DIBBs Brown Dog[1], PaaS[2] for SaaS[3] for PaaS

Kenton McHenry, Jong Lee, Michael Dietze, Praveen Kumar, Barbara Minsker, Richard Marciano, Luigi Marini, Rob Kooper, Dave Mattson

It is largely acknowledged that the scientific method is in jeopardy in the information age. Specifically, the result of the scientific method, the documented procedure, must be reproducible in that when it is carried out the documented result will be obtained each and every time. Today, however, these procedures nearly always involve software and digital data. Both software and digital data are known to have lifespans, typically fairly short lifespans on the span of years or decades. As a consequence science utilizing software and digital data is now to a large degree, if not completely, un-reproducible after a relatively short period of time.

To address this both software and digital data must be preserved. This is not simply storing the bytes on disk or tape. One must be able to properly interpret those bytes in order to execute it if it is software or get at the desired contents if it is data. In order to get at and interpret those bytes one typically needs one or more of the following:

- the existence of metadata describing the data (possibly as simple as useful file/directory names),
- the software used to create and view the data,
- the file format specifications describing how contents are represented within the file's bytes,
- and for software the execution environment (platform, operating system, libraries, other software, etc…).

Preserving this is a daunting task spanning issues such as:

- obsolete operating systems and platforms,
- storage requirements (e.g. storing a working environment in a virtual machine),
- software that is no longer available,
- software licensing,
- the existence of many file formats (even for the same kind of data),
- lack of standards for data formats or enforceability of standards,
- large complex file format specifications,
- unavailable format specifications (either lost or proprietary),
- the ease and reward of creating data versus the burden of curation (e.g. organizing and providing metadata for files),
- different metadata and data schemas,
- assuring the long term availability of preserved software and data,
- assuring the archive preserving the software and data exists over a reasonably long period of time,

[1] http://browndog.ncsa.illinois.edu
[2] Preservation as a Service
[3] Software as a Service

- assuring the archival tools needed to index, find, access, view, retrieve, and utilize the software and data within the archive exists over a reasonably long period of time (being software itself).

We are attempting to attack this need for digital preservation towards scientific reproducibility through a multi-pronged non-linear manner addressing several of the issues above at once. Towards this we also consider a number of other attributes that make up the landscape of science today:

- a growing notion towards the need of academic reward, and perhaps education, surrounding the costly products of software development and data creation
- the necessity for science to build off of the work of others and have software and data reused (possibly in ways not remotely considered by the creator and crossing into other disciplines)
- need for computation during the analysis of data collections
- means of efficiently and reliably transferring large amounts of data

Accessing Data Contents with a Lack of Standards and Many File Formats

Many formats for data, even for the same types of data, exist. This will continue for a variety of reasons spanning the ease by which one can define their own format for data so as to minimize their work during data creation, imperfect formats for various data structure needs, the length of time required to standardize and adopt a community wide format, an inability to completely enforce a standard, as well as commercial reasons. To build software that supports every possible format is a daunting, if not impossible, task given the number of formats, the unavailability of specifications for many of those formats, and the size/complexity of some format specifications.

Towards building a service to access as wide as possible variety of data contents within such an environment we utilize two tools that allow us to attack this problem from opposite ends simultaneously and in an extensible manner. The first, a tool called a Software Server[4,5], allows one to wrap arbitrary software, including GUI driven software, in order to expose a programmable REST interface so that the software might be automated and parallelized within a distributed environment. Realizing that some software somewhere was required to produce a file in a given format in the first place, and that many applications often support importing and exporting to a handful of other formats, a Software Server allows one to assemble a system by which to automatically chain together the capabilities of a large number of applications in order to convert data into a format that is accessible. The second tool, Daffodil, an open source implementation of the Data Format Description Language (DFDL)[6,7], allows one to create an XML schema for file formats (both text based and binary). With such a schema Daffodil is capable of parsing a file of the described format into XML, which in turn is more readily accessible. Future versions of daffodil will also support the inverse, going from XML back to a format (possibly a different

[4] http://isda.ncsa.illinois.edu/drupal/software/polyglot
[5] http://isda.ncsa.illinois.edu/drupal/software/Software%20Servers
[6] https://opensource.ncsa.illinois.edu/confluence/display/DFDL/Daffodil%3A+Open+Source+DFDL
[7] https://redmine.ogf.org/dmsf/dfdl-wg?folder_id=5485

format from the original). DFDL schemas can be used to describe the many ad hoc formats scientists and labs produce internally as well as some of the more straight forward formats that have specifications available. Software Servers will attempt to fill in the rest using available software capabilities. Within the Brown Dog effort these two tools are used towards the creation of a service refereed to as the Data Access Proxy (DAP) which provides conversion capabilities to both users and applications via a programmable REST interface.

