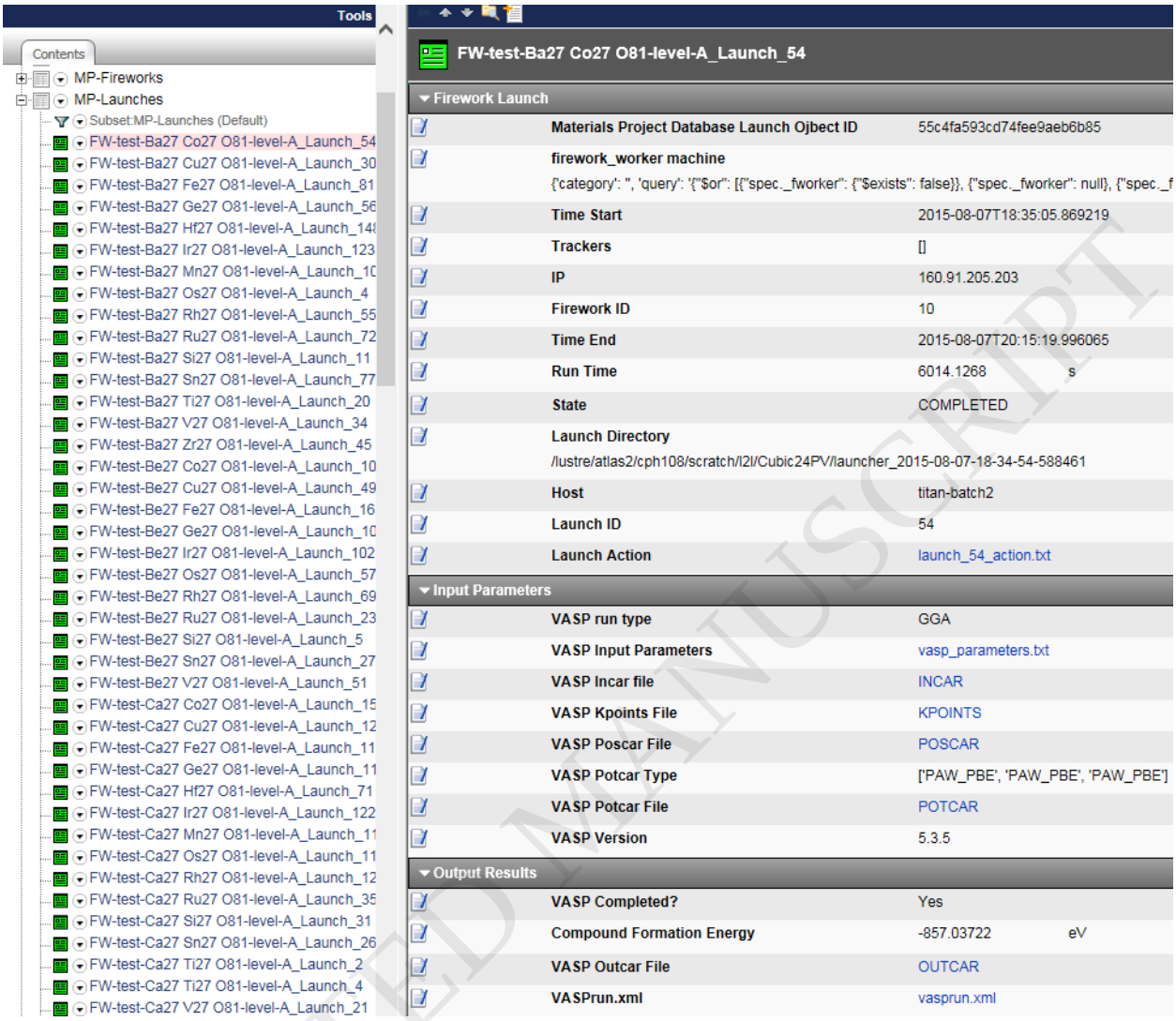


Figure 8, Launch records in Handbook system

4. Gen IV Materials Handbook and VUQ applications

CRUD, which stands for Chalk River Unidentified Deposits, has been studied extensively for its effects of corrosion and reactivity as a layer of corrosion products deposited on fuel rods in nuclear reactor cores. However, the effects of CRUD on reactor thermal-hydraulic performance have received much less attention [36]. Conducted research has been aimed primarily at understanding CRUD's composition, morphology, and deposition mechanisms, but CRUD's thermal properties and effects on fuel heat transfer and the boiling crisis are not well understood. Investigations to understand such effects, including experimental [36] and numerical simulations [37, 38] have been conducted. Significant volumes of data generated either by experiments, such as pool and flow boiling IR/HSV results on CRUD surfaces [36], or by model simulations like heat flux distributions obtained from OpenFOAM solver [38] and CFD software STAR-CCM+ [39], have piled up. During the procedure of model verification and validation iterations between model and experiments, supplemental information such as optimized

