SAND2019-10869C

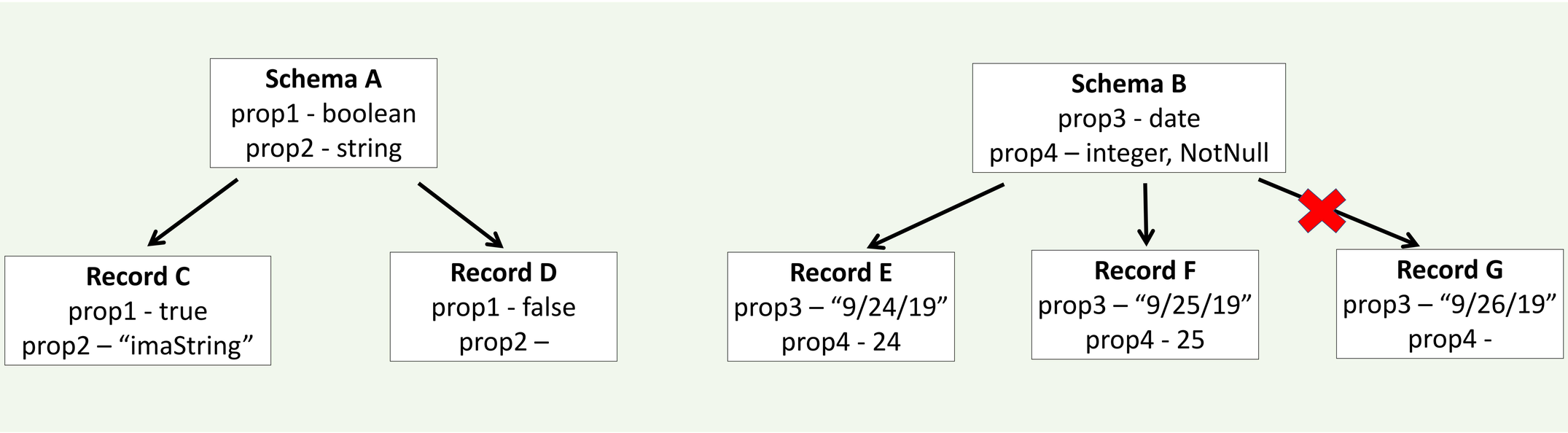
Sandia National Laboratories

Citadel: Dynamic Records

Malachi Tolman

Citadel is a custom code framework designed to quickly roll out multiple data storage applications at Sandia National Laboratories. Citadel helps projects by providing features common to all data system applications. These include: enabling CRUD operations, tracking changes to the data (via a user accessible provenance database), parsing common data types into Parquet files, opening RESTful API's for third party tools to interact with the data (see posters on WASP), and establishing custom access controls to data on a granular level.

Currently the two implementations of Citadel are the SEDS and DataSEA projects. SEDS focuses on replacing a large legacy data system. It has hard-coded data structures and rules for accessing data. DataSEA focuses on quickly rolling out smaller data systems that allow the users to determine their own data structures and custom rules for storing, accessing, and manipulating data.



After quickly discovering the countless permutations of data structures that current and future customers will have when storing data, we sought a flexible solution that would allow a customer to determine their own data structure before storing such in the DataSEA system. We accomplished such via record schemas and dynamic records. Record schemas are json-like objects that determine the structure of records that can be associated with it. Once a schema is declared, users then can create dynamic records based on that schema. In this regard, they have a consistent set of metadata associated with each set of test data that logically goes together.

Schemas are made by listing each property that subsequent records can contain. Each property includes, at least, a property name and property type. The currently supported property types are: string, boolean, date, enumeration, file, file with metadata, float, integer, list, and schema. Optionally, users can add validators to properties while declaring schemas to enforce certain data values in subsequent dynamic records. The most common of these is the "NotNull" validator, enforcing any record made from that schema to have a value in that metadata field. But others that can be currently used are max, min, future, past, and pattern (regular expression for string fields).

Users can create record schemas and dynamic records via three means currently. One being declaratively setting up record schemas and/or dynamic records via third party tool such as Matlab or Python and inserting such directly into DataSEA through our restful API's. The second by using our web interface built on top of Angular. The third being automatically generating schemas and records from TDMS files (National Instrument's/LabVIEW's proprietary file structure) using DataSEA's TDMS parser. This allows current and future customers to store nearly any data they have in a manner that is searchable (see poster on Citadel: Search).

The glaring weakness to this approach is that it is not possible to enforce almost any standards on data structure and best practices. Some steps are being taken to alleviate such, but this problem is still being worked on. Another important note is that record schemas and dynamic records are exclusively for organizing metadata. Dynamic records can have test data attached to such, but the actual parsing and organizing of the test data is done via data frames (see poster on Citadel: Data Frames).

U.S. DEPARTMENT OF ENERGY
National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
SAND No. _____

Sandia National Laboratories