Collaborative WorkBench for researchers - *Work smarter, not harder*

Rahul Ramachandran, NASA/MSFC

Kwo-sen Kuo, NASA GSFC and Bayesics LLC

Manil Maskey, University of Alabama in Huntsville, ITSC

Christopher Lynnes, NASA/GSFC

1. Problem

As scientists work to understand the complex interactions in the Earth's atmosphere, oceans, biosphere, surface, and interior, they increasingly find themselves working with a more diverse set of data, complex science algorithms, and models. However, each new dataset, algorithm, or model requires its own knowledge base to obtain reliable and useful results. Acquiring this knowledge base can be both difficult and time consuming. One way to address this is to enable social collaboration, with experts coming together (virtually) to pool their knowledge, tools, and datasets. The concept of "virtual" research collaborations in science is not new. Bos et al [1] proposed the concept of a *collaboratory* as "an organizational entity that spans distance, supports rich and recurring human interaction oriented to a common research area, and fosters contact between researchers who are both known and unknown to each other, and provides access to data sources, artifacts, and tools required to accomplish research tasks."

While there has been a huge growth in social networking and collaboration platforms aimed at the general public, only a few have been successfully customized to address and impact Earth science research modalities despite the obvious utility inherent in such collaborations. One reason is that to go beyond sharing simple documents, scientific collaboration platforms typically require using a new set of analysis tools to leverage the collaboration infrastructure. As a result, adoption of most new collaboration environments and/or analysis tools for scientific research is inhibited by the steep learning curve faced by individual researchers.

As part of the NASA Computational Modeling Algorithms and Cyberinfrastructure (CMAC) program, we are building an Earth science Collaborative Workbench (CWB) to address this barrier. CWBs augment a scientist's current computational research environment and tool set to allow him or her to easily collaborate with others and share diverse data and algorithms. CWBs provide a science algorithm development environment that seamlessly integrates the researcher's desktop with a cloud infrastructure. CWBs focus on enabling not only the sharing of research artifacts such as algorithms, data, and analysis results, but also the collaborative development of

algorithms and interpretation of data and analysis results as well as their knowledge (in the form of annotations). CWBs mark a momentous improvement in the ability of the Earth science community to effectively exchange and share data and information.

2. Approach/Methodology

It is important to define some commonly used terminology related to collaboration to facilitate clarity in later discussions. We define *provisioning* as infrastructure capabilities such as computation, storage, data, and tools provided by some agency or similarly trusted institution. *Sharing* is defined as the process of exchanging data, programs, and knowledge among individuals (often strangers) and groups. *Collaboration* is a specialized case of sharing. In collaboration, sharing with others (usually known colleagues) is done in pursuit of a common scientific goal or objective. Collaboration entails more dynamic and frequent interactions and can occur at different speeds. *Synchronous* collaboration occurs in real time such as editing a shared document on the fly, chatting, video conference, etc., and typically requires a peer-to-peer connection. *Asynchronous* collaboration is episodic in nature based on a push-pull model. Examples of asynchronous collaboration include email exchanges, blogging, repositories, etc.

2.1 Concept of a Collaborative Workbench

The purpose of a workbench is to provide a customizable framework for different applications. Since the workbench will be common to all the customized tools, it promotes building modular functionality that can be used and reused by multiple tools. The objective of our Collaborative Workbench (CWB) is thus to create such an open and extensible framework for the Earth Science community via a set of plug-ins. Our CWB is based on the Eclipse [2] Integrated Development Environment (IDE), which is designed as a small kernel containing a plug-in loader for hundreds of plug-ins. The kernel itself is an implementation of a known specification to provide an environment for the plug-ins to execute. This design enables modularity, where discrete chunks of functionality can be reused to build new applications. The minimal set of plug-ins necessary to create a client application is called the Eclipse Rich Client Platform (RCP) [3]; The Eclipse RCP also supports thousands of community-contributed plug-ins, making it a popular development platform for many diverse applications including the Science Activity Planner developed at JPL for the Mars rovers [4] and the scientific experiment tool Gumtree [5]. By leveraging the Eclipse RCP to provide an open, extensible framework, a CWB supports customizations via plug-ins to build rich user applications specific for Earth Science. More importantly, CWB plug-ins can be used by *existing* science tools built off Eclipse such as IDL or PyDev to provide seamless collaboration functionalities.

