GA–A28275

AUTOMATED METADATA

Final Project Report

by
D. SCHISSEL

Prepared for the
U.S. Department of Energy
under Contract No. DE-SC0008697

GENERAL ATOMICS PROJECT 30392
DATE PUBLISHED: APRIL 2016

GENERAL ATOMICS

DISCLAIMER

GA–A28275

AUTOMATED METADATA

Final Project Report

by
D. SCHISSEL

Prepared for the
U.S. Department of Energy
under Contract No. DE-SC0008697

GENERAL ATOMICS PROJECT 30392
DATE PUBLISHED: APRIL 2016

GENERAL ATOMICS

Final Project Report

Project Title:	Automated Metadata
Project Period:	09/01/2012 – 04/30/2016
Project Budget:	Total: $649,657.00 (Estimated-Funded Cost)
Submission Date:	04/29/2016
Recipient:	General Atomics
Address:	3550 General Atomics Court San Diego, CA 92121
Award Number:	DE-SC0008697
Project Team:	General Atomics (GA)
GA Contacts:	Principal Investigator: David Schissel Phone: 858-455-3387 Email: schissel@fusion.gat.com Business Officer: Ms. Ziba Ahkami Phone: 858-455-2507 E-mail: Ziba.Ahkami@ga.com
DOE Contacts:	Technical Manager: Rich Carlson Phone: 301-903-9486 Email: richard.carlson@science.doe.gov Project Officer: John Mandrekas Phone: 301-903-0552 E-mail: Thomas.rueckert@ee.doe.gov Award Administrator: Kristie Bromberek Email: Kristie.Bromberek@science.doe.gov Phone: 630-252-2081

1. EXECUTIVE SUMMARY FOR THE PUBLIC

The MPO (Metadata, Provenance, Ontology) Project successfully addressed the goal of improving the usefulness and traceability of scientific data by building a system that could capture and display all steps in the process of creating, analyzing and disseminating that data. Throughout history, scientists have generated handwritten logbooks to keep track of data, their hypotheses, assumptions, experimental setup, and computational processes as well as reflections on observations and issues encountered. Over the last several decades, with the growth of personal computers, handheld devices, and the World Wide Web, the handwritten logbook has begun to be replaced by electronic logbooks. This transition has brought increased capability such as supporting multi-media, hypertext, and fast searching. However, content creation and metadata (a set of data that describes and gives information about other data) capturing has for the most part remained a manual activity just as it was with handwritten logbooks. This has led to a fragmentation of data, processing, and annotation that has only accelerated as scientific workflows continue to increase in complexity.

From a scientific perspective, it is very important to be able to understand the lineage of any piece of data: who, what, when, how, and why. This is typically referred to as data provenance. The fragmentation discussed previously often means that data provenance is lost. As scientific workflows move to powerful computers and become more complex, the ability to track all of the steps involved in creating a piece of data become even more difficult. It was the goal of the MPO (Metadata, Provenance, Ontology) Project to create a system (the MPO System) that allows for automatic provenance and metadata capturing in such a way to allow easy searching and browsing. This goal needed to be accomplished in a general way so that it may be used across a broad range of scientific domains, yet allow the addition of vocabulary (Ontology) that is domain specific as is required for intelligent searching and browsing in the scientific context.

Through the creation and deployment of the MPO system, the goals of the project were achieved. An enhanced metadata, provenance, and ontology storage system was created. This was combined with innovative methodologies for navigating and exploring these data using a web browser for both experimental and simulation-based scientific research. In addition, a system to allow scientists to instrument their existing workflows for automatic metadata and provenance is part of the MPO system. In that way, a scientist can continue to use their existing methodology yet easily document their work. Workflows and data provenance can be displayed either graphically or in an electronic notebook format and support advanced search features including via ontology. The MPO system was successfully used in both Climate and Magnetic Fusion Energy Research.

The software for the MPO system is located at https://github.com/MPO-Group/MPO and is open source distributed under the Revised BSD License. A demonstration site of the MPO system is open to the public and is available at https://mpo.psfc.mit.edu/. A Docker container release of the command line client is available for public download using the command *docker pull jcwright/mpo-cli* at https://hub.docker.com/r/jcwright/mpo-cli.

2. PROJECT SUMMARY

2.1. INTRODUCTION

This report summarizes the work of the Automated Metadata, Provenance Cataloging, and Navigable Interfaces: Ensuring the Usefulness of Extreme-Scale Data Project (MPO Project) funded by the United States Department of Energy (DOE), Offices of Advanced Scientific Computing Research and Fusion Energy Sciences. Initially funded for three years starting in 2012, it was extended for 6 months with additional funding. The project was a collaboration between scientists at General Atomics, Lawrence Berkley National Laboratory (LBNL), and Massachusetts Institute of Technology (MIT). The group leveraged existing computer science technology where possible and extended or created new capabilities where required.

The MPO project was able to successfully create a suite of software tools that can be used by a scientific community to automatically document their scientific workflows. These tools were integrated into workflows for fusion energy and climate research illustrating the general applicability of the project's toolkit. Feedback was very positive on the project's toolkit and the value of such automatic workflow documentation to the scientific endeavor.

