Data Analysis Environment for Materials Science and Engineering Integrating Heterogeneous Data Resources

Toshihiro Ashino [1], Nobutaka Nishikawa [2], and Takuya Kadohira[3]

[1] Toyo University, 5-28-20 Hakusan, Bunkyo-ku, Tokyo 112-8606, Japan
ashino@acm.org
[2] Mizuho Information & Research Institute, Inc. 2-3 Kanda-Nishikicho, Chiyoda-ku, Tokyo 101-8443, Japan
[3] National Institute for Materials Science, 1-2-1 Sengen, Tsukuba, Ibaraki, 305-0047, Japan

Abstract. Materials performance analysis requires to integrate many heterogeneous data and information resources, experimental data, empirical/theoretical models and computational simulations. It means data analysis platform for materials science and engineering should provide many functionalities, e.g., data retrieval, processing, statistical analysis, symbolic mathematics, visualization and scripting capabilities to store the typical data analysis process and also, these heterogeneous data resources should be accessed unified way. Scripting language Python provides many of these capabilities with additional software modules and widely applied to interactive/non-interactive data processing environment. In this paper, a prototype design and implementation of data analysis environment for materials science and engineering is presented.

Keywords: virtual research environment, materials integration, materials ontology, semantic web, heterogeneous data integration

1 Introduction

In many research area, data intensive research, so called the Fourth Paradigm [1], have been increasing its importance. In materials science and engineering, there is a long tradition developing computerized materials property databases [2, 3]. But materials experiment requires huge cost and high skill, materials represent wide variation of properties, there are various measurement methods and substances, data intensive approach is delayed to be introduced into materials design process.

But advancement of computer simulation technology and new measurement method presented a possibility to obtain huge amount of data in this field. It enables to evaluate materials properties such as physical properties and long term performance with minimum experiment, relatively low cost and short period, furthermore, enables to predict materials performance without real experiment [4–6].

One of the important application area is to develop software platform for high throughput computational approach for materials design focused on functional mate-

rials which performances are directly reflect micro-scale physical properties [7, 8]. However, in case of structural materials performances prediction, e.g. creep rupture property, different scales and complexed interactions of physical phenomenon affect the total performance, it requires to integrate heterogeneous data and models.

This approach is called ICME (Integrated Computational Materials Engineering) [9]. In Japan, SIP-MI (Strategic Innovation Promotion Program: Materials Integration) is a project to implement ICME concept. Information platform for MI is required to handle and integrate many kind of information resources, such as experimental data, simulation modules and mathematical equations. Semantic description of data, relationships among data and attributes of data are essential in order to integrate these heterogeneous information.

We applied the Semantic Web framework to this application. It provides several machine readable semantic description standards, XML Schema [10], RDF (Resource Description Framework)/SPARQL (SPARQL Protocol and RDF Query Language) [11, 12], OWL (Web Ontology Language) [13] and OpenMath [14]. MI prototype data platform which can handle these data formats and enables to describe workflows of materials data processing has been developed.

2 Design and Implementation of the Prototype

The prototype system is based on a mathematical system, SageMath [15], which is an open source project integrates many open source mathematical systems, SciPy, R and others. It is based on Python programming environment and this means, various software modules developed for Python can be used in this system and it is easy to develop original data processing modules for this data processing environment.

Fig. 1 shows the design of the prototype system. In order to achieve flexible data management, since it should manage continuously evolving materials measurement and new materials data, metadata, which describes the structure of database is stored in Apache/Jena Fuseki SPARQL endpoint as RDF files. RDF provides conceptual description on the data resources and it is retrieved by using SPARAL query language.

Metadata which describes experimental data and mathematical equations, target materials, equation names, target property, application conditions and link to data and equation body, are written in RDF for retrieval by SPARQL. Sample experimental databases is stored as XML (Extensible Markup Language) documents, they can be accessed by their URI's listed in RDF files. Equation bodies are also stored as XML documents which written in OpenMath semantic representation of mathematics, which provides rich vocabularies contain many operators and mathematical functions [16].