2.2 Infrastructure Components

While the CWB plug-ins aid researchers by supporting data analysis and visualization clients, there are infrastructure components required to support our vision of collaboration. These components are:

A. Catalog Service

One of the key components of the infrastructure is a catalog service that serves to store and manage information for both the personal resources for a researcher as well as all the community resources. The personal resource management or the "myScience" catalog provides active management of a researcher's activities, projects and science artifacts. The information model of the myScience catalog is based on the notion of an Experiment. Experiments serve as containers for collections comprising of one or more logical data sets, science codes (workflows) and ancillary (config) files. A workflow maps to a set of different individual programs that are chained together and executed to meet the goal of the experiment. The myScience catalog supports search capabilities via metadata annotations for resources. The "Community" catalog component supports collaboration by connecting researchers and making metadata available for the shared science artifacts stored on the Cloud infrastructure. The Community Catalog serves as a community metadata repository that can be searched via a browser or service API for all the shared science artifacts.

Both the myScience and Community catalogs are built using Drupal content management framework, a robust open-source collaboration framework well supported by the developer community. This allowed us to utilize a number of existing features within Drupal with minimal code development. These features include the modules which provides REST-based service APIs for the CWB to connect the Community Catalog with the myScience Catalog and built-in roles and permissions functionality enabling users to control permissions on the shared science artifacts with other individuals, groups, or the entire community.

B. Cloud Infrastructure

Cloud computing has now become a viable option to provide scalable computing infrastructure for enterprises. As defined by NIST [6], "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." More and more enterprises are moving toward Cloud computing because of its elasticity and scalability, thereby being able to expand or

reduce resources based on demand. Clouds allow multi-tenancy, where multiple users can leverage the same infrastructure and can balance workloads across servers – both inside the data center and across data centers.

For our initial imlementation, we have configured our CWB to utilize Amazon's EC2 [7] and S3 Cloud infrastructure [8] via the AWS Java API [9]. The user's CWB account and corresponding Amazon Identity and Management (IAM) [10] account is managed by the CMS. Execution of workflows is performed in EC2. User-uploaded data and experiment results are stored on S3. Workflows (and programs) can access the data stored on S3 during execution, which in turn, is transferred to EC2. There is no cost associated within Amazon for data transfers between EC2 and S3. Users are assigned a personal space ("bucket") where they can create new experiments. Asynchronous sharing of data and workflows occurs by allowing a set of users to copy these resources to the searchable "community" bucket. Collaborators can also directly import the contents in any experiment folder stored in the community bucket to their personal space. Currently, any Python or IDL science code as well as some specialized libraries such as ADAM Data Mining toolkit [11], can be executed in the EC2. The deployment and the execution of the programs on EC2 is enabled by a set of REST [12] web services. For supporting synchronous collaborations, openfire implementation of XMPP protocols [13] has been used to customize the Eclipse Communications Framework (ECF) plug-ins. This addition enables real time writing and editing of science code within any application using CWB plug-ins.

3. Results

We demonstrate the value of a CWB via a use case, in which two scientists are involved in algorithm development for NASA's Global Precipitation Measurement (GPM) [14] mission, a Decadal Survey Era mission [15]. GPM is a constellation of satellites-of-opportunity [16], consisting of a GPM core satellite and a number of constellation satellites as they become available. The GPM core satellite has as its payload a dual-frequency precipitation radar (DPR) contributed by the Japan Aerospace and Exploration Agency (JAXA) and a GPM microwave imager (GMI) [17] contributed by NASA. The constellation satellites will be contributed by other US agencies or international partner agencies, most of which will carry only microwave radiometers.

A mission such as GPM involves science expertise of several disciplines and thus corresponding communities, including but not limited to electromagnetic scattering, radiative transfer, radar and radiometer engineering/science, atmospheric numerical modeling, and data analysis. Since GPM is an international mission, members of these communities are spread over diverse nations and regions of the globe. In addition, each member researcher may have his or her own favorite tools, algorithms, and data management approach. This diversity makes effective collaboration exceedingly

difficult either among members of the same community or, even worse, across communities. Our CWB is conceived and designed to address the major “pain points” in such situations. In the following use case, we focus on a (tiny) segment of the retrieval algorithm development process for clarity’s sake to demonstrate the capabilities of our CWB after providing some necessary background.

Just as for the remote sensing of any other atmospheric particulate matter, e.g. aerosol and cloud particles, the single-scattering properties (SSPs) of precipitation particles are also the fundamental quantities for remote-sensing measurements of precipitation. These single-scattering properties, or SSPs, are used by an instrument response simulator (IRS) [18], which is basically composed of a selected Radiative Transfer (RT) model from a suite of RT models as well as modules for simulating engineering components, to simulate instrument (radar and radiometer in this case) responses to precipitating atmospheric columns in various weather systems and regimes. These simulations in turn form a basis for precipitation retrieval.