2.2. BACKGROUND

Data, from large-scale experiments and extreme-scale computing, is expensive to produce and may be used for high-consequence applications. However, it is not the mere existence of data that is important, but our ability to make use of it. Experience has shown that as we make the associated metadata better organized and more complete, the more useful the underlying data becomes. Further, the infrastructure or a set of tools that automatically create, discover, display or explore the semantic relationships of complex data from experiments and computer simulations does not currently exist, though concepts or paradigms from the semantic web may prove useful. Generous provisioning of metadata, including data provenance and data relations, is critical to enhance data sharing, to allow data to retain its usefulness over extended periods of time, and to allow traceability of results. There is an unmet need to better document workflows that create, transform, or disseminate data and to capture (and later present) data provenance [1]. We note that the class of needs addressed by this project have been identified and discussed in a number of recent DOE sponsored workshops. These include:

(1) Data Crosscutting Requirements Review, April 2013 [2]
(2) Accelerating Scientific Knowledge Discovery (ASKD), July 2013 [3]
(3) Workshop on the Future of Scientific Workflows, April 2015 [4]
(4) Integrated Simulations for Magnetic Fusion Energy Sciences, June 2015 [5]
(5) Data Management, Visualization and Analysis of Experimental and Observational Data workshop, Sept. 2015 [6]
(6) ASCR/FES Exascale Requirements Review, December 2015 [7]
(7) Data+Compute Convergence Requirements Review, March 2016 [8]

The goal of this research was to create a data model, infrastructure, and a set of tools that support data tracking, cataloging, and integration across a broad scientific domain. The software infrastructure developed (MPO system) allows the documentation of workflows and data provenance in the widest sense, enabling scientists to answer the questions “who, what, when, how and why” for each data element; provide information about the connections and dependencies between the data elements; and allow human or automatic annotation for any data element. The MPO system is designed to capture information from the creation, recording or importing of physical data, through various levels of analysis, data preparation, data staging, HPCC code execution, storage, post processing, data exporting, and publication. Because researchers use a wide variety of tools for these activities, the approach of the MPO was to provide a mechanism for instrumenting any user scripts, programs or procedures to capture provenance at a level of granularity appropriate for the application. We did not seek to impose a standard workflow engine on all users.

Although using Fusion Energy Sciences as a test bed, the conceptual framework and data model are quite general and do not contain specific references to the fusion domain. Although the equations solved by simulations are different for different fields of science, the basic flow of information, the need to document workflow and provenance, allowing traceability of results is common to all. Similar common needs exist for experimental data. Furthermore, all fields of science struggle to integrate information from simulation and experiment and to extract knowledge from the confrontation between the two and our goal was to aid this task as well.

2.3. THE MPO SYSTEM CONCEPTS

The MPO system captures information by annotating workflow scripts utilizing the application-programming interface (API) for recording provenance. The API allows instrumentation of any suitably extensible workflow engine or other lightweight workflow capability (e.g. scripting) by providing "catalog-aware instrumentation" libraries. The MPO system also contains a Web-based user interface (UI) that allows searching or browsing as well as visualization of the annotated workflow and data relations in a variety of formats including a "scientific notebook" and scientific information visualization.

In order to create a model for capturing and organizing metadata in the widest sense, and supporting systematic analysis, the MPO project defined several entity concepts. The following are the entities defined by MPO data model:

Data Object – A piece of information that is related to a scientific activity or research. It can be a large dataset, one single value, or a research paper. A Data Object can be either produced or consumed by a computer code. The Data Object can be stored as a file or put into a scientific data store (e.g., MDSplus [9,10]) for later retrieval via a client-server API. A key concept of the MPO system is that it does not store data objects but instead records unique pointers to the data objects in the form of a Uniform Resource Identifier (URI). Data Objects are tracked in two ways. First is the general information about the Data Objects themselves, such as the URI, creation date, and author's username. Second is the specific use of the Data Object in a workflow. This is done to allow the system to keep track of Data Objects and their uses in a more organized manner.

Activity – Anything that creates, moves, or transmutes data from one form to another. An activity often both consumes and produces data. It can have multiple input Data Objects, and can produce Data Objects as output. As a special case, activities can connect to each other in order to permit data in memory to be passed from activity to activity. Examples of Activities include data importing, pre-processing, input preparation, executing codes, data storage, post-processing, and data exporting.

Connection – The causal link between multiple Activities and/or Data Objects.

Workflow – A series of connected Data Objects and Activities, which can be organized as a Directed Acyclic Graph (DAG) that consists of one-way flows with no loops allowed. A Workflow shows the individual steps in the processing chain and the parent-child relationship among its elements (Data Objects and

Activities), which can be followed in either direction. Data Objects, Activities, and Connections are the building blocks of a Workflow.

Collection – A group of related entities. It can contain any number of Data Objects and Workflows. A Collection also can include other Collections. Each element in a Collection can be shared among multiple Collections. Relevant use cases are: 1) A series of simulation runs in a parameter scan; 2) Multiple computational data analysis workflows and associated data used in a published paper.

Metadata – A text-based, arbitrary name-value pair that is associated with a Data Object, Activity, Workflow, or Collection. Using Metadata is a flexible way to keep track of values without having to update the MPO system to explicitly handle that type of information. Examples of Metadata include last-updated dates for Data Objects and informative notes for an algorithm in an Activity.

Comment – Text (including hypertext) information that is associated to a Data Object, Activity, Workflow, or Collection. A Comment can also be associated with another Comment, which means that Comments can be added recursively. Comments are unstructured and may come with a few fixed attributes, such as user ID and timestamp of creation. Comments are either automatically generated by programs or manually entered by scientists, and are essential in capturing the user-generated metadata of all basic MPO entities.

Figure 1 shows an example of a workflow for documenting the data preparation steps to execute the GYRO code [11], a large-scale gyrokinetic or neoclassical simulation of plasmas.

In addition to the above basic entities, the MPO System also defines two other concepts needed for documenting workflows: Provenance and Ontology.

