A Scientific Workflow Solution to the Archiving of Digital Media

Chien-Yi Hou, Ilkay Altintas, Efrat Jaeger-Frank, Lucas Gilbert, Reagan Moore, Arcot Rajasekar, and Richard Marciano[1]
San Diego Supercomputer Center, UCSD, 9500 Gilman Drive, 92093-0505 San Diego, California
{chienyi, altintas, efrat, iktome, moore, sekar, marciano}@sdsc.edu

Abstract

This work explores the challenges of embedding long-term preservation processes into existing video production workflows. The challenges are many: size of video data, bulk and parallel transfer into the archive, automation of the processes, elimination of human steps, automatic triggering of the preservation workflows, replication of content, automatic extraction of metadata, and most important mechanisms to "harden" the automation by providing recovery mechanisms in case of errors and discrepancies. We present production-quality solutions that automate this process and make use of Kepler scientific workflows and Storage Resource Broker (SRB) data grid environments.

1. Introduction

The long-term preservation of digital media has been identified as a major research challenge. This study is part of an NSF-DIGARCH [1] funded project that aims at building digital preservation lifecycle management infrastructure for the preservation of large-scale multimedia collections. The infrastructure consists of interfaces to TV production lifecycle systems, metadata definition and collection systems, and a persistent archive workflow, which preserves the material in a Storage Resource Broker (SRB) [2] data grid using the Kepler [3] system to build the workflow.

One of the major challenges in our study is the need to embed preservation processes into existing production video environments. [4, 5] Our goal is to be as unobtrusive as possible and automate the preservation processes while making them robust. We are also interested in developing methodologies for integrating these processes across data management environments by examining active (push) versus passive (pull) triggering mechanisms.

For nearly 25 years now, Harry Kreisler at the Institute of International Studies at UC Berkeley, has been conducting interviews as part of the "Conversations with History" series [6]. Over 230 guests have been interviewed, including diplomats, statesmen, soldiers, economists, political analysts, scientists, historians, writers, foreign correspondents, activists, and artists. These interviews are one-hour video-taped conversations.

Figure 1. "Conversations with History" Interview Series. *A sample set of guests. Top row: Zhores Alferov, Lakhdar Brahimi, James Fallows, Natan Sharansky, Brian Urquhart, Steven Chu. Middle row: Amartya Sen, Amy Chua, Howard Zinn, Kenneth Waltz, Noam Chomsky, Massimo D'Alema. Bottom row: Leon Panetta, Ron Dellums, Robert McNamara, John Galbraith, Evan Hoffman, Hanan Ashrawi.*

This significant "at risk" collection includes video, audio, text transcripts, web-based material, databases of administrative and descriptive metadata and contains diverse types of data, created at multiple

[1] This research was partly sponsored by the National Science Foundation and the Library of Congress. It also builds on the NSF-NPACI ACI-9619020 (NARA Supplement) research and NHPRC Persistent Archives Tesbed (PAT) project, which have both explored scientific workflows for preservation environments. Views and conclusions contained in this report are the authors' and should not be interpreted as representing the official opinion or policies, either expressed or implied, of the Government, or any person or agency connected with them.

stages within the content production workflow. The initial archive has 230 programs in 3 formats:

- Digital master files in DV format (.mov files) of typical size 12GB (compressed)
- UCTV broadcast file in MPEG format of typical size 2GB
- Web archive files in Real Player format of typical size 200 MB

The video content size is roughly 15GB per show or 230 x 15GB = 3.5TB for the complete collection. When preserving this content, we replicate the collection in at least two locations, making this a 7 TB persistently archived collection.

1.1. Scientific Workflows and SRB

Scientific workflows have recently emerged as a valuable aid to data-driven applications. Workflow environments [3, 7, 8] are available to create, edit, publish and execute workflows that analyze information about scientific problems, experimental results and problem solutions. Workflows enormously improve data analysis, especially when data is obtained from multiple sources and generated by computations on distributed resources and/or various analysis tools.

Data Grid [9] applications have diverse management requirements for handling large and complex data, for organizing data in collections (such as adding/updating datasets) and for transforming data. As these data-intensive applications often require transfer of huge amounts of data, another big issue is efficient data management across multiple storage systems.

SRB [2] is a storage management system designed for Data Grid environments. The SRB provides secure and optimized file transfer functionalities including transparent data replication, archiving, caching, and backup. Using logical name spaces, the SRB provides a uniform mechanism for seamlessly accessing data located in multiple storage systems. Moreover, SRB offers bulk data ingestion, version control and metadata query functionalities through a MetaData Catalog (MCAT).

Kepler ScientificWorkflow System. Kepler [3] is a cross-project collaboration to develop a scientific workflow system for use by scientists from multiple scientific disciplines, which supports both design and execution of workflows.

