Documenting Scientific Workflows: The Metadata, Provenance & Ontology Project

By
David Schissel

 Presented to
NITRD
MAGIC Meeting
Arlington, VA

March 4, 2015

http://www.tecplot.com
Long History of International Collaboration in Magnetic Fusion Research

• Data Management for experiments
  – Client/server worldwide access
  – Metadata for discovery

• Electronic logbook
  – ~500,000 entries
  – Real-time and searchable

• Our science has evolved to rely extensively on modeling
  – Diverse community extending well beyond the code developers
  – Data management not as comprehensive in the modeling community

• Desire to create a “scientific notebook” for computational science
  – Data has enduring meaning; foster collaboration (Greenwald, IAEA, 2011)
Acknowledging the Support of ASCR & FES as well as the Contributions of the GA, LBNL, and MIT Team Members

- **DOE/SC support critical**
  - Both ASCR and FES
  - Productive partnership going back to the first SciDAC in 2001
  - Fusion Collaboratory, SWIM, Web Portal, Network QoS, ESL, AToM, etc.

- **Thanks LBNL: Arie Shoshoni and Alex Romosan**

- **Thanks MIT/PSFC: Martin Greenwald, Josh Stillerman, John Wright**

- **Thanks GA: Gheni Abla, Bobby Chanthavong/Liz Coviello, Xia Lee**

- **Based on MPO team’s 2014 presentations at the APS/DPP Meeting (Greenwald, et al.) and the NGNS PI Meeting (Schissel et al.)**
  - Metadata, Provenance, Ontology (MPO) Project: 9/1/2012 - 8/31/2015
Objectives: Document Scientific Data Flow

- Preserve meaning of data by documenting all of the steps taken to produce the data = provenance
  - Capture both data and process
  - Support more systematic management of analysis & simulation data

- Provide and preserve answers to two key questions:
  - Where did a particular piece of data come from?
    - What were the inputs, assumptions and parameters used in its calculation?
    - And where did the inputs come from?
  - Where was this data used?
    - Other calculations
    - Publication or presentation
    - Contributions to databases

- FES as a test bed but applicable to all science domains
Example Use Cases

• How did I arrive at the data plotted in figure 6 of my 2014 Phys. Plasmas article?

• A calibration error was found in Thomson Scattering data taken during 2011
  – the data has now been recalculated, but where was the old data used?
  – What publications used that data? Were they critical for the published conclusions?
  – Did we contribute any of that data to an international database?

• A recently graduated PhD student left behind output from thousands of gyrokinetic simulations
  – Which of these were used in her thesis?
  – Which might be useful in the future? What were the inputs and parameters used in the interesting runs?
Non-Functional Requirements

- **Support all scientific workflow – experimental & computational**
  - Typically involves processing of raw data, with small or large codes often providing inputs to larger simulations, whose output requires processing as well

- **Allow users to record as much/little information as they need**

- **Function in a heterogeneous environment and interoperate with whatever workflow tools people are already using**
  - Researchers use many different languages (Shell scripts, python, IDL, Matlab, etc.) and tools to get their work done
  - Many different computational platforms – laptop to HPC
  - Data is stored in different formats (MDSplus, HDF5, ASCII, etc.)
  - It would be futile to insist that researchers change all of that to get the benefits that we propose

- **Once set up, needs to work as automatically as possible (so best suited for scripted rather than one-time use)**
Basic Components of the MPO System

- **Database/ Database Server**
  - Captures metadata, location of data, and all processing steps

- **API/ API server (Application Program Interface)**
  - Mediates all communication with database
  - Gives users a language interface to instrument their workflow scripts

- **Web Server**
  - Provides interactive user interface to discover & explore workflows
  - Allows users to enter new comments about any MPO object

- **Event Server**
  - Enables automatic updates of workflow information
Basic Components of the MPO System
**Workflows Depicted as Mathematical Graphs**

- **Directed Acyclic Graphs (DAGs)**
  - Directed: flow is defined and one-way
  - Acyclic: Loops are not allowed
  - Graph: Set of objects, connected by links

- **Shows each step in the processing chain**

- **The parent-child relationship is stored and can be followed in either direction**
  - Properties can be inherited
  - Allows simple consistency check e.g. are all parents older than their children?
For Every Record, We Want to Provide Information on “Who, What, When, How, and Why”

• **What:** Each object has a user-supplied name and description
  – Plus contextual information
  – Plus optional metadata
  – Plus data pointers

• **Who and when:** Every MPO object is automatically tagged with a time and the user’s name

• **How:** Via the workflow connections

• **Why:** Supplied through comments/annotation
Data Model: What we Store

- **Data objects**: Structured data, mostly stored outside the database schema
  - MPO keeps pointers – in the form of URI (Uniform Resource Identifier) - that uniquely identifies the data and its access method
  - Additional metadata maintained to aid in searching & browsing
- **Activities (actions)**: Anything that creates, moves or transmutes data from one form to another
  - Includes data importing, staging, file copying, pre-processing, operation of large and small codes, data writing, post-processing, data exporting
- **Connections**: The causal links between inputs, actions & results
- **Comments**: User annotation as unstructured text
- **Collections**: Simple lists of any type of MPO objects, defined by users for any purpose
Persistent Data Store: Data Objects Must be Maintained