Provenance is the lineage of Data Objects. It traces the path of a Data Object, starting from creation, through every transformation until the end. Every time a workflow is executed, an instance of that execution is generated. That instance represents the provenance, which includes the Data Objects and parameters used as input for each step, the Data Objects and parameters generated as output, and the sequential relationships between the steps. Extensive information, such as where a piece of data came from, how it was created, why and by which Activity, are captured as the contents of Provenance.

The Ontology is a structure that captures the common terms used to describe object properties in a specific scientific domain. This is also referred to as controlled-vocabulary or classification structures. It is common to represent such structures as tree structures, where the leaves of the trees are referred to as "narrower terms", and the higher-level elements as "broader terms". "Broader terms" are used for general categories, and narrower terms are used for the

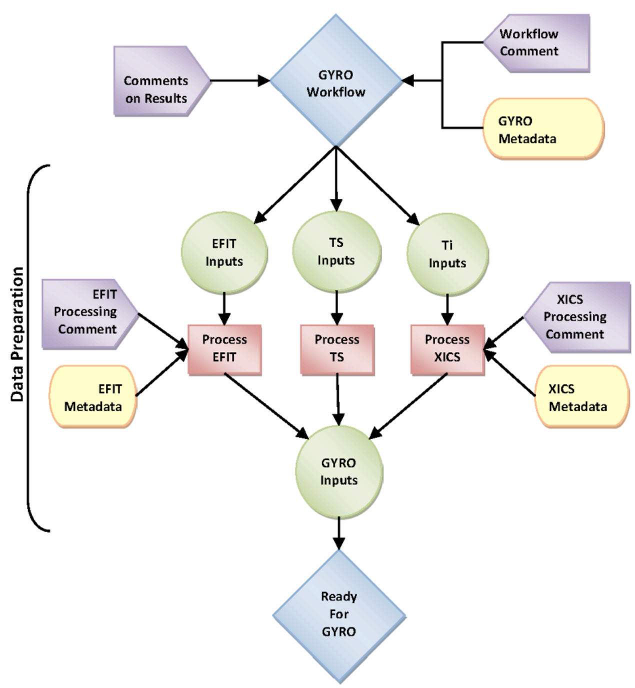

Figure 1. Example data preparation workflow
to execute the GYRO code.

specific branches of a general category. Classification terms can be used for any purpose, depending on the domain, and need to be developed in cooperation with domain scientists. In this project, Ontology is a description of main MPO entities (Data Object, Activity, Workflow, Comment, and Collection) via their hierarchical relationships. Each element of the Ontology category (also referred to as facets) can form a hierarchy of broader terms that spawn narrower terms. The main purpose for Ontologies is to allow for more accurate searching of the all the basic MPO entities and Provenance information.

The MPO entities described above are the basis for automatically generating visual representations and describe relations among multiple Data Objects, Activities, and Workflows. They are also essential in creating a software framework for documenting Provenance and Ontology.

2.4. THE MPO SYSTEM ARCHITECTURE

The architecture of the MPO system has been driven by the need to support both experimental and computational research activities especially in extended, collaborative environments. Such research activities involve processing of raw data, with small or large computational codes often providing inputs to larger simulations, whose output requires processing as well. The computational codes were often developed with a variety of languages (FORTRAN, IDL, C/C++, Python, shell scripts, Matlab), and they may require a variety of computational environments (e.g., operating systems, interconnects, software libraries). The computer hardware can be a laptop, desktop, computer cluster, or supercomputer. The input and output data and storage formats are heterogeneous, including, but not limited to, MDSplus, HDF5, NetCDF, JSON, and CSV. Multiple scientists may repeat some workflows and the same Data Object may be used in multiple similar or different workflows. Some workflows represent single activities, and others can be a part of larger simulation efforts, such as parameter scanning carried out by many workflows.

To address the above reality, the MPO system was developed using multi-tier web services including a RESTful API [12] that can be easily utilized in a variety of programming languages. The architecture of the MPO system and its main components are shown in Figure 2.

The building blocks of the MPO system are: 1) Database; 2) API Server and Event Server; 3) Interactive UI server; 4) Clients. The heart of the system consists of two main servers: the API Server and the interactive UI Server. Only the API server directly communicates with the Database. There are two types of clients: Native Application Client and Browser-based web client. While “Native Application” clients directly communicate with the API server, the web browser connects to the UI server.

The final production version of the MPO system architecture is slightly different

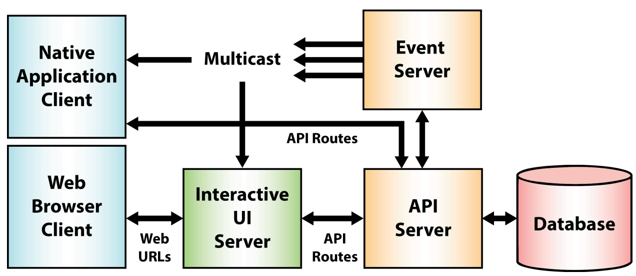

Figure 2. The main components of MPO system.

from the initial prototype. In the prototype version, a single web server provided both the API service and the interactive UI service and they used the same network port. The production version maintains those two services with their responsibilities clearly separated. The idea behind this separation is to keep the API server independent so that other (desktop, mobile, web) client applications can be developed independently from the interactive UI server.

Database. The MPO database is responsible for storing all the MPO information. Its schema is based on the entities discussed in Section 2.2 that are represented with database tables. While some tables hold directly accessible information such as those based on Data Object, Activity and Collection, some are used to describe other entities/tables and are not intended for direct data access. For example, the Connection table is used to describe relations between other entities, such as Workflows, Data Objects, and Activities, in a DAG structure. The database also includes additional tables such as the users table, which stores user information and the authorization table, which holds the access control information.