Kepler builds on top of the mature Ptolemy II software developed at UC Berkley [10]. Ptolemy II is a Java-based system which provides a set of APIs for heterogeneous hierarchical workflow modeling. The focus of Ptolemy II is to build models based on the composition of existing components, which are called *'actors'*, and manage the behavior of the model when executed using different computation semantics. Actors are the stateful encapsulation of parameterized actions performed on input data to produce output data. Inputs and outputs are communicated through ports within the actors. Kepler implements functionality on data grids by adding actors that perform desired operations. Examples include queries on metadata to find files, updating of metadata in relational database systems, transport of files from remote sites using SRB and GridFTP, remote job execution through web services, and statistical and mathematical function evaluation.

Within Kepler, actor-oriented modeling provides a way to link the functionality of different actors as a combined analytical set of steps, allowing the data to flow from one step to another through transformations at each step. These dataflow-based models can support different types of workflows ranging from local analytical pipelines to distributed high–performance and high-throughput applications, which can be data- and compute-intensive. [11] Along with support for workflow design and execution, Kepler has a number of built-in system functions including support for single sign-on grid security infrastructure-based authentication; semantic annotation of actors, types and workflows; creation, publication and loading of plug-ins using the Vergil user interface; conceptual development of hierarchical workflows that abstract the sub-workflows as an aggregate of executables; and support for on-the-fly documentation of entities at different levels of the workflow hierarchy.

1.2. Data Grid Operations in Kepler Scientific Workflow System

Data Grid workflow is the automated process of ingesting, transferring and processing data in the Grid environment. Several SRB actor interfaces have been developed in Kepler to provide efficient storage functionality in Grid workflows. These actors are developed in Kepler using the SRB JARGON java API [12,13].

Connection and Disconnection. Two specialized actors were created for connecting and disconnecting to and from a user's SRB collection (SRB space). Namely, the *SRBConnect* actor creates an authenticated socket connection using a connections pool. The connection is specified by a SRB host, port, username, password and domain. The *SRBDisconnect* actor releases the connection back to the shared pool. Within a scientific workflow all components/SRB actors accessing the same SRB space share the same connection socket by passing a connection token through the actors input and output ports via channels.

Other implemented SRB actors can be classified into the following categories: data access and transfer, server side processing of data, and an efficient search functionality using the MCAT metadata catalog. Below we elaborate on each of these categories.

Data access and transfer actors. Kepler provides several components for data access and transfer, such as *StreamPut* and *SPut*, and *StreamGet* and *SGet* to enable streaming and parallel upload and download respectively. The streaming actors read and write files from and to SRB as a sequence of byte arrays, whereas the SPut and SGet use a parallel put and get approach provided by the JARGON API. A failure in parallel Put/Get automatically activates the streaming mode for data transfer. Another actor, called *SProxy* coordinates several common proxy commands in a single actor interface. Currently the following commands are supported: list directory, copy, move, remove, replicate, create directory, remove directory and change mode. More functions can be added based on necessity.

Server side processing actor. A special actor, called *SRBProxyCommand* was developed to wrap the SRB *Spcommand*. This actor enables execution of any server side command on SRB stored data.

Metadata actors. The last set of actors implement the SRB metadata functions. Through the workflow environment a user can add metadata to a file or a collection, get the metadata and also query for files satisfying a specific set of metadata conditions. These provide Kepler with an efficient Grid resource search capability.

2. Archiving of Digital Media

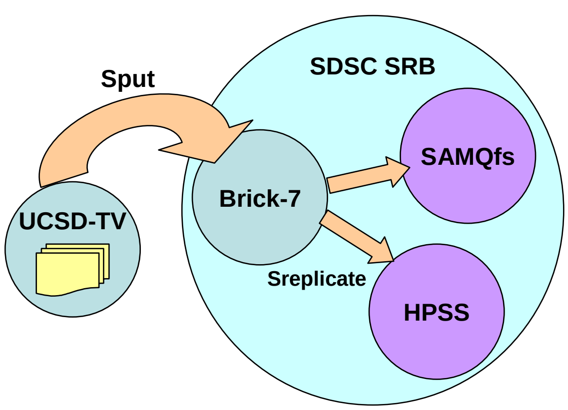

Figure 2. Video file preservation scenario

2.1 Video File Preservation Example

In the rest of this paper, we'll present an implementation of video file preservation using Kepler and SRB. UCTV [14] serves as a content provider for our scenario and is engaged in a legacy digitization conversion process. Master program tapes are being converted to digital DV format using Final Cut Pro (running on several platforms, as the conversion process runs in real-time). As digital DV files are being produced, our challenge is to move them to the SRB archive at SDSC. [15] Figure 2 illustrates the use of an "Sput" type operation, which moves files from UCTV to a grid brick commodity disk-based storage system (brick-7) at SDSC. Brick-7 is used as a staging area to hold the data, which is then replicated using the "Sreplicate" command onto two archival storage systems, SAMQfs and HPSS. Both are hierarchical tape-based storage devices. One of the benefits of using brick-7 as a staging area is much faster data transfer using parallel I/O streams over a wide-area-netework. Staging of data between brick-7 and the final destinations is done from and to machines on the same local network at SDSC. The storage capacity of brick-7 and UCTV disk are limited, so we have to remove data from both sites once the data have been replicated properly to the archive.