parameters, Validation & Uncertainty Qualification (VUQ) data, inevitably turn the job of data management into an additional burden. For the sake of releasing pressure from such complex and huge volume of data, The Nuclear Energy–Knowledge base for Advanced Modeling and Simulation (NE-KAMS) system has been proposed not only to manage the generated data, but also to bridge the data flow in various kinds of thermal-hydraulic studies [40-43].

For instance, one thermal-hydraulic study [38] performed the analysis on a heater infrared thermometric imaging temperature data obtained from high heat flux pool boiling and liquid film boiling experiments. With the OpenFOAM solver, heat flux distribution towards the coolant was obtained by solving transient heat conduction of heater substrate given the heater surface temperature data as boundary condition. The so-obtained heat flux data was used to validate them against the state-of-art wall boiling model developed by Shaver [37] with the assumption of micro-layer hydrodynamics. In the verification and validation workflow, the thermometric imaging temperature data, the heat flux distribution obtained by the CFD solver, and the wall boiling model have constructed the valuable digital data contents, which need to be well organized, deposited and driven to make the workflow running efficiently. Taking advantage of the mature framework developed by the Gen IV Materials Handbook system, the NK-KAMS initiates the data driven task by interoperating data between the Handbook system, the CFD software, and the Uncertainty Qualification (UQ) software involved in this thermal-hydraulic study.