Currently, PostgreSQL, an open source relational database system, is chosen for the database server. However, the MPO database schema design is general enough that the database server can be easily replaced with any other relational database software. The generality of the schema also helps the separation of the database implementation from representing the concepts in the API as well as user interfaces.

API Server and Event Server. The MPO API server exposes its services via a RESTful API and currently only supports GET and POST methods. The basic entities in the MPO data model are represented with corresponding RESTful resources. All routes are accessed in the form of HTTP requests http://<server>/<type>/<version>/<route>/UID. Here <server> refers to the hostname and the port number of the API server. The API server uses <type> to refer the requests to the appropriate MPO environment such as production or testing. The <version> type refers to the version of the MPO API that is being used. This allows for multiple versions of the MPO API that is crucial due to the evolving nature of the software. The UID is a universal unique identifier (UUID v4) created during a POST method and may be used with a GET method to refer to a specific entity. POST responses are in JSON format and include specific information such as the UID of the new object, the URI of the object, the user who created the object, and the time recorded when the entry for the object was made. Queries using GET may include key-value arguments to identify or filter metadata fields. Finally, <route> is the corresponding resource. All routes have a sub-route/<route>/<UID>, which provides information on a specific entity identified by the UID.

The MPO system has core routes, along with the role of GET and POST methods, for /workflow, /dataobject, /activity, /collection, /collection/<UID>/element, and /comment. In addition to the core routes listed above, several supplementary routes are also provided. One such route is /workflow/UID/graph/. This route provides server side rendering of visualization for workflow graphs in SVG or PNG format. Other additional routes mainly provide convenience for reducing the number of server connections thus facilitating faster data retrieval.

The API server has been constructed using Flask [13], which is a lightweight micro web application framework written in Python. The simplicity and extendibility of this framework are the main reasons for choosing Flask for the MPO development.

The MPO event server is an additional service that runs side-by-side with the API server. It is implemented by utilizing the MDSplus system's event features and distributes asynchronous events for real-time updates by clients. "Data updated" events are generated when new data is added to a certain workflow so that clients can asynchronously call the API server to retrieve the latest information. For example, web clients use this event server for triggering Asynchronous JavaScript and XML (AJAX) calls and rendering pieces of the webpage to display the ongoing progression of the workflow in near-real-time.

Interactive UI Server. The MPO system's interactive UI server provides visualization and interactive browsing of the MPO data via a web browser interface. There are four main sections available: Workflow, Data Object, Collection and Search.

On the Workflow page, each workflow is displayed with its Workflow ID, Description, and Creation Time. Additionally, corresponding quality ratings and the number of comments for each workflow are also displayed. Users can choose to view or enter comments by clicking a comment link associated with a workflow. This page also provides an ontology-based filtering of the workflow list.

Clicking on any of the workflows in the Workflow page opens the related Workflow details page. This page provides a rich interactive environment for viewing and exploring all the metadata about the specific workflow, thus creating a dynamic "notebook" interface. It utilizes an interactive and real-time graphical interface to present the data associated with a specific workflow. The interface is comprised of an interactive workflow diagram, an expandable list of nodes that include user comment log, list of metadata and other linked workflows. The diagram offers pan and zoom capabilities, and also displays metadata associated with the node when selected. The workflow node listing provides users with a chronological list of workflow nodes. When a node is clicked, it expands and reveals all associated metadata. Users also have the ability to view and add node-specific comments on this workflow page. Using an event system, these respective workflow elements are updated in real-time as new data is added. For example, when a new workflow node is added, the node will appear in the workflow diagram and is also appended to the node list with its metadata. In the DAG visualization, if a workflow element is shared with other workflows, it is automatically detected and highlighted in blue. Figure 3 shows a workflow details page of an EFIT [14] workflow.