For preserving the "Conversations with History" video files, we divide the work into three phases:

1. Data Retrieval Phase: Due to the limited disk space capacity, it is impossible for us to keep all of the video files on the storage at UCTV. In this phase, a Kepler actor checks whether additional files have been created and are ready to be archived.

2. Data Transmittal Phase: As soon as files are identified as being complete, we transport the files to the SRB staging area and make replicas to two archival storage systems.

3. Data Removal Phase: The success of the preservation process is verified, and the files are removed from both the UCTV storage system and the SRB staging area.

2.2 Video File Preservation Workflows

The main workflow we developed is called "legacy load workflow". Its function is to identify, transport, replicate, and remove files from staging areas. In order control the legacy load workflow, we need an additional workflow, called the *"alarm clock workflow"*, to trigger the execution of the *"legacy load workflow"* (Figure 3) every day or every month. With these two workflows, we can automate the preservation process without interrupting the UCTV production process.

Data corruption errors tend to occur whenever we move or store data. The file preservation workflows need to incorporate mechanisms to detect errors and recover from them. The error recovery mechanisms are implemented as sub-workflows:

Calculating MD5 checksums: We calculate a file checksum before and after each transfer operation. These values are used to determine if the process is successful.

Feedback loops--persistent archive: Feedback loops are used to restart the preservation process in case of failure.

Detecting failures and automating recovery: We compare the checksums to see if the files have been uploaded successfully. In case of failure, we use a feedback loop to restart the process. In this case, we can guarantee the data is archived before going to the next step.

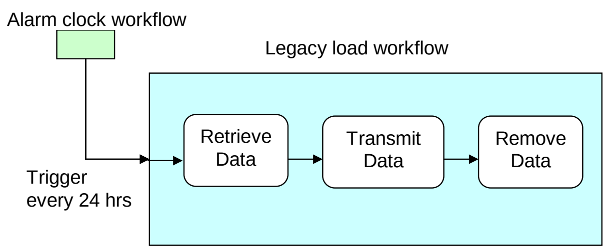

Figure 3. Conceptual legacy load workflow

Cleaning up sub-workflow: For both the UCTV local disk and the SRB staging area, we need to clean up the disk spaces after successful transmittal and replication.

3. Implementation

UCTV manages a video ID list that contains all the video IDs of the "Conversations with History" program. Our first step is to read the list and check if all three types of files are ready (.mov, .rm, and .mpeg). In Figure 4, we process one video ID from the video ID list per round. If there are than one set of video IDs in the list, we iterate until all IDs are exhausted.

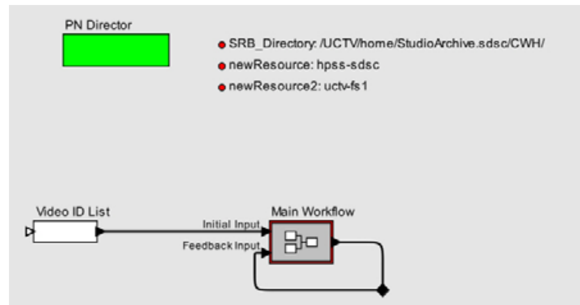

Figure 4. Legacy Load Workflow

Figure 5 shows the most important part of the workflow. From left to right, we use the **Triplet Monitor** to check the availability of videos, **Video File Retrieval** to get the files from specific directories, *Safe_Sput* to put the files to SRB, *Safe_Sreplicate* to replicate files to the archival storage systems, *Safe_Clean* to remove the files from the staging area, and **Update ID List** to remove archived files from the video ID list.