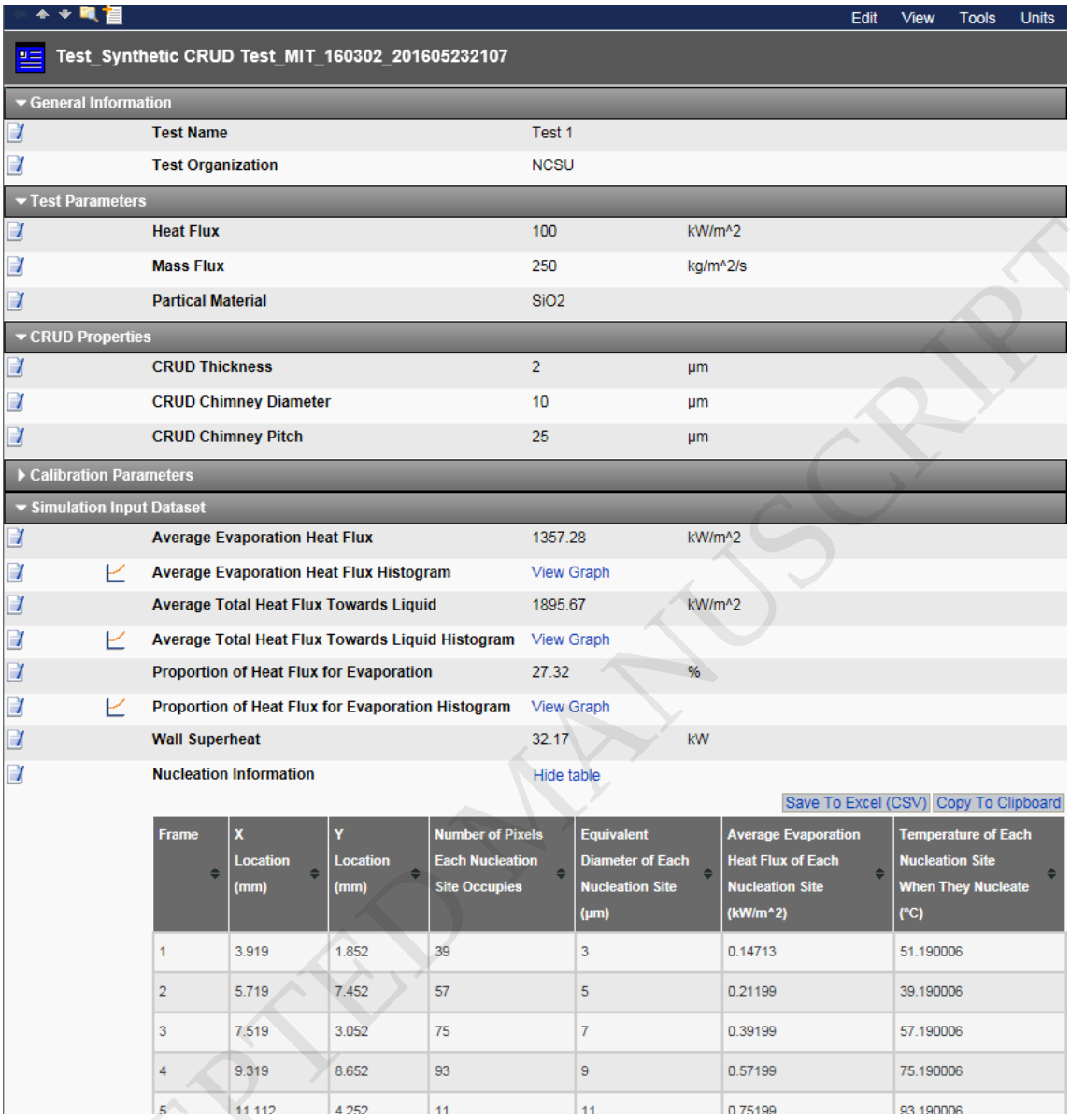
Challenges in this interoperation task come from the ecosystem differences between the two mainstream Operating Systems, Windows OS and Unix/Linux OS. The Handbook system was developed under the Windows system, using C# as developing language and MS-SQL Server as repository support at the server end. While many of the CFD software and UQ applications, particularly those free, open source ones such as OpenFOAM, DAKOTA, thrives better under the Unix/Linux ecosystems despite the fact that their Microsoft Windows versions might be available. Fortunately, the web-based Handbook system ensures that users can access data information through any web connected browser in any Operating System. Moreover, the Handbook system provides a full set of Simple Object Access Protocol (SOAP) web service, a standards-based web services access protocol originally developed by Microsoft [44, 45] that has been around for almost two decades [44] and enjoys all the benefits of long-term use.

With SOAP service APIs, high-level users or developers can access and manipulate the data information remotely through computer languages, such as Python or Matlab without opening any web browser. In this way, an uninterrupted data interoperation takes place between the information repository and application software in high efficiency by leveraging the programmable algorithms. Availing of the SOAP web services provided in Handbook system, a Python algorithm running under Linux system (Ubuntu 14.0) has been created (Fig. 9) to exchange data between local applications and remote Handbook server. In this interoperation algorithm, Python SOAP library SUDS [46] was utilized to decoding/encoding the WSDL information rendered in SOAP web service. Corresponding schema and attributes prepared in Handbook system for the needs of CRUD tests are partially listed in Fig. 10. This Python interoperation algorithm has been tested successfully at North Carolina State University to manipulate data stream from servers at ORNL. Providing that all the interoperation information can be extracted in the Python/Matlab algorithm, along with that Python/Matlab are the favorite scripting

