DIAMOND Extension for Additive Manufacturing

Jeren Browning, Chris Ritter

December 2019

Abstract

Oak Ridge National Laboratory is overseeing the Transformational Challenge Reactor project, an effort to additively manufacture nuclear reactors. For this project, Idaho National Laboratory has been tasked with providing a model for the data involved in the additive manufacturing process as well as the creation of a graphical user interface tool for interacting with this data. The former task is described, detailing the creation of new classes, properties, and relationships within DIAMOND to serve as a model for the TCR data and to act as a point of potential integration with other data sources. The new classes for this data fit neatly into the existing DIAMOND structure, and new or existing relationships allow for proper data representation within the model. DIAMOND can now be used as the schema for a graph or relational database, providing the beginnings of a data store or data warehouse for the TCR project and its data. A front end application that is currently under development can then easily access this data for viewing and manipulation.

Introduction

Data Integration Aggregated Model and Ontology for Nuclear Deployment (DIAMOND) is an open source model and ontology for data related to the various aspects of nuclear reactors deployment, such as operations or design. An ontology is a hierarchical data model. It's a collection of concepts within a domain (nuclear) and their properties and relationships.

DIAMOND has several value propositions, primarily to act as a schema for a data warehouse and to provide a common model for data integration. Integrating data from these various sources and organizations within a plant will allow for several gains because it automates the manual search for data, enables sharing and comparison of data from various tools from single or multiple plants, enables digital transformation of data, increases frequencies of needed data that are sparse in nature (such as failure signatures), reduces the need for training on various tools for plant staff, enables a holistic staff perception of plant activities, reveals cost-saving opportunities, and improves the visual perception of all parts necessary for operations and maintenance.

DIAMOND currently contains hundreds of classes and properties about categories of data and processes within a nuclear power plant such as work orders, operator rounds, process instrumentation and control. Physically, DIAMOND is an Extensible Markup Language (XML) document that utilizes standards including the Resource Description Framework (RDF) and Web Ontology Language (OWL) to properly express the classes, properties, and relationships within DIAMOND. Among other potential uses, this XML document can be used as the basis for the schema of a graph or relational database which in turn can act as a data warehouse.

DIAMOND is therefore an excellent fit for the data involved in the ongoing work on the Transformational Challenge Reactor (TCR) being led by Oak Ridge National Laboratory (ORNL). TCR is concerned with the additive manufacturing potential and process for creating a small modular nuclear reactor. This data centers around the builds performed and the various data associated with these builds. For each build performed, data is recorded around variable attributes of the build, the printing equipment being used, and the material components involved in the build. DIAMOND contains existing structure for elements of nuclear power plant (NPP) design and is easily extendable, allowing for the TCR data to be easily represented within DIAMOND and integrated with other applications and processes which may benefit from providing or receiving the TCR data.

TCR Structure in DIAMOND

The TCR data contains information primarily related to the builds that are planned and then performed. A build is a process and type of action, and two classes have been created in DIAMOND to represent both planned and performed builds. A Planned Build is a child of the existing Planned Action class. In addition to inheriting essential properties around duration of the build and relationships to assets (the materials used in the build as well as the 3d printer performing the work), this class adds properties that hold the number of layers in the build and relationships to the Alarm, Log File, and Project Overview classes.

A build that is either in-process or completed is represented through the existing Actual Action class. This class contains all of the necessary properties including start time, end time, percent complete, status, and a relationship to the Action class (and therefore the instance of Planned Build described previously, which holds key information related to the instance of Actual Action with which we are concerned). A visual representation of these classes in DIAMOND corresponding to example TCR data (instances of the DIAMOND classes) can be seen in Figure 1.

A class was needed in DIAMOND for holding information around the material used in a build. This set of information contains details around the physical makeup of the materials being used in the build as well as supplier information. This information is represented through the Material Batch class, a child of the Component class. The Component class is defined as “…an Asset that is used by other Assets (specifically equipment) to perform their function”, making it the ideal choice as parent class for Material Batch. A number of new data properties were required for Material Batch including lot number, supplier, and procurement date. Relationships were also added connecting Material Batch to the Material and Measurement Entry classes, allowing for references to the types of materials used as well as a history of data points represented through the measurement entry class that allows for managing update attributes (such as variables that change over time or from build to build).