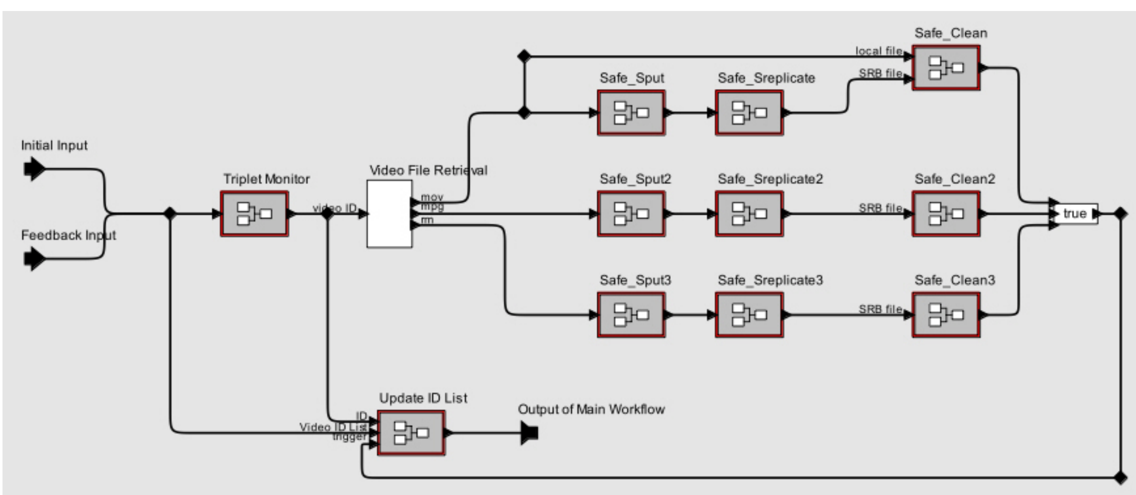

Figure 5. Main workflow

We discuss the use of *Safe_Sput*, *Safe_Sreplicate*, and *Safe_Clean* in greater detail next.

Safe_Sput: Figure 6 shows details of the *Safe Sput* workflow. Sput is an SRB actor used to upload files to an SRB-enabled storage resource. No transmission is guaranteed to be perfect, so we compute the MD5 checksum on the file before and after uploading. If the checksum values differ, we use a feedback loop to re-do the Sput operation. The way Sput makes use of a feedback loop is shown in Figure 7.

Safe_Sreplicate: As in the case of *Sput*, errors can also occur when we try to replicate the files from one resource to another, especially when the file size is large. We use an MD5 checksum again to ensure that the files are replicated perfectly. We also record these checksum values as metadata in the MCAT metadata catalog for further reference. In Figure 6, we show the use of a feedback loop to re-do the replication if the checksum values are different before and after replication. We carry out two *Sreplicate* commands sequentially in Figure 7. We also run parallel processes in *Safe_Clean*.