languages used in OpenFOAM or DAKOTA, technically the obstacles of coupling the CFD simulations, UQ analysis within NE-KAMS knowledge base infrastructure have been cleared.

```
import logging
from suds.client import Client

def addSecurityHeader(client,username,password):

#Session request and authentication:
#Create the Client:
#make sure the SOAP web service is running before running this code
WSDL_BROWSE=server+"mi_servicelayer/Browse.svc?wsdl"


#client Export-----------------------------------
clientExport=Client(WSDL_EXPORT, transport=ntlm3)
clientExport.set_options(service='DataExportImpl', port='BasicHttpBinding_DataExport')
tabRef=clientExport.factory.create('{http://www.grantadesign.com/15/10/GrantaBaseTypes}PartialTableReference')
tabRef.tableName='NEKAMS-test'
nameRef1={'attributeName':"Test Name",'table':tabRef}
nameRef2={'attributeName':"Test Organization",'table':tabRef}
attRef1=clientExport.factory.create('{http://www.grantadesign.com/15/10/GrantaBaseTypes}MIAttributeReference')
attRef1.dbKey=databasekey
attRef1.name=nameRef1
attRef2=clientExport.factory.create('{http://www.grantadesign.com/15/10/GrantaBaseTypes}MIAttributeReference')
attRef2.dbKey=databasekey
attRef2.name=nameRef2


rstAttributes=clientExport.service.GetRecordAttributesByRef(result=[attRef1,attRef2,attRef3,attRef4,attRef5,at
recordReference=recordRefs)
#TabularDataValue
for d in rstAttributes.recordData[0].attributeValues.attributeValue:
    if hasattr(d,'PointDataValue'):
        strAttName=d.AttributeName
        valAtt=str(d.PointDataValue.Point[0].Value)
        strUnit=d.PointDataValue.UnitSymbol
        if(strUnit is None):
            strUnit=' '
        print("Attribute Name: " + strAttName + ", Attribute Value: " + valAtt + " " +strUnit)
    elif hasattr(d,'ShortTextDataValue'):
        strAttName=d.AttributeName
        valAtt=d.ShortTextDataValue.value
        print("Attribute Name: " + strAttName + ", Attribute Value: " + valAtt)
    elif hasattr(d,'TabularDataValue'):
        columns=d.TabularDataValue.columns.column
```

Figure 9, Python code for web service application



Edit View Tools Units

Test_Synthetic CRUD Test_MIT_160302_201605232107

General Information		
Test Name	Test 1	
Test Organization	NCSU	
Test Parameters		
Heat Flux	100	kW/m^2
Mass Flux	250	kg/m^2/s
Partical Material	SiO2	
CRUD Properties		
CRUD Thickness	2	μm
CRUD Chimney Diameter	10	μm
CRUD Chimney Pitch	25	μm
Calibration Parameters		
Simulation Input Dataset		
Average Evaporation Heat Flux	1357.28	kW/m^2
Average Evaporation Heat Flux Histogram	View Graph	
Average Total Heat Flux Towards Liquid	1895.67	kW/m^2
Average Total Heat Flux Towards Liquid Histogram	View Graph	
Proportion of Heat Flux for Evaporation	27.32	%
Proportion of Heat Flux for Evaporation Histogram	View Graph	
Wall Superheat	32.17	kW
Nucleation Information	Hide table	

Save To Excel (CSV) Copy To Clipboard

Frame	X Location (mm)	Y Location (mm)	Number of Pixels Each Nucleation Site Occupies	Equivalent Diameter of Each Nucleation Site (μm)	Average Evaporation Heat Flux of Each Nucleation Site (kW/m^2)	Temperature of Each Nucleation Site When They Nucleate (°C)
1	3.919	1.852	39	3	0.14713	51.190006
2	5.719	7.452	57	5	0.21199	39.190006
3	7.519	3.052	75	7	0.39199	57.190006
4	9.319	8.652	93	9	0.57199	75.190006
5	11.112	4.252	11	11	0.75199	93.190006

Figure 10, Data schema and record initiated in Handbook system

5. Discussion and Conclusions

Databases are one of the three core components of ICME. The multi-scale simulation, the processing simulation and the materials testing always produce giant volumes of data. Databases not only store and manage massive data sets, but more importantly drives the information flow in the ICME ecosystem to take the system's maximum advantages. After decades of materials development, useful materials information has been processed and stored in diverse sources, e.g., distributed materials database systems. The data exchange among distinct database systems and different data consuming applications would maximally share the contemporary database contents, saving time and cost in the ICME development.

