VisTrails: Workflow Management System on NEX

Petr Votava, Andrew Michaelis, Ramakrishna Nemani, Forrest Melton

NASA Ames Research Center/
University Corporation at Monterey Bay

with contributions by David Koop, Claudio T. Silva (NYU Poly) and the VisTrails team

NASA Earth Exchange

Outline

1. Goals
2. Workflows Background
3. VisTrails Overview
4. Demo
5. VisTrails and NASA Earth Exchange (NEX)

Goals

- Provide background on workflows with emphasis on the VisTrails workflow management system.
- Demonstrate some of VisTrails features

(Not an in-depth tutorial, but rather an introduction to a software package that provide capabilities for provenance, transparency, reproducibility, and collaboration)

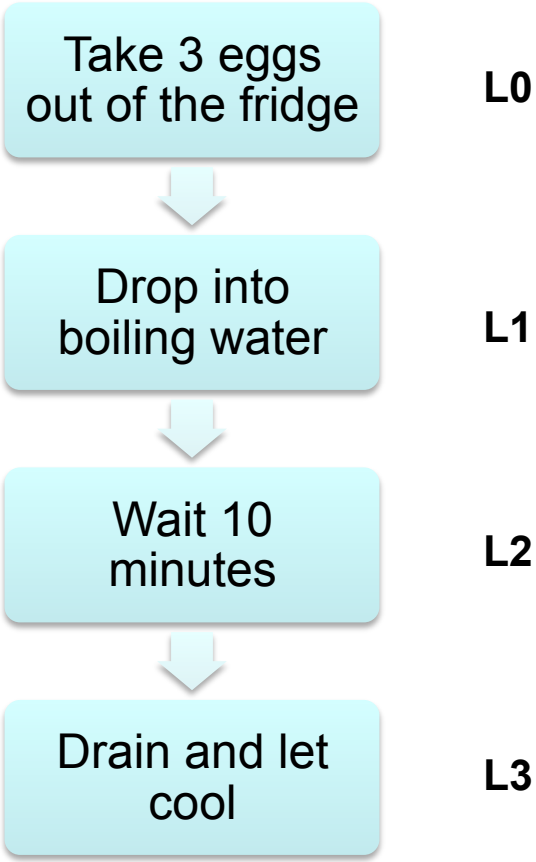
Some Terminology

- Workflow
  - Basically sequence of (connected) steps
  - Originated in manufacturing – think assembly line
- Workflow Management System
  - Manages the execution of a workflow (flow of data and information)
  - Enables users to define and execute new workflows.
- Workflow Engine
  - Implement the coordination (orchestration) of the steps in a workflow
  - Dozens of workflow engines (Business, manufacturing, science, …) – can be hard to navigate

