A Portal Based Interface for Compositing Multiple Streams of Experimental Data to Video

Gilead A. Kutnick
Indiana University
Pervasive Technology Labs
501 N. Morton St., Ste. 224
Bloomington, IN 47404
+1-812-856-1363
gkutnick@indiana.edu

Donald F. McMullen
Indiana University
Pervasive Technology Labs
501 N. Morton St., Ste. 224
Bloomington, IN 47404
+1-812-855-0360
mcmullen@indiana.edu

Kianosh Huffman
Indiana University
Pervasive Technology Labs
501 N. Morton St., Ste. 224
Bloomington, IN 47404
+1-812-856-1363
kihuffma@indiana.edu

Yu Ma
Indiana University
University Information Technology Services
Bloomington, IN 47405
+1-812-856-1937
yuma@indiana.edu

ABSTRACT

As an extension to our work on portals and portlets as user interfaces for remote access to instruments we are developing a set of Web Services to capture and implement tasks in a scientific workflow. The workflow described here, controlled through JSR 168 portlets, composites many different time varying data streams captured during an X-ray diffraction crystallography experiment into a single video movie useful as a permanent record. These movies are useful for debugging the experiment and as authentic research material for teaching applications. In this paper we describe the individual services, their composition into the data-to-video service, and the portal interface for using the service.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *Animations, Evaluation/methodology, Video.*

General Terms

Design, Human Factors.

Keywords

Portal, scientific workflow, Web Services, visualization, CIMA, data to video composition, GridSphere.

1. OVERVIEW

The Common Instrument Middleware Architecture (CIMA) [1] is middleware for making instruments and sensors remotely accessible by humans and software agents. In a prototypical application CIMA is used to acquire data from a number of crystallography labs around the world by connecting the diffractometer instruments and laboratory sensors in these labs to a common data management system [2]. As or after the many data streams for a diffraction experiment are collected, these can be fed into downstream cataloging and analytical applications and these processes as a whole can be organized as a number of workflows controlled and monitored through a portal interface. Because a typical experiment may last hours or days it is possible for the instrument hardware to malfunction, lab conditions to become unfavorable for the crystal being studied or even operator error to occur. As a consequence one valuable workflow is to gather and combine all of the individual data streams for a diffraction experiment into a digital video "movie" capturing all aspects of the experiment in chronological order. These videos can be quickly scanned by scientists and technicians to identify problems with the execution of the experiment before computational processes are performed, and thus avoid wasting time and resources developing structures from data that may have been collected under unfavorable conditions.

A GridSphere portal [3, 4] provides access to data management and other functions for the labs within CIMA and individual functions needed to manage and transform the data are realized in JSR 168 [5] portlets. Services such as data management, metadata searching and computational transformations needed by the portal are implemented as discrete applications with Web Services interfaces. Portlets can call the services directly or through data binding layers such as JavaServer Faces [6].

In the sections below we will describe the combination of several back-end services into a video compositing service or workflow that can be controlled through the CIMA crystallography portal. This application will be used as a model to compose other workflows related to conducting and analyzing experiments based on real-time CIMA data sources.

2. CIMA-ENABLED CRYSTALLOGRAPHY

CIMA software for crystallography consists of several components: an instrument representative service at the diffractometer (the CIMA service), a data manager that is a client to the CIMA service, a Web Services interface to the meta-data database (DataManager Web Services), and a portal-based interface [7] for users to interact with data from the diffractometer and the surrounding laboratory. This paper describes an application in X-ray crystallography, the determination of the precise 3D atomic structure of crystalline materials using X-ray diffraction techniques. [8] CIMA and the related software can be used to record any data sets consisting of images and sensor readings for the lab environment taken at specific times.