So far, no unique protocol has been applied for the data exchange in ICME system. In this ecosystem, different applications or database sources export information in diversified forms, which create huge difficulties in data communication and data consumption. This article presents a practical solution utilizing XML and XSLT, specialized markup languages in Information Technology, for the data exchange between two database systems. The other two examples introduced in this article utilize Python script or SOAP web service for the data communication between database and remote applications.

Complying with specific ISO test standard (ISO-6892-1:2009), the tensile test records from MatDB system and Gen IV Materials Handbook system are exported into external XML files. After further file processing with the automated XSLT transformation, the exported XML files fit each other's importing requirement. The ISO testing standard encapsulates accepted procedure, terminology, and outputs, all of which define the data that is generated when the testing standard is followed [29]. The utilization of a data format that is compliant with an ISO testing standard has proved particularly effective in enabling systems integration because transfer of data according to accepted standards offers the possibility to reduce data collection and transfer overheads and to increase the reliability of data. The implementation of this specific case has created an efficient solution which builds the interface of exchanging tremendous amount of data automatically between these two database systems. With more application of this solution in other equivalent modules, the overall interoperability in system level will finally be achieved.

Implementation of ICME includes not only development of advanced materials computational tools, but also the integration of these tools and the data or databases get involved. Effective data interfaces are needed to guarantee efficient data flow among the computational tools and databases in the integration system. Data transfer through secured network provides a convenient way in physical layer for ICME implementation, while web services like SOAP or RESTful boost the implementation of automated online materials transfer in application layer. As a most popular markup language to store and transport data, XML is one of the best choices to build data interface, especially for the online data exchanging. It's also widely supported by most software in exporting data to external documents. The SOAP web service has been developed for a long time, to fully support the online XML data exchanging across different software and different operating systems through the internet. In the case of data interoperation between the traditional relational database, the Gen IV Materials Handbook, and the state-of-art document-oriented database, the Cubic Perovskites, the interface algorithm in Python codes demonstrates its flexibility and compatibility. RESTful provides a lighter weight alternative in web information exchanging, which has attracted more attentions in recent years. Implementation case in this article utilized SOAP web service only because it fit with existing applications best to accomplish automated materials data exchange. However, utilization of both SOAP and RESTful services should be considered for the comprehensive development of ICME.

Through the implementation instances, it can be concluded that in building effective ICME applications, 1) standards-compliant interface can improve the exchange efficiency; 2) popular web service like SOAP enables automated online materials data transfer; 3) customized data interoperation scripts (like examples in Fig. 6 and 9) along with the methodology behind can provide flexibility and productivity in

building ICME data interoperation interface. With all these advantages, the solutions succeeded in this article can be extended as some basic schemas for the information exchange inside ICME ecosystem.

Acknowledgement

The authors would like to thank Peter Hähner, Hans-Helmut Over and Timothy Austin of the European Commission Joint Research Centre Institute for Energy and Transport for their collaboration in the interoperation between MatDB and Gen IV Materials Handbook, thank Janakiraman Balachandran and Panchapakesan Ganesh of the Center for Nanophase Materials Sciences (CNMS) at Oak Ridge National Laboratory (ORNL) for their sharing of Cubic Perovskites Database, and thank Yang Liu and Nam Dinh of the North Carolina State University for their thermal-hydraulic data.

This work is sponsored by both the European Commission and the U.S. Department of Energy, Office of Nuclear Energy Science and Technology under contract DE-AC05-00OR22725 with Oak Ridge National Laboratory, managed by UT-Battelle, LLC.

References

1. J. Allison, Integrated computational Materials Engineering: A Perspective on Progress and Future Steps, JOM, Vol. 63 No. 4, April 2011

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

3. W. Hunt, R. Brindle and S. Henry, Challenges and Approaches in Materials Data Management for ICME, 2nd World Congress on Integrated Computational Materials Engineering, July 2013

4. G. Olson, Structuring the Genome: Fundamental Materials Databases, 2nd World Congress on Integrated Computational Materials Engineering, July 2013

5. W. Ren, Consideration of Ecosystem for Integrated Computational Materials Engineering, 2nd World Congress on Integrated Computational Materials Engineering, July 2013