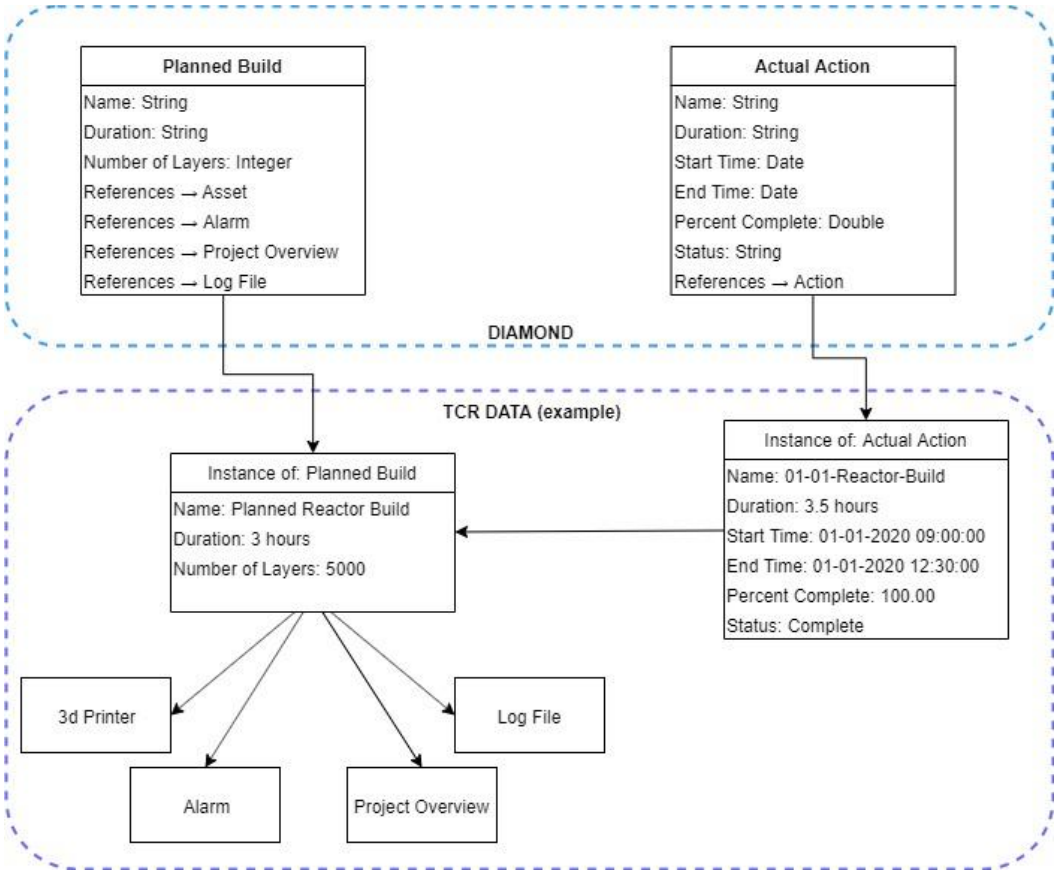

Figure 1

The TCR program also is concerned with data around the 3d printers themselves, and a 3d Printer class under the existing Printer class was created. Much of the core information around the printer itself is inherited. This includes details such as the make, model, serial number, software version, and a date indicating when maintenance was last performed. An additional property indicating whether the filament has been updated and a relationship to measurement entry to handle value attributes (as described previously for the Material Batch class) were also added.

A representation of each of the additive manufacturing classes that have been added to DIAMOND and some of the essential relationships surrounding them can be seen in Figure 2. The classes described previously have been highlighted for easy reference.

A View of Example TCR Data in DIAMOND

Using mock TCR data, an example is provided of how the additive manufacturing data will be represented in DIAMOND. Example builds, material, and machine state and machine updates (primarily through the 3d printer class) along with their properties and associated classes are provided. In this example, the material name being used is “Inconel 738”. A 3d printer that contains machine state and machine updates labeled “Arcam” is used for the builds. Multiple associated builds (actual action) are

given as examples with varying statuses. Additionally, one planned build is provided for understanding of the relationships between it and the other classes.