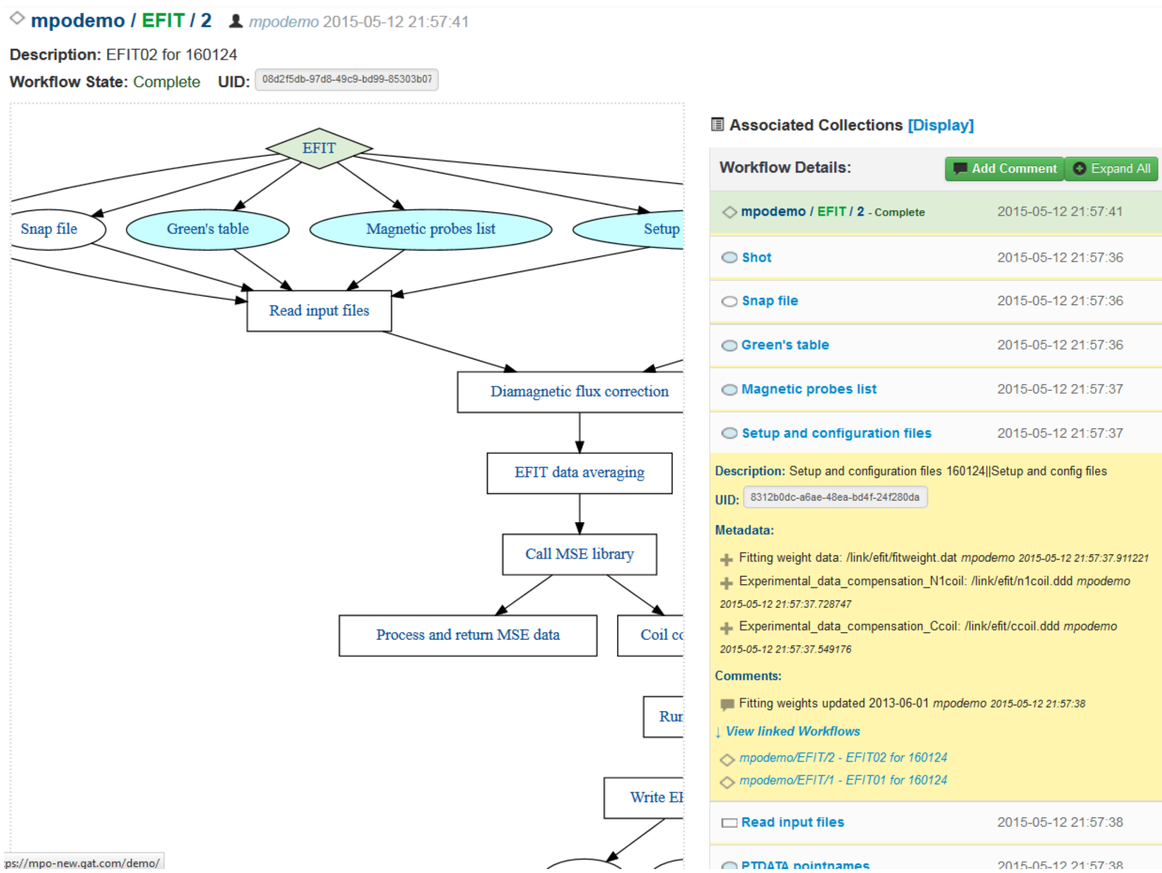

Figure 3. An EFIT workflow displayed on an MPO workflow details page with an interactive DAG and a listing of nodes.

On the Workflow page, the ontology fields are based on the user-defined ontology terms. These are dynamic and depend on the resulting list of workflows. Unlike the metadata fields, not all ontology terms are assigned to all workflows, so only the terms used in the displayed workflows will be available for the ontology filter selection. The UI presents these fields in an intuitive and organized manner by grouping them by their parent terms and presenting each field with a counter of associated workflows from the displayed result. Users can click to toggle the checkboxes next to the desired ontology term(s) to filter and narrow down or widen the result set. Figure 4 includes the initial list of all available workflows with the full ontology filter options and an example of a filtered list of workflows with a subset of filter options only used in the filtered list.

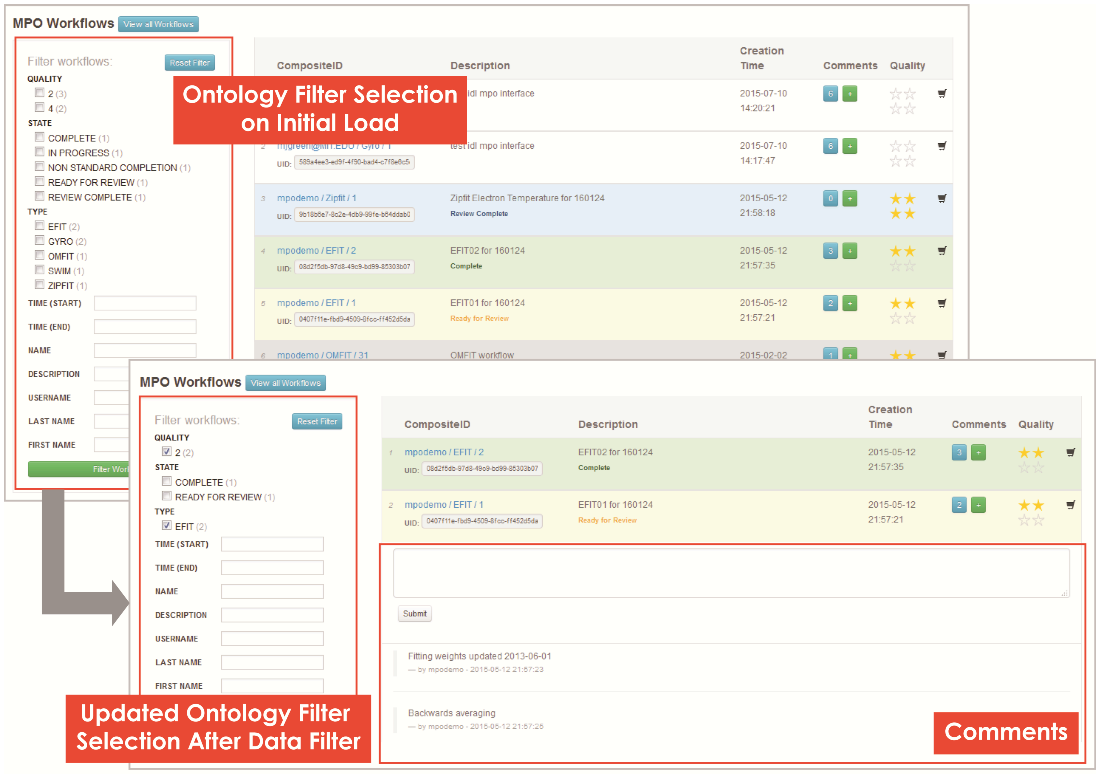

Figure 4. An MPO workflow page partially overlapped with a subset of workflows filtered by conditions on Quality and Type.

The Data Object page and Collections page provide similar lists of corresponding entities. Clicking on any item in the list opens the details page for the specific Data Object or Collection. The Search page provides a single search box to search all recorded MPO entities by a keyword.