To view data from an experiment that may have progressed over hours or days "movies" of the data can be compiled. Data to be composited consist of X-ray diffraction patterns from a Charge-

Gilead A. Kutnick, Donald F. McMullen, Kianosh L. Huffman, Yu Ma, A Portal Based Interface for Compositing Multiple Streams of Experimental Data to Video, International Workshop on Grid Computing Environments 2007, Nov., 12, 2007, Reno, NV, USA. An electronic version of this document will be available at http://casci.rit.edu/proceedings/gce2007

Coupled Device detector (CCD frames), laboratory camera frames showing a view of the instrument and the operator, and images of the crystal from a microscope attached to a video camera mounted on the instrument. Lab environment sensor readings consist of various readings of temperature, relative humidity, cooling agent levels, and so on related to the operation of the instrument which are useful for diagnosing problems with an experiment. Typical problems include temperature instability in the CCD detector, changes in coolant temperature, changes in crystal temperature due to problems with the cryogenic cooling system used to maintain the crystal at a low temperature, misalignment of the crystal in the X-ray beam, and formation of water ice on the crystal due to high humidity in the laboratory. Other types of problems, such as operator error or large changes in laboratory temperature or humidity can also affect the quality of experimental results and loss of data. Many samples are unstable and difficult to synthesize so maintaining high quality results depends on monitoring and controlling a number of factors.

As described above, a system of data acquisition, processing, and data management software allows the information related to a particular sample consisting of a set of CCD frames, camera images, and sensor readings related to a particular diffraction experiment to be saved to a central repository. Exposing this collection of data through the portal and through Web Services allows it to be processed in various ways at any time either interactively by a human through the portal interface or by a program. It is in this data and service-rich environment that the movie generation software operates.

Services in CIMA workflows are coupled via document oriented and RPC Web Services calls. Services can be written in different languages as appropriate for the task. The movie generation system consists of two Web Services servers and a portal interface that use collectively gSOAP for C/C++ code, Axis client and server libraries for Java and JavaServer Faces (JSF) components, and SOAP-Lite for components written in Perl. The core movie generation component acts as both a Web Services server and as a client, for interacting with the CIMA database and providing movie generation to outside clients.

Figure 1 below illustrates how this collection of services is organized. It consists of the GridSphere portal and the CIMA Administrative Portlet that allows users to manage the state of their data collections, the DataManager Web Service and

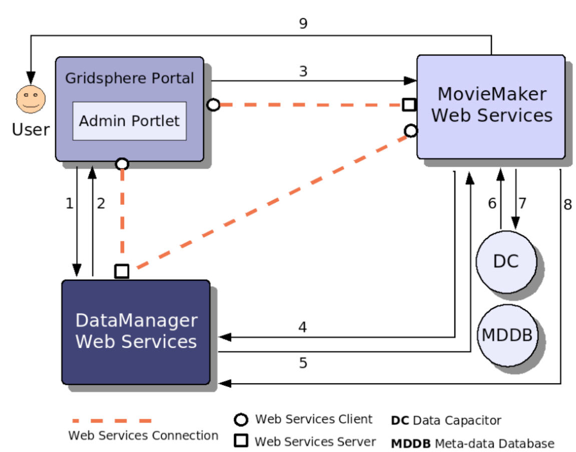

Figure 1. Video output of movie maker process

MovieMaker Web Services, the Data Capacitor (DC) storage system, and the Metadata Database (MDDB). The sequence of steps involved in generating a movie from a data set is shown as numbered arrows, and Web Services invocations are shown as dashed lines in red. The user's view of the portal is shown in Figure 2 on the following page.

3. CLIENT-SIDE TECHNOLOGY

While or after performing data collection on a sample, laboratory administrators are capable of accessing all information about the experiment for that sample via a portal interface. It is here that the option is presented to generate a movie of their experiment. Movies are stored at a central database and may also be accessed from the portal at a future time. Figure 2 below on the following page shows the portlet used as an interface to the movie maker workflow.

The CIMA portal uses the Gridsphere portal framework, where different "portlets" allow access to sample data in different fashions. In particular, the administration portlet [7] allows laboratory administrators to modify various pieces of meta-data pertaining to a sample. From this menu a "Generate movie" option is presented. Upon clicking this option users are presented with a form that describes the sample and gives several options that specify what kind of movie should be created. Options include resolution or size, quality, and format. By allowing different resolutions and quality users may choose movies that fit both space and presentation requirements. Low quality and low resolution movies are sufficient for informal viewing, while higher resolution/quality is available for detailed analysis or high bandwidth presentation.

Options to generate movies in both the "WMV" and "MP4" formats are offered via check-boxes. The decision to include both was made to guarantee that movies are playable in Windows Media Player and QuickTime respectively, the two most popular media players. The movie generator is capable of producing movies of both formats simultaneously; generating both formats at the same time is significantly more efficient than issuing two separate requests.