Discovering and Finding Data with a Lack of Curation while also Considering the Need to Preserve Software and Provide Credit for Software Development

In order to access data one must first find it (and discover if it exists before that). One typically finds data through associated text and metadata describing the contents of the data. For what is referred to as the long tail of scientific data, most data, this does not exist. There are currently a variety of funded efforts looking at providing tools to aid in the curation of data collections, to both make curation less tedious and the "value add" more apparent. We complement these efforts by considering means of minimizing human involvement and providing a tool for the automation of uncurated collection indexing.

To aid in the indexing of uncurated and/or unstructured collections we are developing a service referred to as the Data Tilling Service (DTS). Like the DAP the DTS is also made up of two components. The first is Medici[8,9], a web based content management system which provides mechanisms for the social as well as the auto curation of data. Specifically, as data is uploaded into a repository a suite of tools called extractors are automatically executed should the uploaded file or dataset match the conditions for an extractor (e.g. a type or format of a file). These extractors generate previews of the data, derived data products, and metadata. We adopt Medici and its extractor framework and focus on developing the tools to automatically extract metadata from a file's contents. Specifically, we consider tools that take in data and produce, or extract, new data via some form of analysis. This can involve tools from artificial intelligence, machine learning, computer vision, audio recognition, natural language processing or scientific tools carrying statistical analysis as well as other tasks. The second tool is Versus[10]. Versus is a framework for content based comparisons which essentially provides a numerical level of proximity between two files based on how similar their contents are. Versus breaks these comparisons down into reusable and interchangeable components called adapters, extractors, descriptors, measures, and indexers. We focus largely on the extractors within Versus which produce typically binary signatures which numerically capture some semantic aspect of a files contents. Within the DTS we utilize Versus and content based comparison as another, complementary, mechanism by which to find and relate uncurated/unstructured data. Like the DAP the DTS provides a service in the extraction of metadata and/or content based signatures to users and applications via a programmable REST interface.

[8] http://medici.ncsa.illinois.edu
[9] https://opensource.ncsa.illinois.edu/confluence/display/MMDB/Home
[10] http://isda.ncsa.illinois.edu/drupal/software/Versus

In addition to building a number of broadly serving extractors, for both the Medici and Versus frameworks within the DTS, we also attempt to position the service towards taking in analysis tools from the scientific community. We do this for a variety of reasons including: gaining additional functionality, preserving scientific code, and allowing others to reuse scientific code. Towards this we aim to make the inclusion of new tools as Medici or Versus components as simple and minimal as possible, as inclusive as possible (essentially anything that takes data in an produces some derived product as output), back the service up with computational resources as an immediate "value add" to scientists who include their tools, and provide mechanisms to provide scientists with credit should their tools be used by others.

Creating Tools for Accessing Data while Addressing Archival Tool Sustainability

At the National Center for Supercomputing Applications several pieces of software developed there for scientific needs live on to this day, decades after their creation. These include Mosaic[11], httpd, and telnet. While these tools were built for scientific needs, towards accessing resources and data on high performance computers, they also ended up serving a role for users at large, i.e. the general public. For example, few today do not rely on a browser that has its roots in Mosaic, scientist or otherwise.

We attempt to position the DAP and DTS services constructed here towards a similar form of software sustainability by deliberately adding a general public use case. Specifically, we are developing DAP and DTS services in a manner that would potentially allow it fill a role within the internet much like a Domain Name Service (DNS) for data. Through the programmable interfaces provided by both services we plan to construct a number of software components including a module to an open source operating system which would allow one to specify a DAP and DTS hostname within their network configuration. Once set a number of other modified components would utilize these services. For example a plugin to an open source browser would allow one to search and access data over the web more easily by indexing whole sites based on file contents and converting files automatically so that they are loadable and/or previewable to the user within their native environment. A similar type of functionality would be provided at the file system level by modifying the file manager within the operating system.

Summary

We attempt to address a number of the issues facing scientific reproducibility within the information age, specifically those surrounding long tail scientific data. Towards this we propose two services, the DAP and DTS, which each address a handful of the considerations being faced and provide services by which to allow users to better search, access, and use uncurated data collections while simultaneously serving as a means of preserving, advertising, and crediting the code used to populate these services. Lastly, we attempt to do this in a manner that also serves the general public towards the sustainability of the two services.

---

[11] http://en.wikipedia.org/wiki/Mosaic_%28web_browser%29