The UI also includes a Collection Cart that allows users to temporarily save Workflow, Data Object and/or Collection items to assign to desired collection(s) at a later time (Figure 5). This utilizes HTML5 local storage to temporarily hold the cart information within the user's web browser, until they are saved to the database when added to a collection.

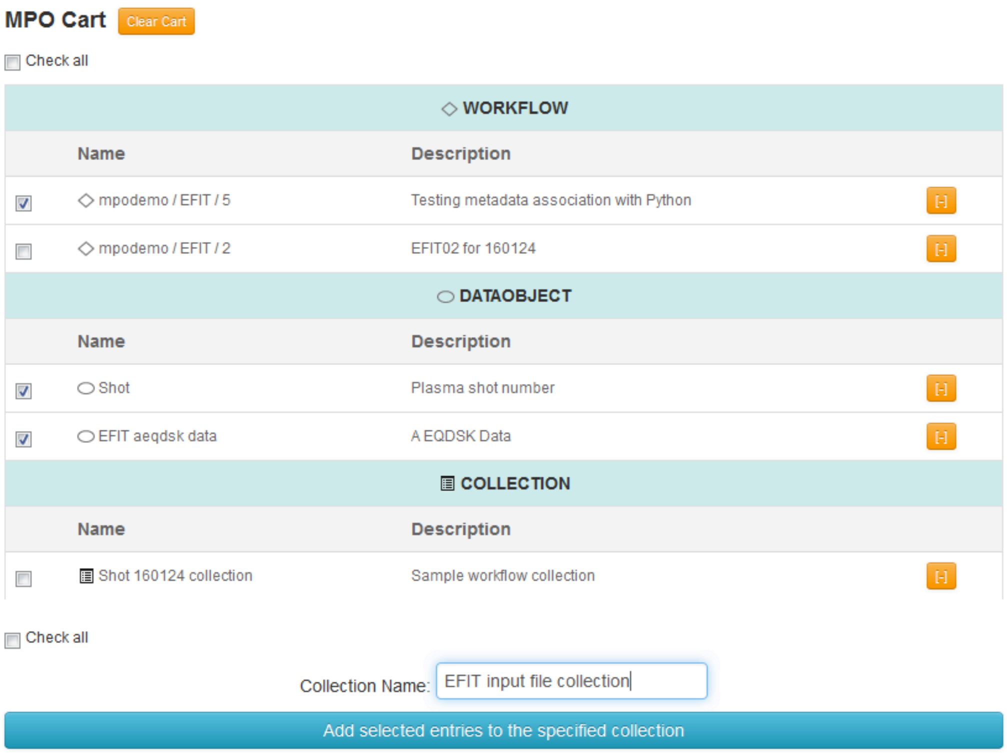

Figure 5. MPO Collection Cart page with Workflow, Data Object and Collection entries. Some entries are selected to be added to the collection, “EFIT input file collection”.

Similar to the API server, the interactive UI Server development also relies on the Flask framework. Workflow visualization, and interactive browsing as well as searching capabilities were developed via open source web technologies, such as HTML, Graphviz, Ajax, JavaScript, and Twitter Bootstrap.

MPO Clients. The MPO API server provides necessary routes for implementing client commands and the RESTful API facilitates integrating them into a variety of applications. There are RESTful-supported tools and libraries in almost all the major programming languages. For example, the interactive UI server is actually a Python client of the API Server and it is implemented using RESTful libraries in Python.

To further simplify the process and hide unnecessary and repetitive details from the end-users, a command line interface package using the Python language (mpo_arg.py) was developed. The package supports the following methods:

init/stop: begins/finishes a new workflow,

add: creates a new dataobject (optionally adds a reference),

step: adds an activity to a workflow,

comment: attaches a comment on an item in a workflow,

collection: creates a new collection or add an element to an existing collection,

meta: adds a key/value metadata to an item,

auth: authenticates using OpenSSL certificates,

archive: storing data in persistent store and/or creating data objects from persistent store.

These commands can be invoked in Python codes or directly from the command line. The mpo_arg.py package and its commands have been integrated with multiple computational workflows when instrumenting them as an MPO client. Those workflows have been developed with IDL, Python, Matlab, and Linux shell scripts.

2.5. AUTHENTICATION AND AUTHORIZATION

The MPO project was careful to implement robust and standard authorization and authentication protocols that are widely used and accepted for web based applications. All data is communicated with secure socket layers (SSL) over HTTPS using either MPO issued SSL certificates or an institution's own. Users may login to the MPO using either certificate authorities of their institutions choice, user certificates issued by the MPO or usernames and passwords either specific to the MPO or from the institutions own LDAP (lightweight directory access protocol) user database. This authentication is mediated through the API server. Since for architectural flexibility, the MPO uses separate servers for the web user interface, the API server and the event server, there is a challenge of backend client authentication. That is, when the user logs in, they authenticate themselves to the API server, but the API server needs to forward that authentication to the UI server or a command line client. This is done by setting up an OAUTH (Open AUTHorization) service on the API server that issues user specific tokens to clients after the user logs in.

Once the user is authenticated and has an access token, it is used for subsequent transactions between the API server and the user interface, command line clients or MPO calls within an application. The token uniquely identifies the user through supported API routes that clients can access. When a transaction is initiated though an API route, for example to GET a list of workflows, the token is provided. On the API side, the token is used to resolve the user and the user's authorization for the requested action is verified based on their role as stored in the database. User's roles and privileges are configurable through a web interface accessible by users with an Administrator role. In this way, authorization is performed and may be configured.

2.6. DEPLOYMENT

After testing the early prototype system with several workflows, many improvements have progressively been made to the MPO system and its first production version has been released. The production version of MPO has been integrated with multiple new workflows.