Additionally, an e-mail option is presented which will, upon completion of the movie, inform the user via e-mail with a location from where the movie may be downloaded. The default e-mail address is taken from the user's Gridsphere profile, but may be changed (and will remain persistently changed until the user logs out). If an error occurs during movie generation e-mails will also be sent to both the user and the movie maker administrator alerting them that the movie could not be generated. With this information the administrator is capable of manually restarting the movie generation request at a later time, after the relevant problem has been resolved.

Once a request for movie generation is submitted, the portal environment packages the request into a SOAP message using the Axis library. This message is sent to a movie maker server, which is running on a dedicated workstation. The destination server address may be configured on a per-lab basis; this configuration is done via plain text and may easily be modified. Currently four different MovieMaker servers are dedicated to support a certain number of CIMA labs, with a fixed distribution in correlation with how many samples a given lab processes per year.

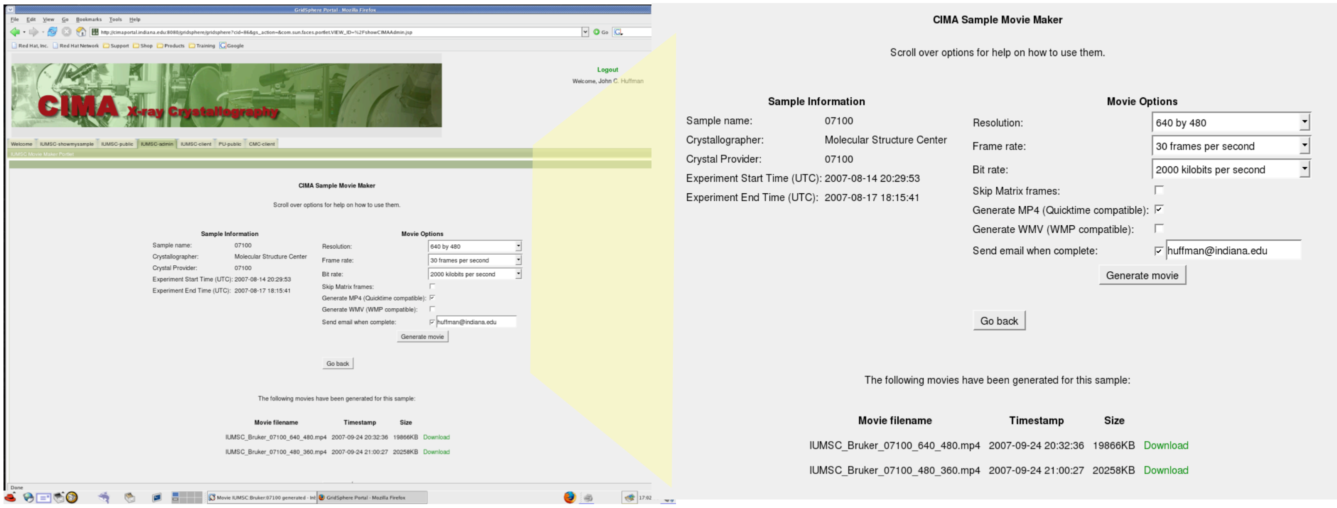

Figure 2. Portal with portlet interface to the movie maker workflow. The portlet is magnified on the right for legibility.

After movies have been successfully generated (see Figure 2) they appear as downloadable list items at the bottom of the movie generation page for that sample (Figure 1, Steps 1 and 2). Specifying a different resolution than that of a currently existing movie for the sample will result in a different movie being made, however, different quality specifications will not, hence the original movie will instead be overwritten. Currently the movie list will not be automatically updated when new movies are created, but there is a "refresh" button to allow for this to be done manually.

The movie generation portlet was constructed using standard Java Portlet Specification (JSR 168), which Gridsphere is designed to interpret. The interface was written in Java Server Pages (JSP), containing standard JSP tags along with embedded HTML. The back-end, which performs initialization, request parsing, and communication with the movie generation server, is coded in Java using the Axis client library for creating Web Services requests and implemented as a "managed bean." This back-end code interfaces with other portlet components storing data pertaining to the samples.

