Developing Ontologies and Web-based Data Management System for Additive Manufacturing Processes

Congrui Li
Tetherless World Constellation
Rensselaer Polytechnic Institute
110 8th St
Troy, NY 12180
lic10@rpi.edu

Bryan Chu*
Department of MANE
Rensselaer Polytechnic Institute
110 8th St
Troy, NY 12180
bchu@sandia.gov

Charles Parslow
Department of MANE
Rensselaer Polytechnic Institute
110 8th St
Troy, NY 12180
parslc@rpi.edu

Johnson Samuel
Department of MANE
Rensselaer Polytechnic Institute
110 8th St
Troy, NY 12180
samuej2@rpi.edu

Peter Fox
Tetherless World Constellation
Rensselaer Polytechnic Institute
110 8th St
Troy, NY 12180
pfox@cs.rpi.edu

ABSTRACT

This article introduces a project in which researchers from across different disciplines including semantic web, materials science, manufacturing engineering, and software engineering collaborate to develop ontologies and a data management system for additive manufacturing processes.

KEYWORDS

ontology, additive manufacturing, information modeling, data management

1 INTRODUCTION

The field of additive manufacturing (AM), also known as 3D printing, is growing at an incredible pace. One issue that is slowing its progress is the difficulty in finding data for using new equipment and materials as well as the design of new processes. This often necessitates lengthy and expensive periods of trial and error. Worse yet, intellectual property constraints often lead to these parameters being kept proprietary to prevent competition, adding to the difficulties.

The creation of an ontology modeling AM processes will help solve both the issue of finding useful data and using the data to develop and improve AM processes. Creating a unified ontological framework will reduce the fragmentation caused by the multitude of proprietary formats, making searching for useful data easier. Leveraging the power of semantic web and web-based software applications will aid in validating hypothesized relations, as well as potentially revealing hidden ones.

*Currently at Sandia National Laboratories.

WebSci '17, June 25–28, 2017, Troy, NY, USA
 ISBN 978-1-4503-4896-6/17/06.
DOI: http://dx.doi.org/10.1145/3091478.3098863

2 ONTOLOGY DESIGN

A number of existing ontologies were used in the creation of the Additive Manufacturing Processing Ontology (AMPO):

- PROV Ontology (PROV-O). This provides a set of classes, properties, and restrictions that can be used to represent and interchange provenance information generated in different systems and under different contexts. It can be specialized to create new classes and properties to model provenance information for different applications and domains.[1]
- Semanticscience Integrated Ontology (SIO). This ontology is used to facilitate biomedical knowledge discovery. SIO features a simple upper level comprised of essential types and relations for the rich description of arbitrary objects, processes, and their attributes.[2]
- Quantities, Units, Dimensions and Data Types Ontologies (QUDT). This provides semantic specifications for units of measure, quantity kind, dimensions, and data types.[3]

Figure 1 shows a schematic representation of major classes in AMPO. Each line connecting two boxes represents an object property. This is just an oversimplified representation of a conceptual map (Cmap) to give readers a brief idea.[1]

Here are some key points we would like to mention about AMPO:

(a) The class hierarchies of AMPO are connected with the upper level ontology, SIO. Each class in AMPO is a subclass of one of these three SIO classes: sio:Attribute, sio:Object, or sio:Process.

(b) Both ampo:Step and ampo:Process are subclasses of sio:Process. They are disjoint classes. Their definitions are used to differentiate them. ampo:Process must contain more than one step. ampo:Step is atomic and indivisible. Additionally, ampo:Process may be a component of a higher level ampo:Process. ampo:happensDirectlyBefore defines the sequential relationship between two processes, or two steps, or a process and a step.

(c) We define ampo:Process and ampo:Step as subclasses of prov:Activity. In order to record the timestamps of ampo:Process and ampo:Step instances, we use the properties prov:startedAtTime

[1]Due to space limit, the full AMPO Cmaps could not be included. Readers can access them at: https://goo.gl/WvgZwa

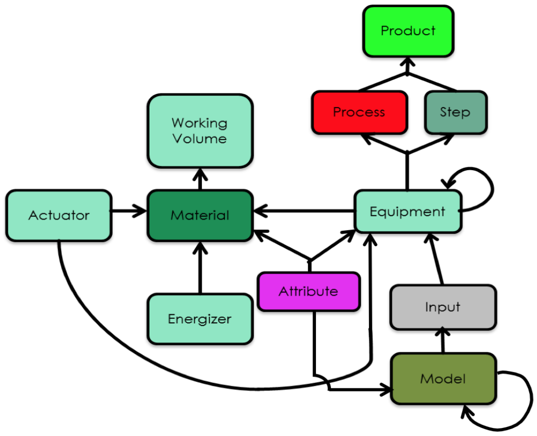

Figure 1: Simplified graph representation of AMPO.

and prov:endedAtTime from PROV-O. The above assertion also allows AMPO to connect ampo:Process and ampo:Step with prov:Person and prov:Role.