- Underlying the model is an assumption that data objects will be maintained
  - If the underlying data are allowed to change in untracked ways, the descriptions and provenance are corrupted
  - Data can be moved to a new location or converted to a new format – as long as this is written down in the MPO database

- MPO does not dictate the implementation of the persistent store
  - Data objects can be a reference to a user’s file system
  - Data objects can be a description of how to retrieve the item from a database or record store
  - Data objects can be a description of how to retrieve files or directories from a file store

- Methods are available to manage persistent store’s data in a manner consistent with maintaining the integrity of the MPO system
Collections

• Users can define “collections”
  – Each tagged with description or purpose

• Arbitrary sets of objects of any kind

• Example Uses:
  – Multiple runs in a parameter scan
  – Workflows that contribute to a particular publication or presentation

• Objects can be members of any number of collections

• Collections of collections can be defined
Shared Objects and Connected Workflows

• **Typically a user will employ multiple workflows in a particular application**
  – For example: Code A provides the spatial mapping for raw data; processed data is input into Code B; Code B’s output is compared to Code C’s output

• **We chose not to define sub-workflows and sub-sub-workflows as too complex and confusing**

• **Instead, workflows are linked via shared data objects**
  – i.e. these data objects have more than one connection
  – Each connection is tagged with the workflow id
  – This provides the head to tail coupling between workflows
  – Shared data objects are highlighted in the user interface, allowing users to navigate from workflow to workflow
  – It allows easy re-use of data objects – a common occurrence
Managing the Namespace

• Each object in the MPO has a globally unique numerical identifier

• Workflows can be found by searching or browsing, but how would you convey that information to someone else?
  “Take a look at my TRANSP run 1234”
  – We define a composite ID that is easy to remember

• Each data object is provided with a pointer in the form of a URI
  – Uniform Resource Identifier,
  – Superset of the URL (Uniform Resource Locator)
  – The URI is the pointer to the data object

• Searching is enhanced by defining a “controlled vocabulary”
  – User-defined, hierarchical ontology
Controlled Vocabulary: Ontology

• In computer science, an ontology is a formal framework for representing information
• The MPO employs a user-defined ontology to describe types of metadata
• This enhances searching since the vocabulary for a particular application is defined
  – So in a particular application I can see that I want to search for "confinement_mode" = “H-mode” and not “conf_mode” = “H-Mode” or “Hmode”
• Users can browse or search the ontology
• Users can add terms to the ontology
• The MPO ontology is arranged in a hierarchy to enhance browsing
MPO Project is an **Applied** Computer Science Project

- MPO software utilizes open source solutions wherever possible
- MPO is a “web service”
- “PostgresSQL” database used for current implementation
- Both API server and Web UI server use “Flask”, a lightweight web application framework
  - API based on REST abstraction = Representational State Transfer
  - Database operations through HTTP verbs (e.g. post) and URLs
- Twitter “Bootstrap” to create standardized Web front-end
  - Hides Javascript complexity
- DAGs rendered by “Graphviz” software
- Authentication via x.509 certificates (OSG, MIT & MPO certs)
- MDSplus event services
Substantial Progress has been Made

• **Basic components all built**
  – Database schema defined and implemented
  – API available in shell, python, IDL
  – Web-base user interface built - supports searching and browsing, dynamic display of workflows and metadata

• **Production and development environments are available**

• **In the process of beta testing**
  – SWIM, the SWIM Portal, and the AToM Project
  – GYRO
  – EFIT including DIII-D’s between shot analysis while operating
  – TORIC

• **Evangelizing the philosophy throughout the community**
  – 3 IAEA/TM papers (2013, 2015), APS/DPP (2014), PI meetings
MPO Web Site Operating with Ontology-based Search, Automatic Real-Time Graphics, Live Data Loading
Project’s Final Year Goal is to Expand System’s Depth and Expand the Reach of our Tools into other Sciences

- **Taking on friendly beta users**
  - Presentation at APS/DPP Nov. 2014 (attracted beta users)
  - Support other languages, add requested features, documentation

- **Beginning to work with a difference science domain**
  - CASCADE Project: DOE’s Regional/Global Climate Modeling Program

- **Hardening for Production**
  - Formalize schema updates, separate development/production/user sandbox, develop/guarantee our persistent store

- **Continue to evolve MPO UI and data schema**
  - For example: UI evolving to handle large quantity of workflows, adding collections
Questions We are Asking Ourselves Today

• How to expand the reach of our MPO framework?
  – Across many science domains (ease of adoption, robust)
  – Federated system within a science (fast at large scales)

• Compatibility with W3C Standards (e.g. PROV)
  – How to import/export to MPO?
  – Can draw in this ecosystem (e.g. Annotation WG)?

• Efficient UI operation at large-scale
  – How to do better/faster Graphical Navigation?

• Provide rich data centric tools
  – Are there different UIs to the MPO data?
Summary

• Substantial progress towards a production system
  – API, data store/Ontology, & UI all evolved

• Production workflows have been MPO instrumented
  – DIII-D experimental analysis & SWIM simulations

• Our results validate our approach
  – Simple API to instrument basically any existing workflows
  – General data store and UI to store and navigate

• Include Climate Modeling science domain moving forward
  – Yield feedback to allow iteration on the MPO framework

• Presentation at the 10th IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation in Fusion Research