EFIT Workflow. The EFIT (equilibrium fitting) code [14] is a data analysis application used in magnetic fusion research to calculate the magnetohydrodynamic (MHD) equilibrium in a toroidal magnetically confined plasma. The automated between-pulse computation of the plasma shape by EFIT at the DIII-D National Fusion Facility was instrumented with the MPO API via the MDSplus data acquisition and data management system. The MPO client commands have been adapted into workflow scripts and automatic metadata generation has been instrumented during the experiments. Figure 3 shows a portion of an instance of the EFIT workflow. This segment shows the main trunk of the workflow, which takes a few input files, connects to the source data (from diagnostic measurements) to extract the necessary parameters for the simulation. The MPO development team has worked with the workflow developers to ensure that only the high level information is captured, which ensures the information presented is a clean workflow diagram, and all interesting aspect of the workflow could be reflected in the figure.

SWIM Workflow. The Simulation of RF Wave Interactions with Magnetohydrodynamics (SWIM) was a proto-Fusion Simulation Project (FSP) project that coupled advanced fusion simulation codes into a computational framework that operates on computational clusters including supercomputers at ORNL and NERSC [15]. The SWIM system has been in use for a number of years to do production scientific data analysis and instrumentation via the MPO API was straightforward. After each simulation is completed, the MPO client methods are triggered automatically. The metadata information has been created by filtering status data stored in the SWIM web portal service.

CASCADE Workflow. Workflows from a climate data analysis project known as CASCADE [16] have been instrumented with the MPO system. The particular workflow is a simplified version for detecting Atmospheric Rivers, a process that is primarily data parallel in nature. The particular run of the workflow starts with hundreds of data files, which creates hundreds of identical branches of the DAG, where each of the branches is essentially the same. To reduce clutter, we have chosen to only show three such branches in Figure 6.

This is one of the first workflow instrumented without hands-on help from the MPO developer team. The graduate student who performed the instrumentation spent only a few hours on the task and was able to instrument about half a dozen functions. The test run was conducted on a supercomputer named Edison at National Energy Research Scientific Computing (NERSC) center located at Berkeley California. From the captured metadata, we created a simple high-level flow diagram that shows the essence of the workflow.

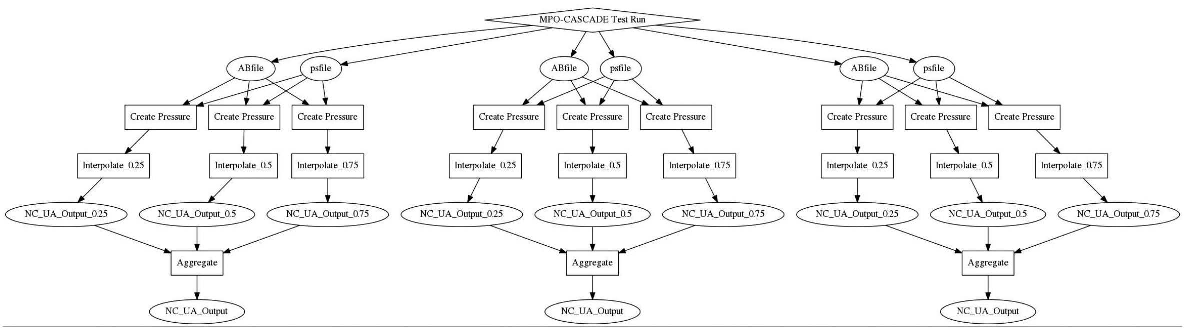

Figure 6. A sample workflow is displayed from the CASCADE climate project.

OMFIT Workflow. OMFIT [17] is a newly developed python-based data analysis framework for magnetic fusion energy. The MPO Python client package was used to instrument the workflow engine in OMFIT to create a hierarchical representation of scripts that has been executed. The kinetic EFIT regression test cases executed through OMFIT collected MPO metadata along with graphical representation of the workflow. The initial results were well received by the OMFIT developers.

The software for the MPO system is located at https://github.com/MPO-Group/MPO and is open source distributed under the Revised BSD License. A demonstration site of the MPO system is open to the public and is available at https://mpo.psfc.mit.edu/. A Docker container release of the command line client is available for public download using the command *docker pull jcwright/mpo-cli* at https://hub.docker.com/r/jcwright/mpo-cli.

2.7. ASSESSMENT

The project would be considered a success if assessed by its initial objective to create a system to allow automatic capture of workflow steps in such a way to allow easy browsing and searching. But even more important, the MPO system has received positive feedback from the scientific community on its unique value for data provenance tracking and workflow documentation. The integration into the AToM project, the usage during DIII-D operations, and the initial work on the

CASCADE climate project demonstrates the systems diverse appeal. From the proposal, the project's goal/objectives were to:

1) Create an enhanced metadata, provenance, and ontology storage schema combined with innovative methodologies for navigating and exploring these data that is generally applicable to both experimental and simulation-based scientific research across a broad scientific domain.
2) Allow application users to continue with the tools they are accustomed to working with (i.e. scripting languages) but provide capabilities for annotating the scripts, so that the workflow structure can be viewed externally, as well as commands that can automatically write provenance to a database that can later be queried.
3) Capture the workflow steps and present it in a graphical form (DAG) by automatically using the annotations methodology created by the project.

The MPO system, with is various components (Fig. 2), satisfy the three goals listed above. The database schema is rich enough to capture all relevant metadata and allow the creation of rich ontology to describe the specific domain science (Goal 1). The web UI is highly interactive, allowing for rapid searching and browsing, in both a textual and graphical context (Goal 1&3). The MPO API is versatile enough to allow scientists to instrument their existing workflows with minimal modification allowing automated workflow documentation (Goal 2).