In this CWB demonstration, an electromagnetic scattering expert, Alice, is in charge of providing SSPs of irregularly shaped atmospheric ice particles for retrieval algorithm development. Bruce is the instrument response simulator (IRS) expert who needs to work with Alice to incorporate the SSPs along with other types of precipitating particles into his simulations. Since the types of ice particles are rather complex and there is no simple way to classify them unambiguously, Alice uses alphanumerically coded names in a directory structure to identify and organize the particles she has created, as well as their corresponding SSPs obtained for a number of microwave frequencies covering the range needed for the mission.

With the conventional means of collaboration, Alice would have to “package” all her results and generate the documentation, explaining what the coded names mean and how the package is organized, i.e. what means what and which information is where. More likely than not, Bruce would have trouble using the SSP package without further consulting Alice. Alice would probably explain further in a few more emails but there is no guarantee that Bruce “gets it”. More interactive means of communication, like a phone call, may be called for. But, due to their time-zone disparity, finding a suitable time may take a couple more email exchanges.

All Bruce actually needs to know is just where and how to read in Alice’s results as input to his simulator! Alice probably already has a routine somewhere that already does that, which she has used to find statistics of the particles and their associated SSPs; it just needs to be tailored for Bruce’s purpose. However, because Alice uses IDL while Bruce does his code development in Python, the “programming language divide” further complicates the collaborative process.

With the CWB, Alice can not only share the entire directory structure that contains the particles and associated SSPs, but also share the programs to extract statistics. She can annotate these artifacts, i.e. files, directories, and programs, with their explicit meanings to help Bruce, or any other on the team, learn to use these SSPs correctly.

Probably the most innovative feature of the CWB is the “collaborative code development” capability. Alice and Bruce can log into their respective workbenches on their own computers, connect and co-develop code together. Let’s say Alice and Bruce have arranged to have such a co-development session. After they establish connection via XMPP, Bruce starts editing his Python code in the CWB for extracting IRS input out of Alice’s SSP collection. He can use the instant messaging capability of the CWB to ask Alice and obtain answers immediately to his questions. Bruce can even “share” the code editor panel with Alice. That is, both of them can edit the same code in their respective CWB editors and in real-time see what the other is doing, analogous to co-editing a Google Doc. Alice can type in the path(s) (directory locations for appropriate files) in the shared editor for Bruce since she is a lot more familiar with the directory structure. Such collaborations drastically shorten the science algorithm development time needed for locationally distributed teams to obtain simulated instrument responses.

Furthermore, what if another team member, Charlie, who is also an IRS expert but uses a radiative transfer module of a different methodology and would like to compare the simulated responses with those obtained by Bruce? In the conventional collaboration mode, Alice would now have to send Charlie her package, her documentation, and all of her communication with Bruce. In addition, Charlie basically has to repeat what Bruce had done by learning how to use Alice’s data as well as creating another copy of this data. With the CWB, Charlie can acquire that capability quickly by having a co-development session with Alice and/or Bruce where they all use the same copy of data shared by Alice in the community storage bucket. With the help from both Alice and Bruce, it should take Charlie even less time than Bruce to learn to use Alice’s data. In fact, if Charlie also uses Python for development, it is not difficult to imagine that he creates a robust working code for his IRS input preparation after just one co-development session with Bruce.

Figs 1 and 2 show how CWB tool can be used for both sharing asynchronously and real time code development collaborations.

4. Discussion

As science becomes more interdisciplinary, collaboration to share expertise and resources has become more essential. Tools that enable such collaborations within research teams and across teams--distributed both geographically and organizationally--will serve as useful productivity augmentors. With environments like the CWB, collaboration will become much easier and thus more effective. Common data sets,

tools, and results can be hosted where every member can access them, thereby reducing duplication. Dissemination of experience and knowledge can take place immediately via tags, comments, and one-on-one collaborative sessions tied directly to each piece of data, tools, or results. The overall effect will be increased efficiency and productivity. CWBs represent a firm step towards addressing these problems and achieving these goals.

One of the lessons learned during this work of adapting an exemplar science use case to promote “sharability” was the inherent difficulty involved in capturing the implicit “data management structures and rules” needed along with the science software code itself. The science code often use some implicit file/directory structures and naming conventions. As the code and data gets propagated further out into the research network, it becomes harder to correctly interpret and reuse these resources without requiring explicit collaboration, thereby hampering reusability. Asking all researchers to conform to a set of agreed-upon conventions does not appear to a practical and viable approach. New solutions that are able to log all the intended semantics with minimal input from the researchers are indeed needed to address these individual-level data/resource “sharing” problems. These solutions need to go beyond the current catalog-based metadata approaches that only support search and discovery but do not address reuse.