4. SERVER-SIDE TECHNOLOGY

Once the CIMA movie maker portlet sends out a request for movie creation to one of the movie generator servers, a server program called "ws_server" intercepts the request and processes it (Figure 1, Step 3). This is done mainly by invoking several auxiliary programs that are used to gather the sample data, composite the images and build the movie, update the database with the freshly generated movie, and send out notification e-mails.

The first auxiliary program invoked is called "get_lab_data." Its purpose is to retrieve all relevant sample information from the DataManager Web Services (DM-WS). First, the DM-WS is queried to determine specifically what parameters exist for the laboratory and instrument setup used to perform the experiment (Figure 1, Step 4). This will determine which cameras and sensor devices are available and thus should be queried. Then, the DM-WS will be queried for a list of all image files and a location of where to download them from. These files will be stored in a list, which corresponds each image file with a time-stamp. This list is made as opposed to depending on a directory traversal for purposes of easy and deterministic sorting and retrieval.

For the CCD frames and each individual camera device used to capture information about the sample a separate directory will be made where each image file will be downloaded. Downloads are done by creating a list of URLs and submitting this as input to the HTTP retrieval program "wget" [9]. This is significantly faster than submitting individual wget queries for each file. Sensor data is queried from the DM-WS and listed in a fashion similar to what is done for the image data; a file is made for each sensor device listing a time-stamp and a sensor reading value for each relevant reading i.e. those readings with time-stamps in the timeframe of the experiment.

If data have already been fetched for a previous run "get_lab_data" may be informed to not retrieve any data that has previously been obtained. This is an option specified by the Web Services call – currently it is set to on but it may be turned off (or an option may be provided in the portal). The benefit of this capability is that many movies may be generated from the same sample data without having to download the relevant data each time. The download process is typically the most time consuming part of movie generation, so this time saving can be very significant.

The next program to be invoked, "compositor", is responsible for arranging each frame of the resulting movie. Images are placed at the top portion of the frame and sensor readings are listed at the bottom in two columns. Currently a fixed layout of three image types is used, with the main one (diffractometer CCD frame) taking the upper left portion of the movie frame and two secondary images (lab and crystal cameras) positioned in the upper right (Figure 3). Up to 14 sensor readings may be included. Additionally, the primary filename and time-stamp of the CCD image are displayed, as well as a current frame number.

"Compositor" operates by displaying the CCD frame and all information that is chronologically closest to that frame. This is done because the CCD frames are considered the most fundamental information relevant to the experiment; other

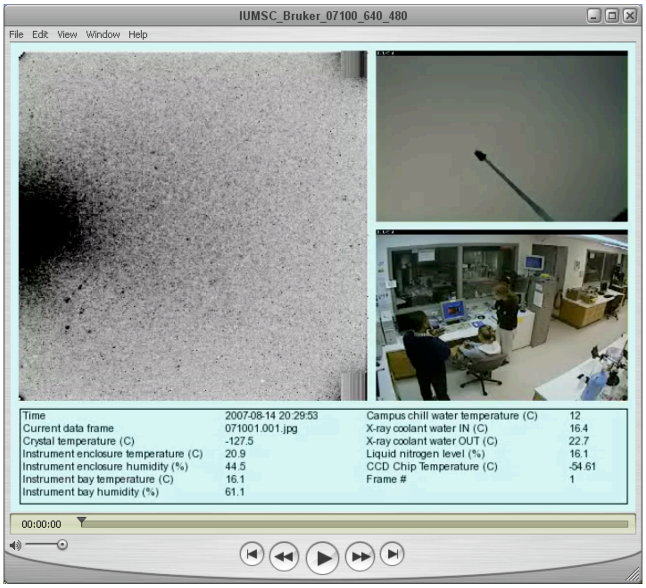

Figure 3. Video output of movie maker process.

information is supplementary. If no more information is available for the time of a CCD frame then nothing is displayed in the region where that information would normally be. Because of this it is possible for a movie to have less than the three visual images shown in Figure 3. The image compositing portion of "compositor" uses the "GD" library [10] that provides all necessary services for creating an appropriate bitmap movie frame. For each frame, captured camera images are prepared by loading the necessary JPEG files stored in the camera directories (retrieved by wget in the previous step). These images are then scaled and written to a primary buffer. Sensor data is written using a TrueType font that is included with "compositor", so their text is as legible as possible at the requested movie resolution.