(d) ampo:QuantitativeInput and ampo:QuantitativeAttribute are subclasses of qudt:Quantity since an instance of these two classes is a quantitative measurement along with a unit.

(e) ampo:Process is defined as a subclass of both prov:Activity and sio:Entity. This makes ampo:Process a subclass of two classes which belong to two different ontologies. This is valid and no conflict would occur.

(f) ampo:QuantitativeInput and ampo:DynamicQuantitativeAttribute have a datatype property ampo:downloadURL. Instances of these two classes are usually time series data which contain different values at multiple timestamps. Our best practice is to access each time series as a .csv file through a cloud storage application such as Dropbox or Google Drive, while only providing the access URL, i.e. the metadata, in the use case RDF data.

3 SYSTEM ARCHITECTURE

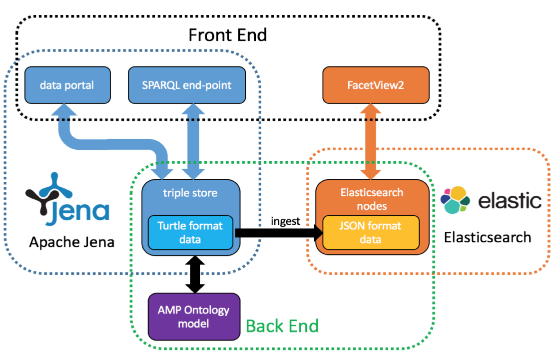

Figure 2: Data management system architecture.

Figure 2 shows the detailed architecture of our data management system with data storage, search, and browsing features for AMPO use cases.

Apache Jena Fuseki is a SPARQL server, which contains a built-in data portal user interface (UI), a SPARQL end-point, as well as triple stores. Fuseki provides REST-style interaction with the RDF data which allows the triples it contains to be exposed as a SPARQL end-point accessible over HTTP.[4] Both the AMPO ontology and use cases are coded in machine readable RDF Turtle (.ttl) format. We could load all the RDF data through the frontend UI. The ontology Turtle file needs to be loaded first since it serves as the fundamental framework for all the use cases, similar to the schema in a relational database. Elasticsearch is a distributed, RESTful search and analytics engine which uses standard RESTful APIs and JSON.[5]

Python scripts were written to automatically extract, transform, and load (ETL) the relevant data from Fuseki into an Elasticsearch instance, which serves as the backend of the faceted data browsers. The extracting functions in the scripts send GET query requests to the SPARQL end-point of Fuseki. The transforming functions convert the query results into semi-structured JSON format data. The loading functions load the JSON data into our Elasticsearch instance by sending PUT or POST requests. We give the whole ETL process another name — ingest.

FacetView2[6] offers a decent faceted-browsing frontend for the JSON data in ElasticSearch. For AMPO use cases, extensive modifications to the code of FacetView2 are made to customize the browsers.

Readers can access a toy demo of the faceted browsers through the following URLs[2]:

- equipment browser: https://dofamp.tw.rpi.edu/equipment.html
- material browser: https://dofamp.tw.rpi.edu/material.html

Users can select different facets on the left side of the webpage. Users can also search and rerank the data using the toolbar on top.

4 FURTHER WORK

The design process of AMPO is use case driven. Although the use case data is often sparse and non-structured, the AMPO framework can easily be modified or extended to accommodate additional use cases due to its flexibility.

Our innovative work, initially sponsored by the Rensselaer Institute for Data Exploration and Applications, has drawn tremendous attention from the Defense Advanced Research Projects Agency (DARPA). We were given additional funding supports to build another management system on a much larger scale for the materials processing data provided by DARPA (grant number DARPA: A12746.1102). AMPO has also been reused as part of a bigger ontology named DARPA Materials Ontology (DMOnto). Due to the confidential nature of the data owned by DARPA, no further details can be revealed here.

REFERENCES

[1] Timothy Lebo, Satya Sahoo, and Deborah McGuinness. PROV-O: The PROV Ontology, 2013.
[2] Michel Dumontier, Christopher JO Baker, Joachim Baran, Alison Callahan, Leonid Chepelev, José Cruz-Toledo, Nicholas R Del Rio, Geraint Duck, Laura I Furlong, Nichealla Keath, et al. The semanticscience integrated ontology (sio) for biomedical research and knowledge discovery. *Journal of biomedical semantics*, 5(1):14, 2014.
[3] http://www.qudt.org/, 2017.
[4] https://jena.apache.org/, 2017.
[5] https://www.elastic.co/products/elasticsearch, 2017.
[6] https://github.com/cottagelabs/facetview2, 2014.

[2]This toy demo might not be very interesting since only the data of one toy use case (fused deposition modeling) is used. In real situations, the faceted browsers could easily handle huge amount of instances and many more faceted categories.