5. Acknowledgements

This work is funded by a NASA grant and the authors would like to acknowledge Dr. Tsengdar Lee and Mike Seabloom for supporting this research. In addition, the authors would like to acknowledge the contributions of Thomas Harris and his Exlis VIS team in helping make IDL compatible with the CWB.

6. References

[1] Bos, N., A. Zimmerman, J. Olson, J. Yew, J. Yerkie, E. Dahl, and et al. 2007. From shared databases to communities of practice: A taxonomy of collaboratories. Journal of Computer-Mediated Communication 12.

[2] www.eclipse.org/

[3] Clayberg, E., and D. Rubel. 2008. Eclipse: Building Commercial-Quality Plug-ins: Addison Wesley Professional.

[4] Norris, J.S.; Powell, M.W.; Vona, M.A.; Backes, P.G.; Wick, J.V., "Mars Exploration Rover Operations with the Science Activity Planner," Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on , vol., no., pp.4618,4623, 18-22 April 2005, doi: 10.1109/ROBOT.2005.1570832

[5] Lam, Tony, Nick Hauser, Andy Götz, Paul Hathaway, Fredi Franceschini, Hugh Rayner, and Lidia Zhang. 2005. GumTree - An Integrated Scientific Experiment Environment. In International Conference on Neutron Scattering. Sydney, Australia.

[6] Mell, P., and T. Grance. 2011. The NIST Definition of Cloud Computing: Recommendations of the National Institute of Standards and Technology: National Institute of Standards and Technology, U.S. Department of Commerce. [http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf]

[7] aws.amazon.com/ec2

[8] aws.amazon.com/s3

[9] aws.amazon.com/sdkforjava/

[10] aws.amazon.com/iam

[11] John Rushing, Rahul Ramachandran, Udaysankar Nair, Sara Graves, Ron Welch, Amy Lin (2005),ADaM: A Data Mining Toolkit for Scientists and Engineers, Computers & Geosciences 31 (5) p. 607-6

[12] Fielding, Roy Thomas (2000), Architectural Styles and the Design of Network-based Software Architectures, Doctoral dissertation, University of California, Irvine

[13] xmpp.org/xmpp-protocols/

[14] http://pmm.nasa.gov/GPM

[15] http://science.nasa.gov/earth-science/decadal-surveys/

[16] http://pmm.nasa.gov/GPM/constellation-partners

[17] http://pmm.nasa.gov/GPM/flight-project/spacecraft-and-instruments

[18] Simone Tanelli ; Wei-Kwo Tao ; Toshihisa Matsui ; Chris A. Hostetler ; Johnathan W. Hair ; Carolyn Butler ; Kwo-Sen Kuo ; Noppasin Niamsuwan ; Michael P. Johnson ; Joseph C. Jacob; Leung Tsang ; Khawaja Shams ; Sermsak Jaruwatanadilok ; Shadi Oveisgharan ; Marc Simard ; Francis J. Turk; Integrated instrument simulator suites for Earth science. Proc. SPIE 8529, Remote Sensing and Modeling of the Atmosphere, Oceans, and Interactions IV, 85290D (November 8, 2012); doi:10.1117/12.977577.