Python modules XML, RDFlib, SPARQLWrapper and py-openmath are incorporated into SageMath symbolic-math environment and original OpenMath parser have been developed for this prototype. Metadata which describes experimental data and mathematical equations, target materials, equation names, target property, application conditions and link to data and equation body, are written in RDF for retrieval by

SPARQL. Materials Ontology written in OWL is managed by the same SPARQL endpoint.

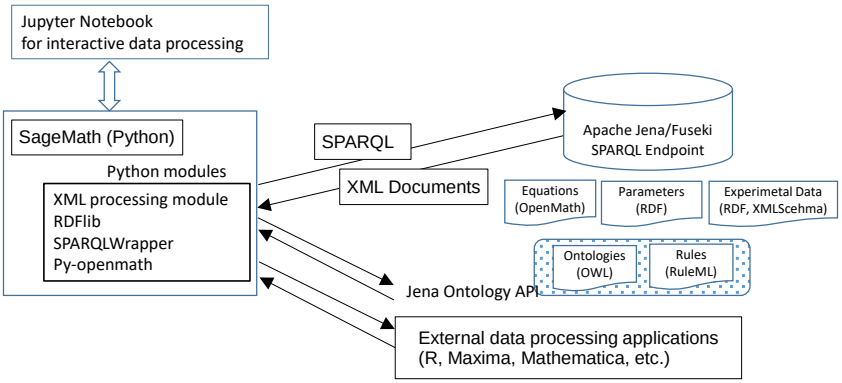

Fig. 1. Concept of the prototype system for materials data processing

Fig. 2 shows examples of metadata description for experimental data (a) and equation (b) in RDF. In current sample database, tags are selected from Dublin Core tag set defined in order to describe metadata [17], but there are many tag sets which defined to represent data meanings and any of them can be added into these RDF data anytime.

Experimental datasets and equation bodies are divided from RDF metadata file. RDF file contains URI's (Uniform Resource Identifier) indicate datasets and equations, since such files may have written in different data formats like XML Schema and OpenMath. Data retrieval requires two-steps, at first, find a RDF description by SPARQL and second, traverse the URI which is indicated by <dc:relation> tags.

Vocabularies used in database, property names, material names, units for measured values and other keywords are selected from extended Materials Ontology [18]. It intended to realize uniform data retrieval on heterogeneous data resources, in this case, experimental data and equation library stored in different RDF documents. In current prototype, words are selected manually from the ontology as a common vocabulary.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:mi="http://www.codata.jp:8080/mi/mat-ontology.owl#">
 <rdf:Description rdf:about="http://www.codata.jp:8080/mi/9cr.mga.haz.creep.xml">

  <dc:title>Mod.9Cr-1Mo MgT Creep Test</dc:title>
  <dc:description>creep</dc:description>
  <dc:subject>mi:Creep_Test</dc:subject>
  <dc:description>Mod.9Cr-1Mo MgA HAZ Creep Test</dc:description>
  <dc:type>mi:MgA_HAZ</dc:type>

  <dc:relation>http://www.codata.jp:8080/mi/9cr.mga.haz.creep.550.200.xml</dc:relation>
  <dc:relation>http://www.codata.jp:8080/mi/9cr.mga.haz.creep.550.190.xml</dc:relation>
  <dc:relation>http://www.codata.jp:8080/mi/9cr.mga.haz.creep.550.170.xml</dc:relation>
  <dc:relation>http://www.codata.jp:8080/mi/9cr.mga.haz.creep.600.140.xml</dc:relation>
  <dc:relation>http://www.codata.jp:8080/mi/9cr.mga.haz.creep.600.120.xml</dc:relation>
  <dc:relation>http://www.codata.jp:8080/mi/9cr.mga.haz.creep.600.100.xml</dc:relation>
```

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
 <rdf:Description rdf:about="http://www.codata.jp:8080/mi/creep.nortom.xml">

  <dc:title>Creep Equation Norton</dc:title>
  <dc:subject>mi:Norton</dc:subject>
  <dc:description>Creep Equation Norton</dc:description>
  <dc:type>mi:Norton</dc:type>

  <dc:relation>http://www.codata.jp:8080/mi/creep.norton.openmath.xml</dc:relation>

  <dc:source>http://dx.doi.org/*****</dc:source>

 </rdf:Description>
</rdf:RDF>
```