After the static frame contents are generated they are added to an in-progress movie. The "FFmpeg" development library [11] is utilized to generate the movies. This includes the libraries avcodec, avformat, and swscale. By interleaving the frame generation with the movie creation process a high level of performance is achieved. Unfortunately, this means that only "single pass" movie generation may be used, which slightly diminishes the potential output quality/size. As the files in question do not contain a high degree of motion, this is not deemed to be a significant concern.

The movie compositing program was based on a similar PHP program [Piper, personal communication] designed to be executed in a standalone manner initiated by a URL sent to a web server. The process described in this paper, by virtue of being coded in C and interleaving the compositing process with the movie generation, gives better performance than the PHP-based system, while using more modern and capable movie codecs.

The next program executed, "update_database", merely informs the database that a movie is now present, or if the movie already exists performs an update of the file's attributes in the DM-WS's meta-data database. (Figure 1, Step 8) The "ws_server" program itself is responsible for copying the freshly generated movie to permanent storage (Figure 1, Step 7). This is done by writing to a direct mount of a Lustre remote file system [12] that is part of a high capacity storage system at Indiana University called the "Data Capacitor" [13].

Finally, "ws_server" will send out an e-mail to the recipient whose e-mail address is contained in the SOAP request initiating the process informing the user that the movie has been generated. (Figure 1, Step 9) Consequently, if something went wrong in the movie generation causing one of the other programs in the workflow to crash, or through the failure of an internal "ws_server" operation, a failure e-mail will be sent to the user. A diagnostic e-mail will also be sent to a predetermined address to inform the movie generator administrator of the problem.

5. DISTRIBUTION

With the advent of Internet movie broadcasting it has become possible to present these remotely generated movies to a broad audience. Websites such as SciVee [14] allow for posting of scientific videos to the scientific community (Figure 4 below). This kind of integrated flash based technology may be used to allow most web viewers to easily view the movie (thus not having to worry about compatibility issues). Furthermore, it may be possible to submit freshly created movies automatically to such a web based video repository. We believe this is an excellent example of social computing in support of scientific research and collaboration.

6. ACKNOWLEDGMENTS

Support for this work from the National Science Foundation is gratefully acknowledged through cooperative agreement and grants SCI 0330568, DBI 0446802, IIS 0513768, and IIS 0513687. Data Capacitor hardware was funded by National Science Foundation grant CNS-0521433. We thank Mr. Hao Yin, School of Computer Science, Sichuan University, Chengdu, China for developing the portlets on which this work is based. We thank Mr. Derek Piper, IU Chemistry Dept. for inspirational discussion, help and hints and development of the PHP based movie generator. We thank Dr. John C. Huffman, IU Chemistry Dept. and all the crystallographers at the various CIMA participating laboratories worldwide for their invaluable help and feedback. We thank the Open Grid Computing Environments group for assistance with the crystallography portal development.

7. REFERENCES

[1] Devadithya, T., K. Chiu, K. Huffman, D.F. McMullen, "The Common Instrument Middleware Architecture: Overview of Goals and Implementation", In Proceedings of the First IEEE International Conference on e-Science and Grid Computing (e-Science 2005), Melbourne, Australia, Dec. 5-8, 2005.

[2] Bramley, R., Chiu, K., Devadithya, T., Gupta, N., Hart, C., Huffman, J.C., Huffman, K.L., Ma, Y., McMullen, D.F. (2006) "Instrument Monitoring, Data Sharing and Archiving Using Common Instrument Middleware Architecture (CIMA)". Journal of Chemical Information and Modeling, 46(3) p.1017-25, 2006.

[3] McMullen, D.F. and Huffman, K., Connecting users to instruments and sensors: portals as multi-user GUIs for instrument and sensor facilities. Concurrency Computat.: Pract. Exper. 2006; 18:1-11.

[4] Novotny, J., M. Russell, O. Wehrens, (2004) “GridSphere: a portal framework for building collaborations”, Concurrency - Practice and Experience, 2004, 16(5), pp. 503-513.