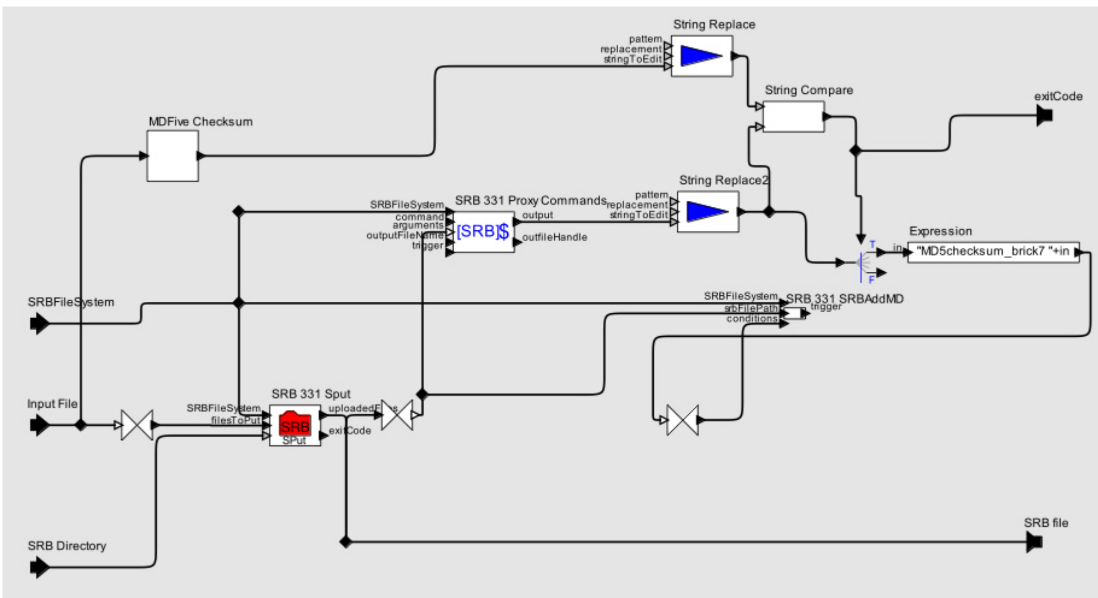

Figure 6. Detail of Safe Sput

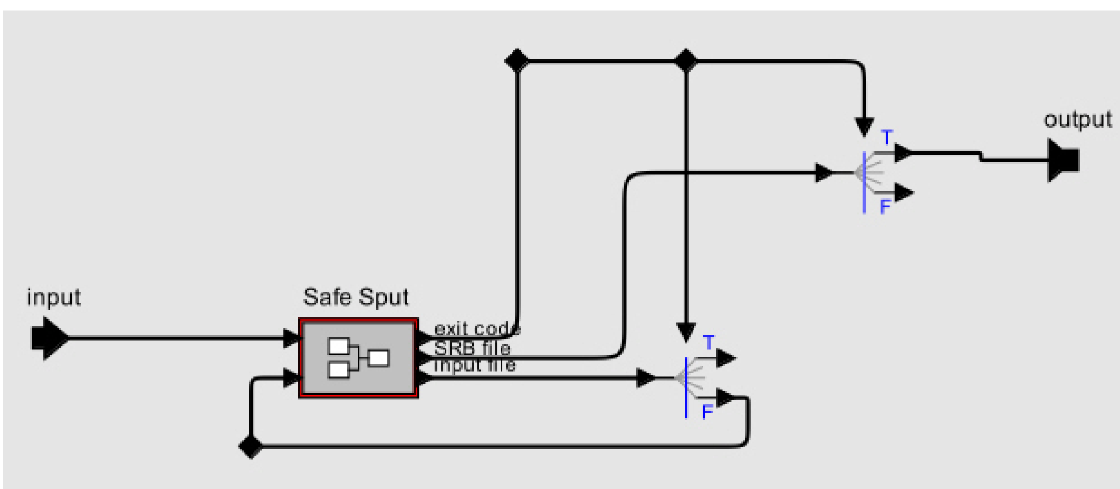

Figure 7. Feedback loop in Safe Sput

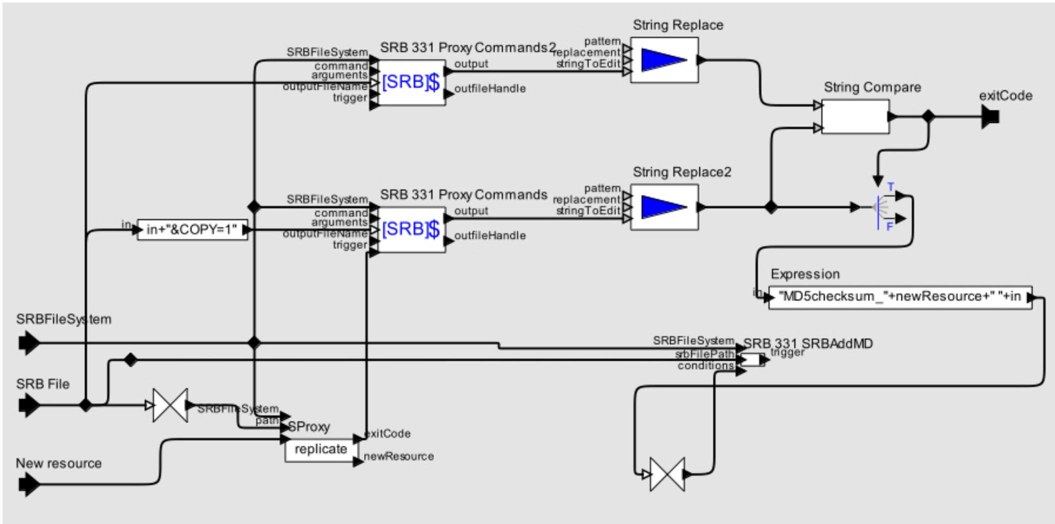

Figure 8. Detail of Safe Sreplicate

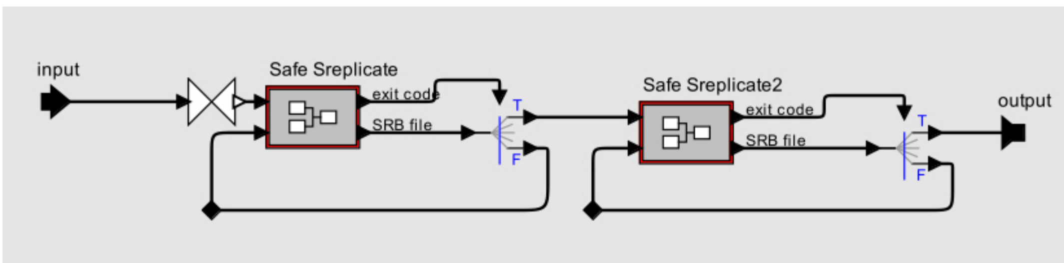

Figure 9. Feedback loop in Safe Sreplicate

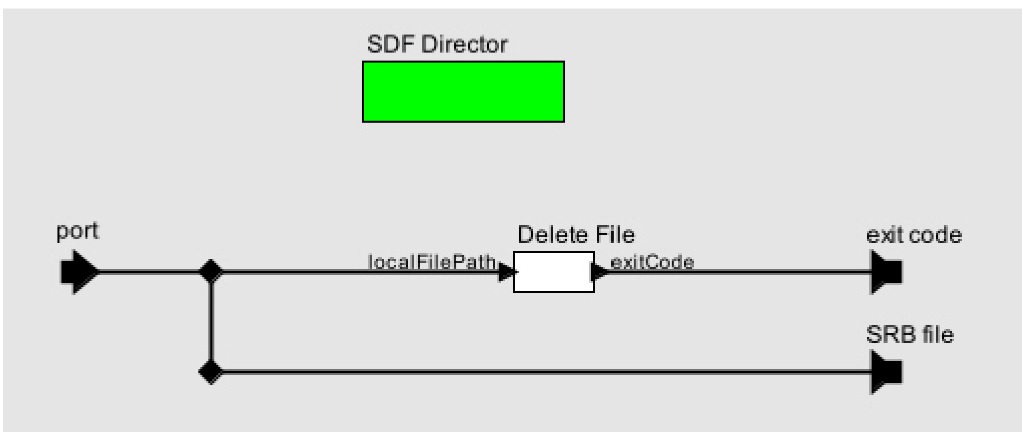

Figure 10. Detail of Safe Clean (on UCTV local machine)

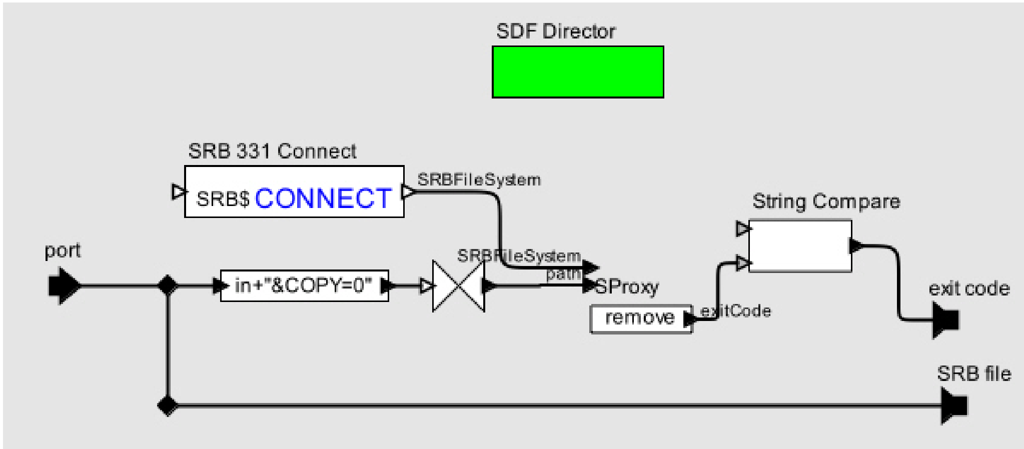

Figure 11. Detail of Safe Clean (on SRB)

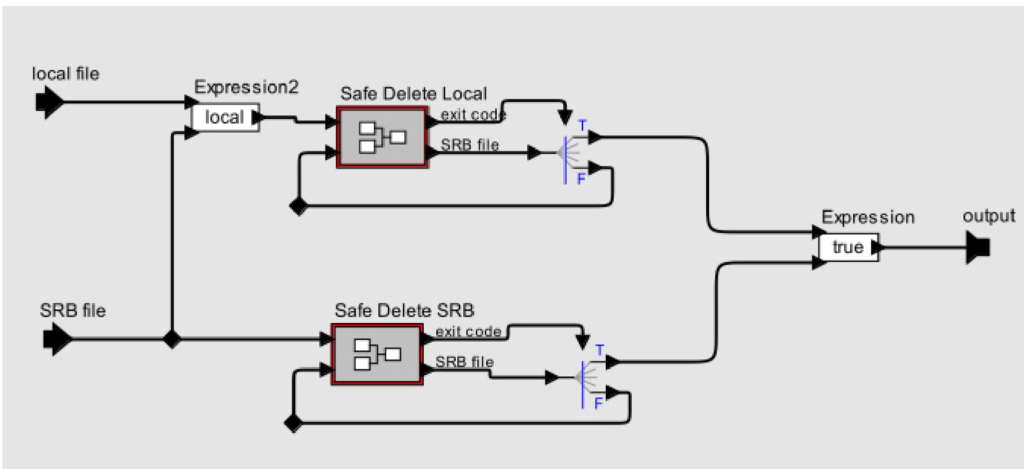

Figure 12. Feedback loop in Safe Clean

Safe Clean: Safe clean means that we can purge the files from the staging machines safely. The staging areas include the disk at UCTV and brick-7 at SDSC. Ensuring the correctness of file removal is easier than executing Sput and Sreplicate. We simply need to check for the existence of the file after removing it. Figure 10 and Figure 11 show the details of removing files on the UCTV local disk and from the SRB respectively. Here we perform two remove file actions in parallel (Figure 12).

