Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

Scientific data management with navigational metadata[☆]

J. Stillerman[*], M. Greenwald, J. Wright

Massachusetts Institute of Technology, Cambridge, MA, 02139, USA

ARTICLE INFO

Keywords:
Data science
Metadata
Annotation
Navigation
Graph data

ABSTRACT

Modern science generates large complicated heterogeneous collections of data. In order to effectively exploit these data, researchers must find relevant data, and enough of its associated metadata to understand it and put it in context. This problem exists across a wide range of research domains and is ripe for a general solution.

Existing ventures address these issues using ad hoc purpose-built tools. These tools explicitly represent the data relationships by embedding them in their data storage mechanisms and in their applications. While producing useful tools, these approaches tend to be difficult to extend and data relationships are not necessarily traversable symmetrically.

We are building a general system for navigational metadata. The relationships between data and between annotations and data are stored as first-class objects in the system. They can be viewed as instances drawn from a small set of graph types. General-purpose programs can be written which allow users explore these graphs and gain insights into their data. This process of data navigation, successive inclusion and filtering of objects provides powerful paradigm for data exploration.

1. Introduction

The size and complexity of the data collected during all kinds experimental science has been growing rapidly over time. This is due at least in part to improved measurement equipment, faster computers, larger storage, and better computer networks. In many spheres, experiments are also getting larger and more collaborative, and these collaborations are often spread out geographically. These factors combine to make descriptive metadata for these data sets imperative for their exploitation. Expressive metadata provides context documenting recorded measurements and computed results. If sufficient metadata is present, then the data can be understood and used by current and future collaborators, without relying on direct communication with the original experimenter, and relying on their memories. The smaller scale experiments of the past had a less urgent need to store and manage metadata, in that the data sets were smaller and more homogeneous, and the size of the scientific teams involved tended to be smaller. A modern experiment such as a tokamak or a particle accelerator may involve hundreds or even thousands of scientists. In order for them to work effectively they need to be able to access and understand the results recorded and stored by their colleagues. Data acquisition and data management tools have been evolving to meet these challenges.

This evolution has been not only in the capabilities and capacity of these tools but also in their level of abstraction. Hand recorded measurements were replaced by pen based chart recorders, oscilloscopes, and Polaroid scope cameras. These in turn led to purpose built hardware and software to record measurements with computers, general purpose hardware with purpose built programs, general data collection programs like LabView [1] and MDS [2], and finally general data collection and management systems such as MDSplus [3] and workflow engines like Kepler [4]. Over time these tools increased in their generality and allowed users to represent their data at higher levels of abstraction.

Most current experiments have ad hoc systems to store and navigate a subset of the relationships between their data. For example, they connect experimental proposals to experiment operations, comments on those operations, machine status, and data provenance. These systems have tended to encode data relationships explicitly, often in the user interface program code itself. Furthermore, the navigation is often asymmetric (e.g. a logbook entry refers to some particular stored data, but this fact is not easily discernable when the data is accessed). Many of the ideas for this project stem from previous work by the authors, Metadata Ontology Project [5], which represented data provenance information as graphs, and data object references as uniform resource identifiers, URIs [6]. It is a logical next step, generalizing the storage and retrieval of relationships between stored data.

[☆] This paper lays out the underlying core ideas, progress to date, and plans for near term work.
[*] Corresponding author.
E-mail address: jas@psfc.mit.edu (J. Stillerman).

https://doi.org/10.1016/j.fusengdes.2018.01.063
Received 15 June 2017; Received in revised form 9 January 2018; Accepted 26 January 2018
0920-3796/ © 2018 Published by Elsevier B.V.