[5] Sun Microsystems Java Community Process (2004) “JSR 000168 Portlet Specification – Final Release” (2004) http://jcp.org/aboutJava/communityprocess/final/jsr168/index.html

[6] Java Server Faces, Sun Microsystems, Inc, http://java.sun.com/javaee/javaserverfaces/

[7] Yin, H., D.F. McMullen, M.A. Nacar, M. Pierce, K. Huffman, G. Fox, Y. Ma. (2006b) “Providing Portlet-Based Client Access to CIMA Enabled Crystallographic Instruments, Sensors, and Data.” Proceedings of The 7th IEEE/ACM International Conference on Grid Computing (Grid2006), Barcelona, September 28th-29th 2006.

[8] Giacovazzo, C., Fundamentals of Crystallography, Oxford University Press, New York, 1992.

[9] GNU Wget, Free Software Foundation, Inc. http://www.gnu.org/software/wget/

[10] GD Graphics Library, Thomas Boutell, http://www.libgd.org/

[11] Ffmpeg, Fabrice Bellard, http://ffmpeg.mplayerhq.hu/

[12] Lustre Filesystem, Cluster File Systems, Inc. http://www.lustre.org/

[13] Data Capacitor, Indiana University, http://datacapacitor.iu.edu/

[14] SciVee, SciVee is operated in partnership with the Public Library of Science (PLoS), the National Science Foundation (NSF) and the San Diego Supercomputer Center (SDSC), http://www.scivee.tv

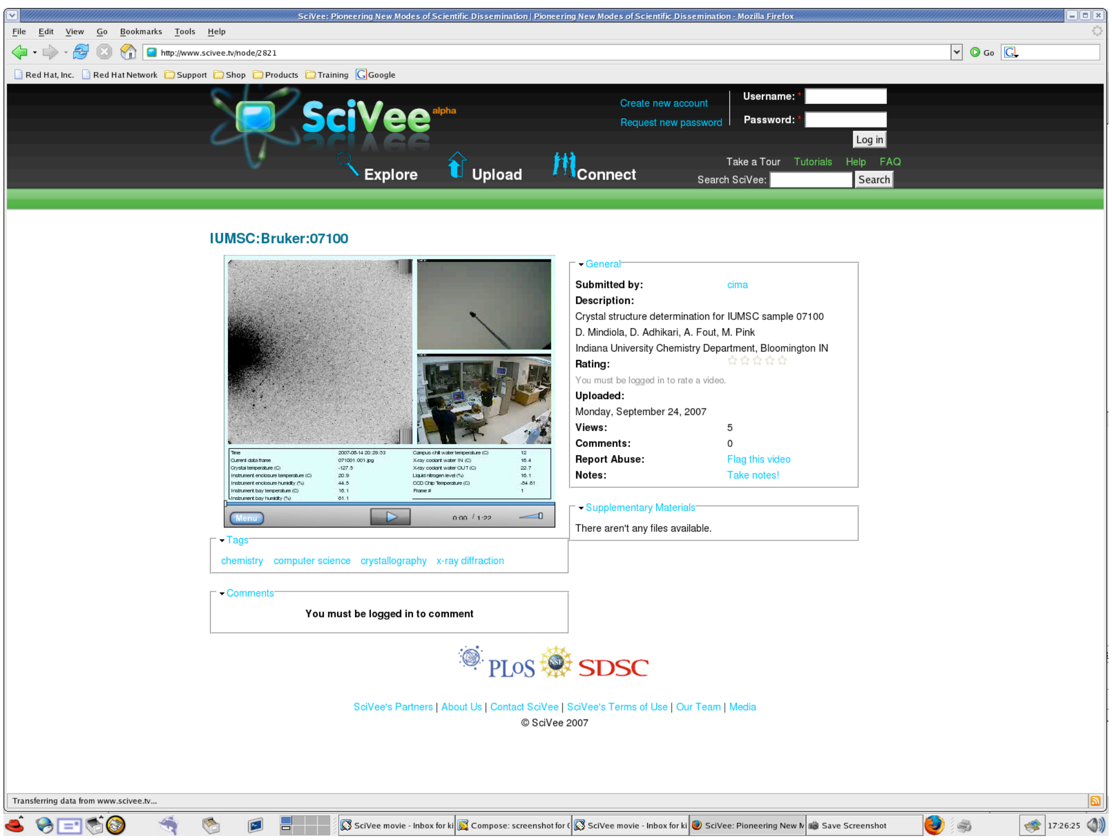

Figure 4. Publication of experiment record movies from the movie maker workflow to SciVee (view at http://www.scivee.tv/node/2821)