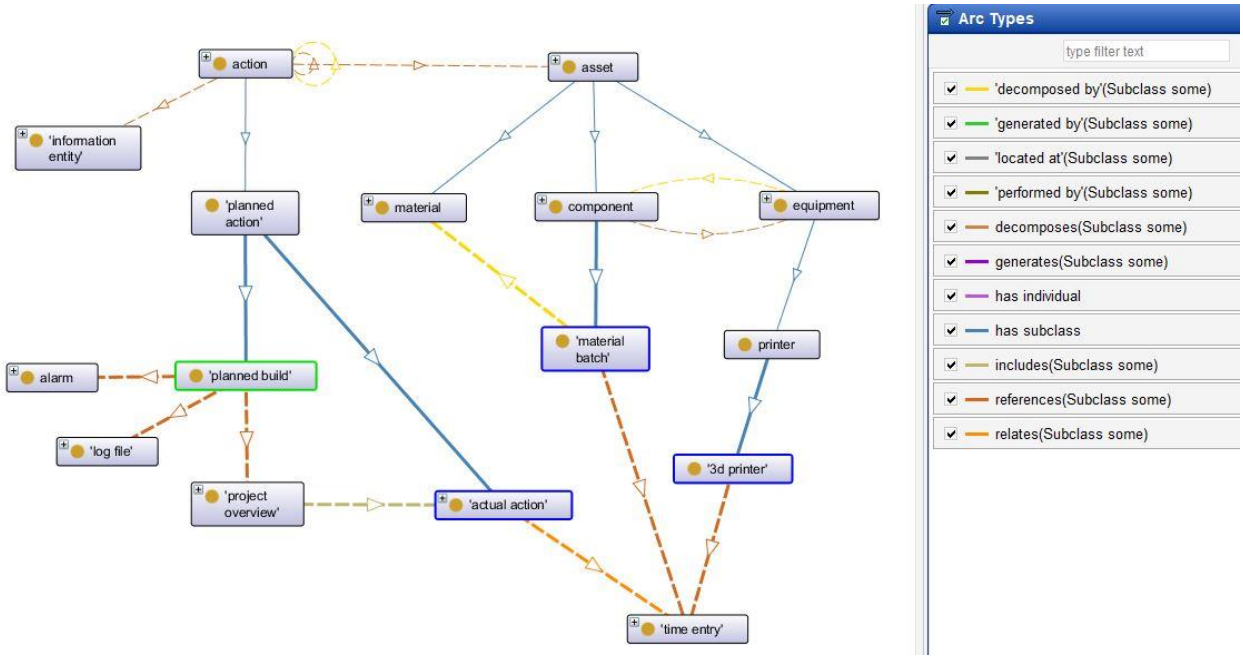

Figure 2

The example TCR data stored in the described DIAMOND model can be seen in Figure 3. The key classes described above (Planned Build, Actual Action, Material Batch, and 3d Printer) are shown along with some of their key properties and the relationships between them. The Material and Measurement Entry classes associated with the Material Batch and 3d Printer classes are also included. Please note that relationships to multiple Material and Measurement Entry classes are shown in the diagram. In this manner, multiple Measurement Entry classes may be grouped together as a form of list to hold current and historical values of some variable over time. Relationships to multiple instances of the Material class can be used to denote the details around the various types of materials, elements, etc. that are included in a Material Batch. Additionally, only classes and properties that are essential to representing the example data of TCR are displayed here. Additional properties, classes, and relationships associated with the TCR data exist within DIAMOND (Figure 2).

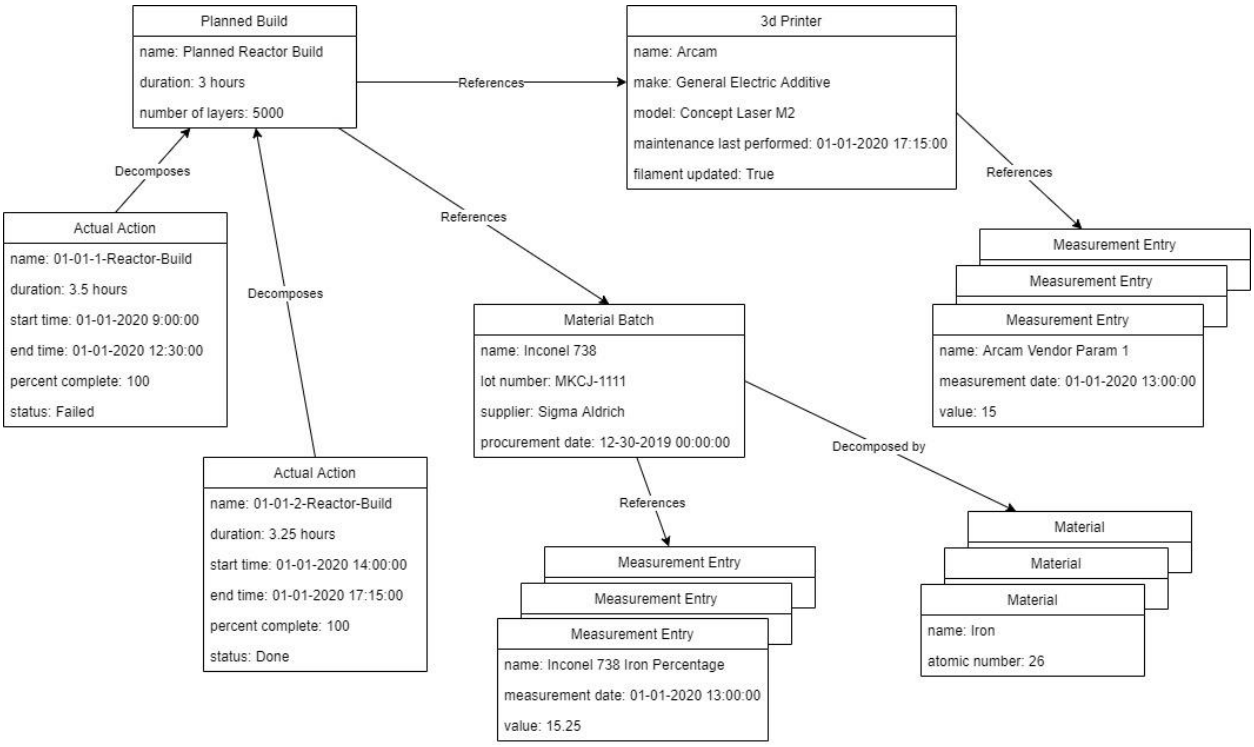

Figure 3

Future Work

With the TCR data structure accounted for within DIAMOND, work can now move forward on instantiating the DIAMOND model within a graph database, such as Microsoft's Cosmos. The actual TCR data can then be ingested into this database, with the data being represented as instances of the classes that have been described within this document. Application Programming Interfaces (API) will be provided for accessing the data from the front end system, allowing for visual interaction with the TCR data.