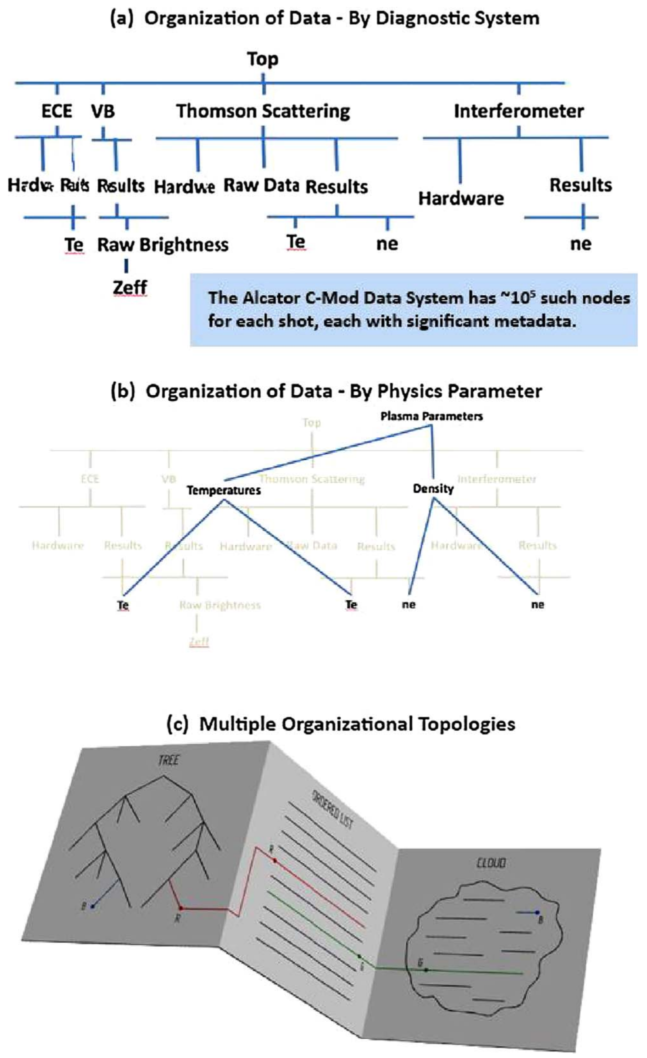

Fig. 1. Data relationships as graphs. (a) and (b) are alternate hierarchical representations of the same collection of nodes. (c) shows a node as part of a hierarchy, an ordered list, and an unordered set.

2. Problem statement

This project will provide a set of tools to define, store, and retrieve a general representation of the relationships between stored data. Distinguishing these from the underlying data stores allows users to create multiple organizations of the same underlying set of objects. These organizations can then be explored using a successive browse (navigate) and filter/search user interface. Users filter by specifying criteria to limit the result set. For example, find things that are ‘annotations’, ‘made by me’, ‘yesterday’. Once a reasonably small list of answers is returned, one can browse their summaries, reorder them and apply further filters or expansions to the list. After locating objects of interest, the user will be able to discover what relationships the objects are members of, and the adjacent objects in those relationships. This adjacency is a key feature enabling data discovery.

A key element of the system is its ability to query the relationships of research objects, and then look at the adjacent objects with respect to those relationships. Fig. 1 provides a cartoon of this multiple facet traversal.

Because we will store relationships and adjacency internally, will be able to make them traversable symmetrically, addressing one of the limitations of previous ad hoc systems. Once target objects are located, the system will, where possible, invoke the native tools to examine them. At all stages, users will be able to add annotations which will become first class relatable elements in the system. Adding new relationships will be done by modifying the schema which is also stored as data in the system. Programs will automatically be aware of the new stored relationships, without requiring new coding in applications. A typical modern science experiment stores a large set of kinds of information, many of which must be integrated by data access tools, or by the user by hand, in order to understand the experimental results. For example, at a tokamak doing magnetic fusion research they collect and store:

- Hierarchical data stores with raw and processed data
- Relational databases with “high level” results
- Electronic logbooks & annotations
- Data provenance graphs
- Data catalogs
- Experiment planning documents
- Records about personnel
- Experimental proposals
- Simulation inputs & outputs
- Source code management systems
- Facility information, with details of
- experiment, measurement systems
- Document management systems
- Publications & presentations

as well as many other items. Some of these are navigable using purpose built applications, others are hermetic external systems. Each of these kinds of information, has a primary organizational paradigm, which orders them and facilitates access to their records. For example, the logbook might contain records organized around experimental run days, instances of the experiment and topic. The raw and processed data, stored in MDSplus for example, are arranged in a hierarchy defined by the site or by individual users. The new system will support multiple organizations of the same underlying data. A user could view the collection of stored records by diagnostic, by measurement type, or by the location/sightline of the diagnostic, see Fig. 1. These multiple organizations will facilitate users finding data relevant to their research.

3. Solution

Our solution to these challenges is based on the idea that relationships can be thought of as graphs, that is collections of nodes and edges. Both of which are subclasses of a general object type which all of the user stored records in the system share. The ‘object’ meta type provides fixed properties that all instances are guaranteed to have: author, date-time-inserted, and type. Type is used both as a gross filter and to inform the user interface system how to display, modify and interact with the object. Each node type will have a list of required and optional properties. They can be either self-contained, with their ‘data’ stored as values of the properties, or refer to external data stores. In this latter case, one or more of their properties are URIs. Examples of these include: records in MDSplus trees, HDF5 files [7] or records in them, other files in a filesystem, documents or drawings in a document management system, etc… Similarly, the relationships, or edges in these graphs, will have types and properties. The required and optional lists of properties for each type of object and relationship form the schema for the system.

A small set of graph topologies can accommodate most of the relationships we need to represent. Users of the system will define their schemas as instances of these metaschemas, and ontologies of the properties of their nodes and edges. The small number of graph types allows general applications to be produced that can be applied not only within local groups or domains, but between disparate communities of users. A hierarchical data explorer could be used on multiple data hierarchies at a site, and shared with users in other science domains.

Download English Version:

https://daneshyari.com/en/article/6743280

Download Persian Version:

https://daneshyari.com/article/6743280

Daneshyari.com