6. W. Marsden, B. Cope, Tools to Support the Flow of Traceable Materials Information Needed by ICME, 2nd World Congress on Integrated Computational Materials Engineering, July 2013

7. MatNavi, NIMS Materials Database, http://mits.nims.go.jp/index_en.html. Accessed 10 May, 2017.

8. MatDB, https://odin.jrc.ec.europa.eu/alcor/Main.jsp. Accessed 10 May, 2017.

9. W. Ren, Gen IV materials handbook functionalities and operation (4A)—handbook version 4.0, 2013, ORNL/TM-2013406_4A

10. Kalidindi, S.R., Brough, D.B., Li, S., Cecen, A., Blekh, A.L., Congo, F.Y.P. and Campbell, C., MRS Bull., 41(8), pp. 596-602 (2016).

11. B. Blaiszik, K. Chard, J. Pruyne, R. Ananthakrishnan, S. Tuecke and I. Foster, JOM, 68: 2045-2052 (2016).

12. B. Puchala, G. Tarcea, E.A. Marquis, M. Hedstrom, H.V. Jagadish and J.E. Allison, JOM, 68: 2035-2044 (2016).

13. J. O'Mara, B. Meredig and K. Michel, JOM, 68: 2031-2034 (2016).

14. A. Dima, S. Bhaskarla, C. Becker, M. Brady, C. Campbell, P. Dessauw, R. Hanisch, U. Kattner, K. Kroenlein, M. Newrock, A. Peskin, R. Plante, S.-Y. Li, P.-F. Rigodiat, G.S. Amaral, Z. Trautt, X. Schmitt, J. Warren and S. Youssef, JOM, 68: 2053-2064 (2016).

15. M.D. Jacobsen, J.R. Fourman, K.M. Porter, E.A. Wirrig, M.D. Benedict, B.J. Foster and C.H. Ward, Integrating Materials and Manufacturing Innovation, 5: 12, (2016).

16. HUBzero, https://hubzero.org/. Accessed 24 Feb 2017.

17. NanoHub, https://nanohub.org/. Accessed 24 Feb 2017.

18. Faical Yannick Palingwende Congo, Proc. of the 14th Python in Science Conf. (SCIPY 2015)

19. RDA/CODATA Materials Data, Infrastructure & Interoperability IG, https://www.rd-alliance.org/groups/rdacodata-materials-data-infrastructure-interoperability-ig.html. Accessed 24 Feb 2017.

20. Steven M. Arnold, Frederic A. Jr. Holland, Brett A. Bednarcyk, Science and Technology Forum and Exposition (SciTech 2014), National Harbor, Maryland, 13-17 January, 2014.

21. Lynne Robinson, JOM, Volume 66, Issue 8, pp 1356–1359 (2014).

22. Janet K Allen, Farrokh Mistree, Jitesh Panchal, BP Gautham, Amarendra Singh, Screedhar Reddy, Nagesh Kulkarni, and Prabhash Kumar, Second World Congress on Integrated Computational Materials Engineering, Salt Lake City, Utah, 7-11 July, 2013

23. Prasenjit Das, Raghavendra R Yeddula, Sreedhar Reddy, 2nd Modelling Symposium (ModSym 2016), Goa, India, February 18, 2016

24. A.W.A. Konter, H. Farivar, J. Post and U. Prahl, JOM, Vol. 68, No.1 (2016).

25. Alan A. Luo, CALPHAD, Volume 50, Pages 6–22 (2015).

26. Michael A Groeber, Michael A Jackson, Integrating Materials and Manufacturing Innovation, 3:5 (2014).

27. D. Howe, B. Goodlet, J. Weaver and G. Spanos, JOM, Volume 68, Issue 5, pp 1378-1384 (2016).

28. G. J. Schmitz, A. Engstrom, R. Bernhardt, U. Prahl, L. Adam, J. Seyfarth, M. Apel, C. Agelet de Saracibar, P. Korzhavyi, J. Ågren, B. Patzak, “Software Solutions for ICME,” JOM, January 2016, Volume 68, Issue 1, pp 70-76

29. Lin, L., Austin, T., and Ren, W., "Interoperability of Materials Database Systems in Support of Nuclear Energy Development and Potential Applications for Fuel Cell Material Selection," Materials Performance and Characterization, Vol. 4, No. 1, 2015, pp. 115-130.