Fig. 2. Metadata description in RDF for (a) experimental data and (b) constitution equation. Data and equations are stored in XML files pointed by URI's

3 An Example Materials Performance Analysis Workflow in Python

One of the typical materials data processing workflow, creep data analysis is displayed in Fig. 3. Workflows can be written in Python scripting language in the prototype, it provides quite flexible and extensible description. 1st, relevant creep experimental data is retrieved from database with SPARQL. Results are obtained in XML documents and they are transformed into appropriate format for further processing by the XPath functions of Python XML processing module. XML data format stored in database is defined in this project locally, but it should be standardized for test method or property in XML Schema.

2nd, appropriate equation, in this case Norton equation, constitution equation for creep behavior is selected by its metadata written in RDF. The metadata contains a URI which points semantic representation of the equation in OpenMath. It can be parsed and converted into the corresponding input format required by specified data processing package, e.g. R, SciPy and other packages which is integrated to SageMath.

In the package, non-linear least square method is applied to the equation with the retrieved experimental data set. Obtained parameter values, in this case A and n, are written into RDF format, added appropriate metadata, e.g. link to corresponding experimental data, equation and version of software package, and stored into the database for further utilization in MI software modules.

This workflow can be stored as Python script and also, all functions can be used in interactive programming environment Jupyter notebook. This script has properly worked and proved the extensibility and flexibility of this system.

4 Discussions

There are many trials to develop ontology and integrate data with ontology [19–22]. Ontology can be used a fundamental dictionary for data integration. But in order to integrate heterogeneous information resources, all description of these resources should be based on common ontology or be mapped to the correspondence of ontology. This work is done manually, it requires continuous efforts to standardize and disseminate ontology, and also support system to select vocabulary with ontology reasoner.

Materials ontology has been extended to contain some concepts which relate to creep performance evaluation. In this prototype, ontology written in OWL can be accessed via Apache/Jena API, we are now testing utilization of reasoner in data retrieval and rule based data analysis with this functionality.