4. Evaluation and Discussion

The goal of the approach described in this paper is to fully automate the time-consuming and error-prone collection building steps, which are often done manually. Using the clock mechanism workflow, daily triggering of the preservation processes is done. We use parallel I/O transfer with multiple threads (I/O channels) to minimize the time needed to move the data. We measured parallel transfer times from the UCTV eMac machine into a staging grid-brick in the SRB at SDSC. The slowest transfer time will be through a "Sput –rv" which is a single-threaded (sequential) loading of a file into the archive. The larger DV master file takes 1902 seconds with this command and only 1309 seconds using three parallel I/O channels. The corresponding MPEG2 file transfer time decreases from 284 seconds to 180 seconds and the smaller RealPlayer file transfer time from 32 seconds to 22 seconds. These timings are given for a representative Conversations with History 1-hour program. The different Sput transfer options correspond to:

- **-m** for server-initiated parallel I/O (in this case the server specifies the number of I/O channels automatically – this is the only option allowed when using the Java-based Jargon library)
- **-M** is client-initiated parallel I/O (also allows for

explicit specification of the number of threads (-N option))

Table 1. Parallel transfer rates

	Sput -rv	***Sput –rvM (4 threads)***	***Sput –rvMN2 (2 threads)***	***Sput –rvMN3 (3 threads)***	***Sput -rvm (4 threads)***
DV master	1902 sec	1696 sec	2359 sec	1309 sec	1826 sec
MPEG2	284 sec	238 sec	326 sec	180 sec	263 sec
RealPlayer	32 sec	27 sec	37 sec	22 sec	31 sec
				25 min 11 s.	

Table 1 illustrates that a 3-thread parallel transfer can move 50 sets of videos per day (.mov, .rm, .mpeg). However the time needed to convert a digital master tape to DV format is only an hour. So, conversion is the bottleneck, not transmission to the staging area.

Table 2 shows the time for a SRB "Sreplicate" command to replicate an existing SRB object onto a physical (storage) resource. Note that hpss-sdsc is the IBM High Performance Storage System, and uctv-fs is the Sun Sam-QFS file system. Also note that the time to replicate a file into Sam-QFS is greater than the time needed to transfer a file from UCTV to San Diego, indicating the importance of using a staging disk area at SDSC.

Table 2. Archival file loading time

	Sreplicate –S hpss-sdsc	***Sreplicate –S uctv-fs***
DV Master	7 min 13 sec	30 min 43 sec
MPEG2	1 min 20 sec	4 min 8 sec
RealPlayer	10 sec	28 sec
	~9 min	***~35 min***

5. Summary

We have implemented the mechanisms that are needed to embed preservation processes into production video workflows. Making our preservation processes robust enough to withstand "real-world" requirements is a challenge that pushes the limits of the tools themselves.

A second challenge is to create generic workflows that can be reused in other preservation projects. This requirement has already led to the development of a more modular workflow design, with simpler actors that can be reused in other workflows. To fully test the success of this aspect of the project, we are developing a library of data management modules.

The workflow examples we presented provide mechanisms to handle well known data management problems, as well as support for manipulation of SRB collections. Specifically, the workflows support:

- loading of a collection (registration of files),
- transport of files,
- replication of files,
- synchronization of files, and
- validation of checksums.

The workflows are built from Kepler actors that encapsulate SRB collection commands with additional actors that manage error recovery and recursion.

6. Future Work

Future work includes the integration of these Kepler/SRB workflows with digital library services. The UCSD Libraries XDRE (eXtensible Digital Resources Environment (prounounced "extra") is a scalable framework for content management. XDRE is implemented in Java, uses the SRB for storage, is based on an RDF representation mechanism, and provides a simple schema for encapsulating extensible metadata in XML and a parametric URL mechanism for retrieving the XML. This last mechanism will be exploited next in this project. We intend to use a simple web form-based import mechanism to populate the XDRE content management system with state information the Kepler workflows and dynamically record workflow activity logs for the legacy ingestion workflows. The XDRE web-server offers real-time snapshots of the state of the conversion process.

Once we have "hardened" the approaches discussed in this paper we will also focus on:

- Adding preservation metadata to the archive.

- Modifying the legacy workflows to accommodate dynamic creation of new videos using incremental-update workflows. We expect many of the legacy workflow components to be reusable.
- Incorporating a second archiving input stream by adding the TV transcripts as they are being added from UC Berkeley

As a part of future work, we also plan to extend the SRB functionality in Kepler. Currently Kepler supports a few *SRB Proxy* commands which help in data archive management and retrieval. However we intend to add more SRB SCommands (e.g. *Serror* to display error and *SgetColl* to display information on SRB data objects) to existing native Kepler-SRB actors. We also plan to design an experimental SRB domain in Kepler, which optimizes data transfers and provides connections at the system level so that the user doesn't have to concentrate on details of the interaction of the workflow system with SRB. Finally, we are starting to add additional "products" to the archive: the transfer of interview transcripts in particular. These add additional complexity into the existing workflows for managing a larger number of file types.