Figures

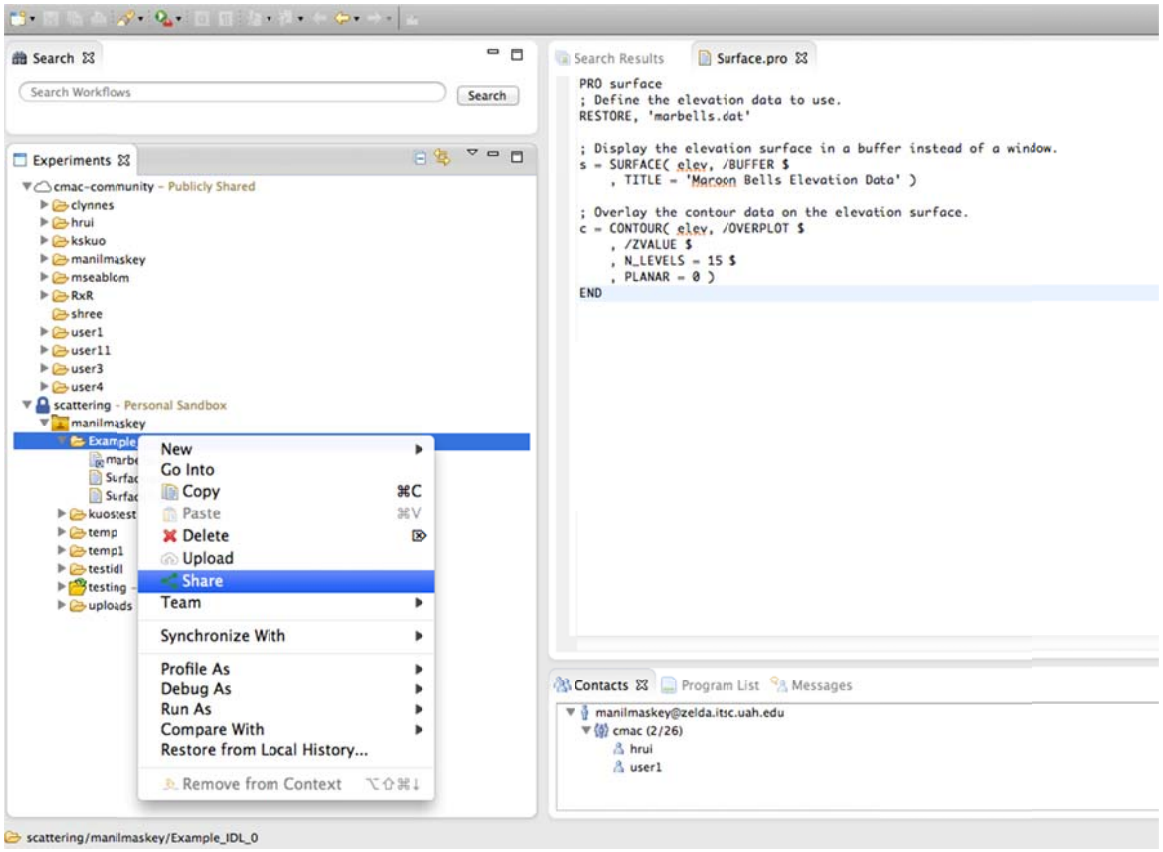

Figure 1 (A) A screen capture of the CWB showing the functionality that enables a user to seamlessly share their science experiments using a community bucket in the cloud

Figure 1 (B) A screen capture showing the built-in real time communication capability within the CWB

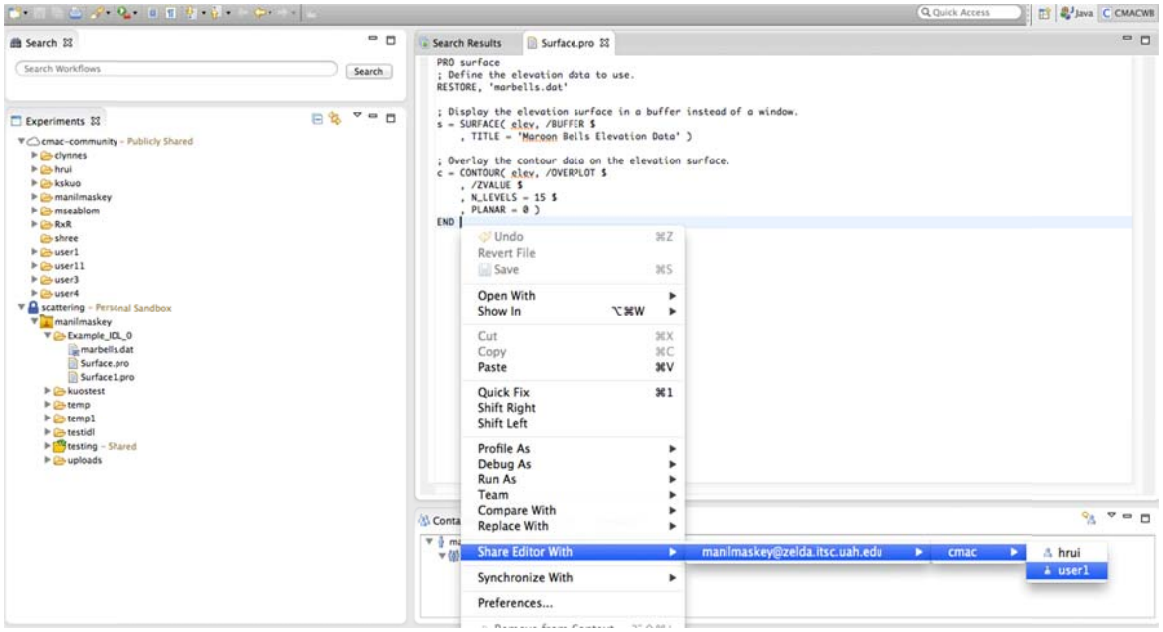

Figure 2: A screen capture demonstrating the capability to share the code editor pane with other research team members to enable real time code development