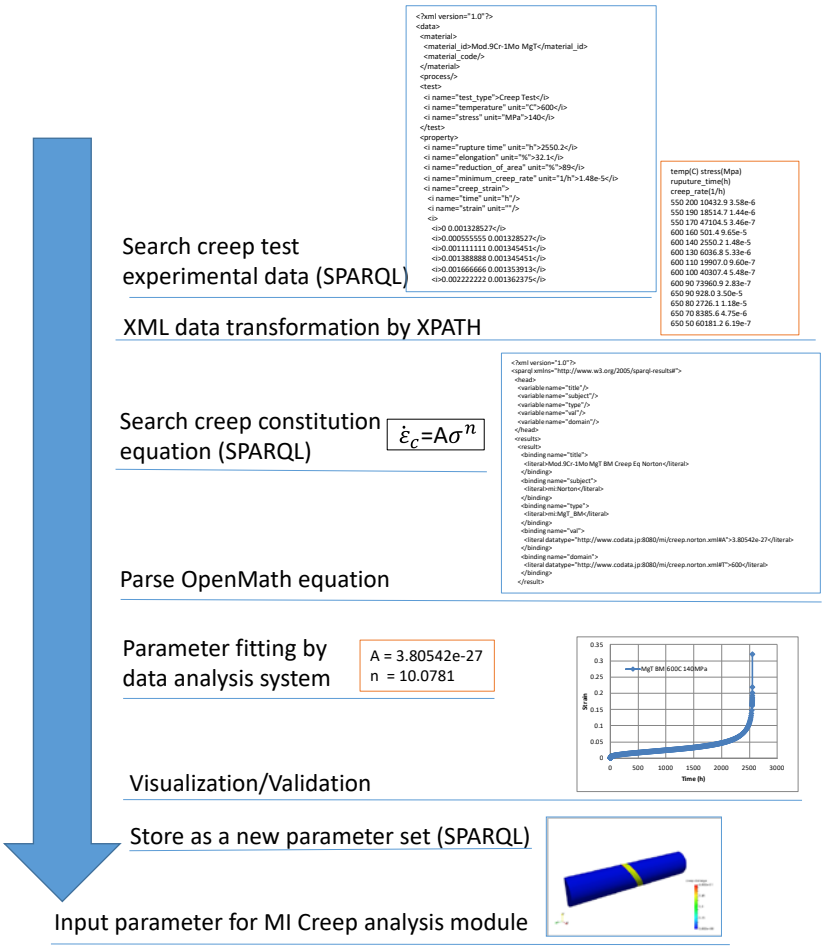

Fig. 3. A creep data processing workflow and corresponding operation on the system

5 Conclusion

Prototype of data analysis environment which has capability integrating heterogeneous materials information resources have been developed based on Python programming language and the design have been verified by sample database and script. RDF metadata representation for materials experimental data and mathematical equations is defined and tested for further development of MI system.

Acknowledgments

This work was supported by Council for Science, Technology and Innovation (CSTI), Cross-ministerial Strategic Innovation Promotion Program (SIP), Structural Materials for Innovation" (Funding by JST).

References

1. Hey, T., Tansley, S., and Tolle, K.M.: The fourth paradigm: data-intensive scientific discovery (Microsoft Research, Redmond, 1969).
2. Rumble Jr., J.R.: Integr. Mater. Manuf. Innov. (6), 172–186 (2017).
3. Austin, T.: Mater. Discov. **3**, 1–12 (2016).
4. Curtarolo, S., Hart, G.L.W., Nardelli, M.B., Mingo, N., Sanvito, S., and Levy, O.: Nature Mater. **20**, 191–201 (2013).
5. Broderick, S.R., Santhanam, G.R., and Rajan, K.: JOM **68**, 2109–2115 (2016).
6. Editorial: Scripta Mater. **70**, 1–2 (2014).
7. Ong, S.P., Richards, W.D., Jain, A., Hautier, G., Kocher, M., Cholia, S., Gunter, D., Chevrier, V.L., Persson, K.A., and Ceder, G.: Comp. Mater. Sci. **68**, 314–319 (2013).
8. Kalidindi, S.R., Niezgoda, S., Landi, G., and Fast, T.: Comp., Mater. and Cont. **17,** 103–125 (2010).
9. National Research Council: Integrated Computational Materials Engineering: A Transformational Discipline for Improved Competitiveness and National Security (The National Academies Press, Washington, DC. 2008).
10. W3C: https://www.w3.org/standards/xml/schema, last accessed 2019/5/5
11. W3C: https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/, last accessed 2019/5/5
12. W3C: https://www.w3.org/TR/2013/REC-sparql11-overview-20130321/, last accessed 2019/5/5
13. W3C: https://www.w3.org/TR/2012/REC-owl2-overview-20121211/, last accessed 2019/5/5
14. OpenMath Society: https://www.openmath.org/standard/om20-2017-07-22/, last accessed 2019/5/5
15. SageMath, the Sage Mathematics Software System (Version 8.0), The Sage Developers, 2017, https://www.sagemath.org. last accessed 2019/5/5
16. Ashino, T. and Yamashita, Y.: Data Sci. J. **11**, ASMD17-ASMD21 (2012).
17. Dublin Core Initiative: http://dublincore.org/, last accessed 2019/5/5
18. Ashino, T.: Data Sci. J. **9**, 54–61 (2010).
19. Zhao, S. and Qian, Q.: AIP Advances **7**, 105325 (2017).
20. LeBlanc, E., Balduccini, M., and Regli, W.C.: AAAI-14 Workshop (AAAI, Quebec, 2014) 39–42.
21. Madalli, D., Sulochana, A., and Singh, A.K.: Data Technol and Appl. **50**, 103–117 (2016).
22. Remolona, M.F.M., Conway, M.F., Balasubramanian, S., Fan, L., Feng, Z., Gu, T., Kim, H., Nirantar, P.M., Panda, S., Ranabothu, N.R., Rastogi, N., and Venkatasubramanian, V.: Comp. and Chem. Eng. **107**, 49–60 (2017).