7. Acknowledgments

This work is collaboration between SDSC, UCTV, UC Berkeley and UCSD Libraries. SDSC partners also involve Francine Berman (co-PI). UCTV partners include: Lynn Burstan (co-PI), Steve Anderson, Matt Gaffney, Melissa Weber, and Bee Bornheimer. UC Berkeley involves Harry Kreisler. UCSD Libraries include: Brian Schottlaender (co-PI), Luc Declerck, Brad Westbrook, Arwen Hutt, Ardys Kozbial, Chris Frymann, and Vivian Chu.

Authors would also like to thank to the Kepler team, especially Bertram Ludaescher, Nandita Mangal, Bing Zhu and Tim Wong, for their collaboration and support.

7. References

[1] NSF DIGARCH program: solicitation and 1ist of 10 projects funded: http://www.nsf.gov/pubs/2004/nsf04592/nsf04592.htm and http://diggov.org/library/library/dgo2005/digarch/index.jsp

[2] SRB, Storage Resource Broker, Version 3.1, http://www.sdsc.edu/dice/srb, 2004.

[3] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger-Frank, M. Jones, E. Lee, J. Tao, Y. Zhao, "Scientific Workflow Management and the Kepler System", *Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows*, to appear, 2005. See also: http://kepler-project.org/

[4] A. Rajasekar, R. Moore, F. Berman, B. Schottlaender, "Digital Preservation Lifecycle Management for Multimedia Collections", ICADL 2005, The 8th International Conference on Asian Digital Libraries, December 12-15, 2005, Bangkok, Thailand. See also: http://www.icadl2005.ait.ac.th/program.htm

[5] R. Marciano, C.-Y. Hou, L. Burstan, H. Kreisler, R. Moore, A. Rajasekar, "Long-Term Preservation of Large-Scale Multimedia Collections: a Digital Workflow Approach", Archiving 2006, May 23-26, 2006, Ottawa, Canada. See also: http://www.imaging.org/conferences/archiving2006/

[6] H. Kreisler, "Conversations With History", UC Berkeley, Institute of International Studies, http://globetrotter.berkeley.edu/conversations/

[7] T. Oinn, M. Greenwood, M. Addis, M. N. Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, D. Marvin, P. Li, P. Lord, M. R. Pocock, M. Senger, R. Stevens, A. Wipat and C. Wroe, "Taverna: Lessons in creating a workflow environment for the life sciences", accepted for publication in Concurrency and Computation: Practice and Experience Grid Workflow Special Issue. See also: http://taverna.sourceforge.net/ .

[8] D. Churches, G. Gombas, A. Harrison, J. Maassen, C. Robinson, M. Shields, I. Taylor and I. Wang, "Programming Scientific and Distributed Workflow with Triana Services", in Grid Workflow 2004 Special Issue of Concurrency and Computation: Practice and Experience, to be published, 2005. See also: http://www.trianacode.org/ .

[9] A. Rajasekar, M. Wan, R. Moore, G. Kremenek, and T. Guptill, "Data Grids, Collections and Grid Bricks", 20th IEEE/ 11th NASA Goddard Conference on Mass Storage Systems & Technologies (MSST2003) San Diego, California, April 7-10, 2003.

[10] Information on the Ptolemy Project, See Website: http://ptolemy.eecs.berkeley.edu/ptolemyII/

[11] I. Altintas, A. Birnbaum, K.K. Baldridge, W. Sudholt, M. Miller, C. Amoreira, Y. Potier, B. Ludaescher, "A Framework for the Design and Reuse of Grid Workflows", Lecture Notes in Computer Science, Scientific Applications of Grid Computing: First International Workshop, SAG 2004, Beijing, China, September 20-24, 2004, Volume 3458 (3), pp 119-132, ISBN3-540-25810-8.

[12] Java Api for Real Grids On Network (JARGON), http://www.sdsc.edu/srb/jargon/

[13] N. Mangal, E. Jaeger-Frank, I. Altintas, C.-Y. Hou, L. Gilbert and A. Rajasekar , "Storage Resource

Broker Actors and Applications in Kepler", SRB Workshop, February 2-3, 2006, San Diego, CA.
[14] UCSD-TV, http://www.ucsd.tv/
[15] A. Rajasekar, R. Marciano, R. Moore, C.-Y. Hou, F. Berman, L. Burstan, S. Anderson, M. Weber, B. Bornheimer, H. Kreisler, B. Schottlaender, L. Declerck, B. Westbrook, A. Hutt, A. Kozbial, C. Frymann, V. Chu, "Building a Demonstration Prototype for the Preservation of Large-Scale Multimedia Collections", SRB Workshop, February 2-3, 2006, San Diego, CA. See also: http://www.sdsc.edu/srb/Workshop