As with any newly created software system, there are always improvements that can be made. Although the MPO system has been used in a production setting, the development team has identified several areas where they feel enhanced capabilities would be most valuable. Those areas of interest include allowing for metadata evolution over time paying particular attention to schema and ontology evolution. As the MPO system is used more and more, data and workflow discovery will become more challenging and therefore novel ways to rapidly discover data relevant to their work must be deployed. These will undoubtedly need to include enhanced UI capability.

2.8. PROJECT PRESENTATIONS AND PUBLICATIONS

The MPO team presented and published its results throughout the project's lifetime.

Schissel, D.P., et al., "Automated Metadata, Provenance Cataloging and Navigable Interfaces: Ensuring the Usefulness of Extreme-Scale Data," presented at the NGNS PI Meeting, Berkeley, CA, 2013.

Schissel, D.P., et al., "Automated Metadata, Provenance Cataloging and Navigable Interfaces: Ensuring the Usefulness of Extreme-Scale Data," presented to DOE/ASCR at the International Conference for High Performance Computing, Networking, Storage, and Analysis, Denver, CO, 2013.

Schissel, D.P., et al., "Automated Metadata, Provenance Cataloging and Navigable Interfaces: Ensuring the Usefulness of Extreme-Scale Data," presented at the 9th IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research, Hefei, China, 2013; published in Fusion Engineering and Design 89 (2014) 745-749.

Wright, J., et al., "The MPO API: A Tool for Recording Scientific Workflows," presented at the 9th IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research, Hefei, China, 2013; published in Fusion Engineering and Design 89 (2014) 754-757.

Schissel, D.P., et al., "Automated Metadata, Provenance Cataloging and Navigable Interfaces: Ensuring the Usefulness of Extreme-Scale Data," presented at the NGNS PI Meeting, Rockville, MD, 2014.

Greenwald, M.J., et al., "Documenting Scientific Workflow: The Metadata, Provenance, and Ontology Project," presented at the 56th Annual Meeting of the APS Division of Plasma Physics, New Orleans, LA, 2014.

Schissel, D.P., et al., "Documenting Scientific Workflows: The Metadata, Provenance, and Ontology Project," presented to DOE/ASCR & DOE/FES, Germantown, MD, 2015.

Wright, J., et al., "The Need for a General Methodology for Automatic Workflow Documentation," presented at the Next Generation Networking for Science/Computer Science Scientific Workflows Workshop, Rockville, MD, 2015.

Schissel, D.P., et al., "Documenting Scientific Workflows: The Metadata, Provenance, and Ontology Project," presented at the NITRD MAGIC Meeting, Arlington, VA, 2015.

Abla, G., et al., "The MPO System for Automatic Workflow Documentation," presented at the 10th IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research, Ahmedabad, India, 2015; accepted for publication in Fusion Engineering and Design.

Wright, J., et al., "How to Find Your Data – Six Months Later," presented at the International Conference on Numerical Simulations of Plasmas, Golden, CO, 2015.

Schissel, D.P., et al., "The MPO Project," presented at the NGNS PI Meeting, Rockville, MD, 2015.

Wu, K., et al., "MPO: a System to Document and Analyze Distributed Heterogeneous Workflows," submitted to the 6th International Provenance and Annotation Workshop, McLean, VA, 2016.

ACKNOWLEDGMENT

This work was supported by the US Department of Energy, Office of Advanced Scientific Computing Research and the Office of Fusion Energy Sciences under Award Numbers DE-SC0008697, DEAC02-05CH11231, and DE-SC0008736.

REFERENCES

[1] M. Greenwald, T. Fredian, D. Schissel, J. Stillerman, "A metadata catalog for organization and systemization of fusion simulation data", Fusion Engineering and Design 87, 2205-2208, 2012

[2] http://science.energy.gov/~/media/ascr/pdf/program-documents/docs/ASCR_DataCrosscutting2_8_28_13.pdf

[3] http://www.orau.gov/ASKD2013/

[4] http://extremescaleresearch.labworks.org/events/workshop-future-scientific-workflows#block-system-main

[5] https://www.burningplasma.org/activities/?article=Integrated%20Simulations

[6] Report in preparation (2016).

[7] Report in preparation (2016).

[8] Report in preparation (2016).

[9] T. Fredian, J. Stillerman, "MDSplus Remote Collaboration Support-Internet and World Wide Web," Fusion Engin. Design 43, 327 (1999).

[10] http://www.mdsplus.org/

[11] J. Candy, et al., "An Eulerian Gyrokinetic-Maxwell Sover," J. Comput. Phys. **186**, 545 (2009).

[12] R.T. Fielding, et al., "Principled Design of the Modern Web Architecture," ACM Trans. Internet Technol., **2**(2), 115-150 (2002).

[13] http://flask.pocoo.org/.

[14] L.L. Lao, et al., "Reconstruction of Current Profile Parameters and Plasma Shape in Tokamaks," Nucl. Fusion, Vol **25**, 1611 (1985).

[15] D.B. Batchelor, et al., “Advances in Simulation of Wave Interactions with Extended MHD Phenomena,” Journal of Physics: Conference Series, **180,** 012054 (2009).

[16] The Calibrated and Systematic Characterization, Attribution, and Detection of Extremes (CASCADE) Project, http://cascade.lbl.gov/.

[17] O. Meneghini, et al., “Integrated Modeling Applications for Tokamak Experiments with OMFIT,” Nucl. Fusion **55**, 083008 (2015).

GENERAL ATOMICS