30. T.S.P. Austin and H.H. Over, MatDB Online - A Standards-Based System for Preserving, Managing, and Exchanging Engineering Materials Test Data, Data Science Journal, Vol. 11, 23 November 2012

31. H.H. Over, W. Dietz, The Web-enabled Database of JRC-EC, a Useful Tool for Managing European Gen IV Materials Data, *Journal of Nuclear Materials*, Vol. 376, Iss. 3, 15 June 2008, pp. 346-352

32. C. H. Newton, ASTM Manual on the Building of Materials Databases, ASTM Manual Series: MNL 19 (Philadelphia PA 1993)

33. T. Ojala and H.H. Over, Approaches in Using MatML as a Common Language for Materials Data Exchange, Data Science Journal, Vol. 7, 4 November 2008

34. Janakiraman Balachandran, Lianshan Lin, Jilai Ding, Yongqiang Cheng, Raymond R. Unocic, Gabriel Veith, Weiju Ren, Craig Bridges, Panchapakesan Ganesh, Proton Transport in Solid Electrolytes - Insights from High Throughput Computations and Data Analysis, 2016 MRS Spring Meeting & Exhibit, Phoenix, Arizona, March 28-April 1, 2016.

35. Materials Project, https://github.com/materialsproject. Accessed 10 May, 2017.

36. C. Coyle, J. Buongiorno, T. McKrell, "Synthesis of CRUD and its Effects on Pool and Subcooled Flow Boiling", MIT Report, CASL-U-2015-0068-000, USA (2015).

37. Dillon R. Shaver, Development of a mechanistic model for forced convection subcooled boiling, ProQuest Dissertations and Theses; Thesis (Ph.D.) -- Rensselaer Polytechnic Institute, 2014.

38. Y. Liu and N. Dinh, Analysis of heat transfer under high heat flux nucleate boiling conditions, Kerntechnik, Vol. 81, No. 3, pp. 308-314 (2016).

39. Lindsey Anne Gilman, Development of a General Purpose Subgrid Wall Boiling Model from Improved Physical Understanding for Use in Computational Fluid Dynamics, PhD Thesis, Massachusetts Institute of Technology, June 2014.

40. Weiju Ren, Status and Near-Future Activities for NE-KAMS Development to Support Verification and Validation of Advanced Modeling and Simulation, ASME 2015 Verification and Validation Symposium, Las Vegas Nevada, May 11-15, 2015

41. Hyung Lee, Kimberlyn C. Mousseau, Richard W. Johnson, and Weiju Ren, NE-KAMS Demonstration Final Technical Report, Nuclear Energy Advanced Modeling and Simulation Program, U. S. Department of Energy, September 30, 2012.

42. Kimberlyn C. Mousseau, Weiju Ren, Hyung Lee, and Greg Weirs, Nuclear Energy – Knowledge base for Advanced Modeling and Simulation Pathways Demonstration, NEAMS Program NE-KAMS Review Meeting, Department of Energy Headquarters, Germantown, October 24, 2012.

43. V. Gregory Weirs, An Initial Review of NE-KAMS Validation Data Quality Standards, Sandia Report, Sandia National Laboratories, Oct 2012.

44. SOAP Wiki: https://en.wikipedia.org/wiki/SOAP. Accessed 5/1/2017.

45. Understanding SOAP, MSDN, https://msdn.microsoft.com/en-us/library/ms995800.aspx. Accessed 5/1/2017.

46. SUDS, https://bitbucket.org/jurko/suds/wiki/Original%20Documentation. Accessed 5/1/2017.

47. Wei Xiong and Gregory B Olson, Cybermaterials: Materials by Design and Accelerated insertion of Materials, Computational Materials (2016) 2, 15009, February 12, 2016.

48. David L. McDowell and Surya R. Kalidindi, The materials innovation ecosystem: A key enabler for the Materials Genome Initiative, MRS Bulletin, Vol 41, April 2016.