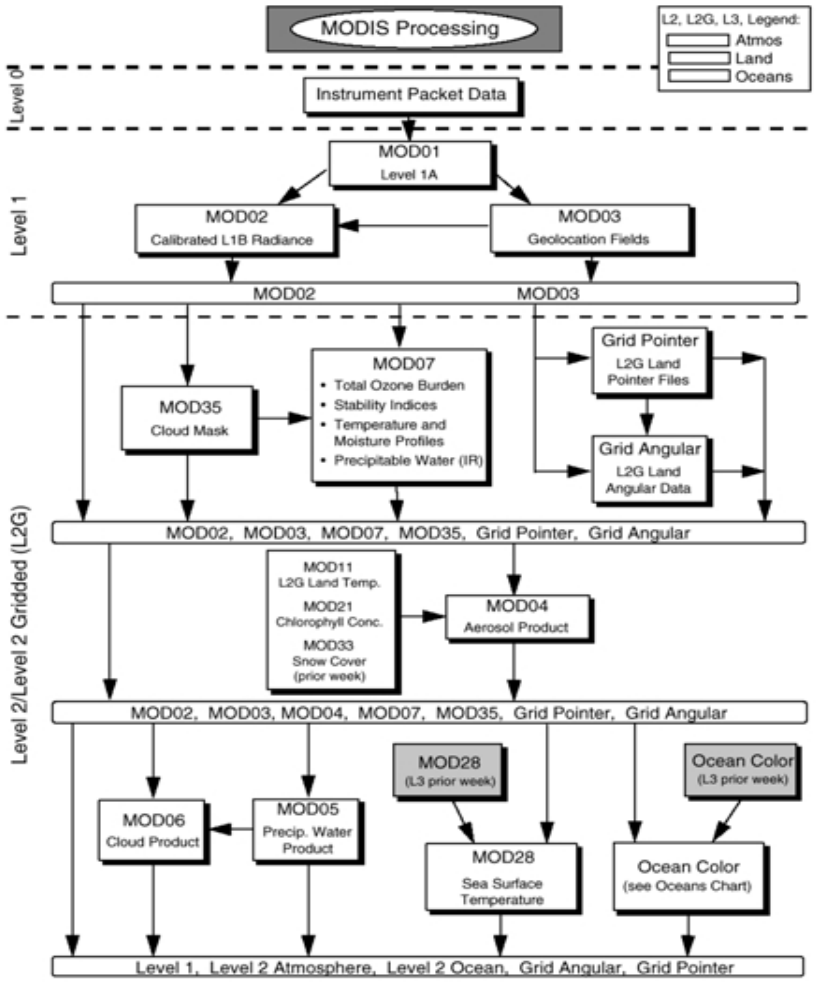
Some Examples

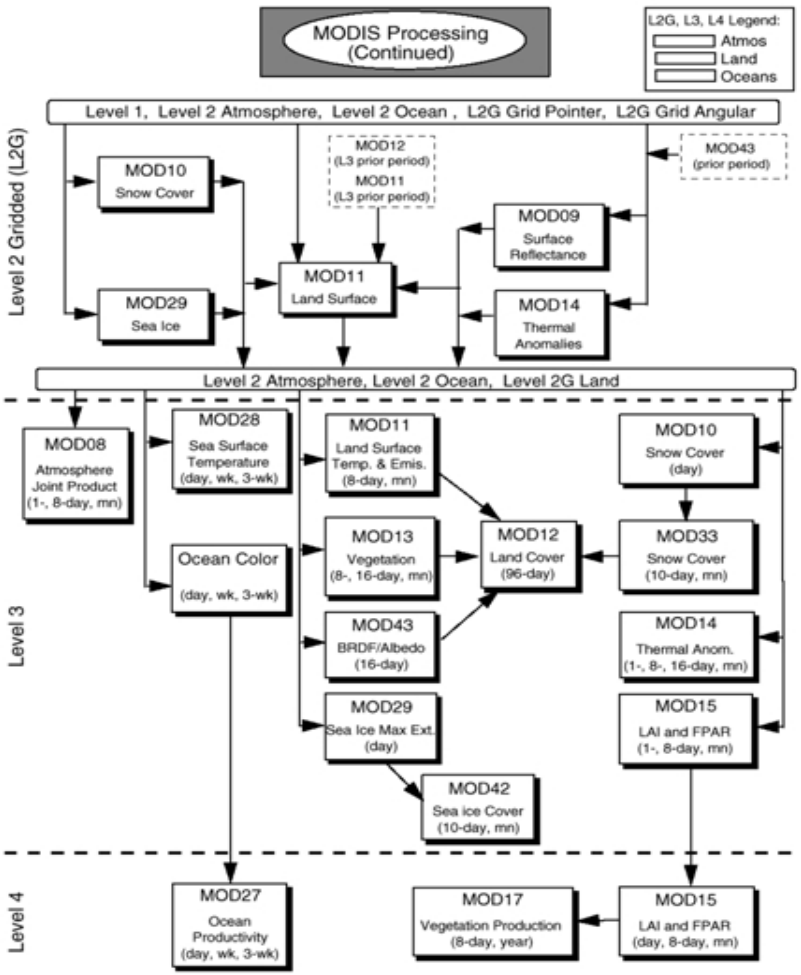
- Apache ODE
  - For execution of business processes written in WS-BPEL standard
- Taverna Workbench
  - Graphical interface for building and execution of workflows
  - myExperiment collaboration site
- SciFlo (JPL)
  - Web services orchestration of scientific workflows
  - Can move operations to data – this may still be problematic if your data is distributed in many places.

Workflows can be simple …



… or complicated

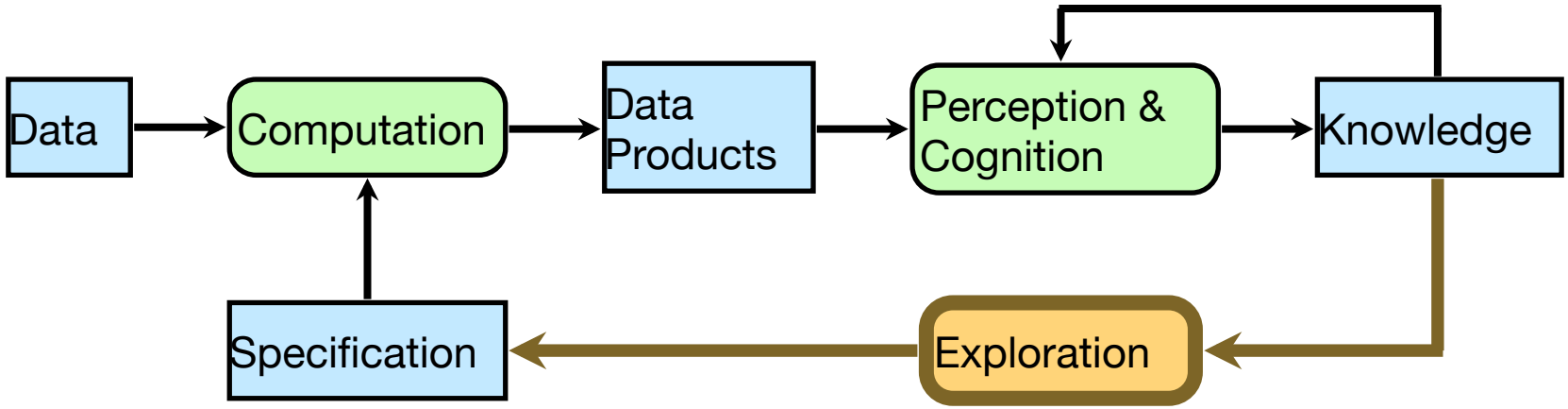




Why VisTrails

- Provides built-in provenance capabilities
- Python based
  - Just as a majority of the NEX platform development
- Excellent graphics capabilities
  - Important to convey the message
- **Easy integration of legacy software**

Data Analysis and Visualization



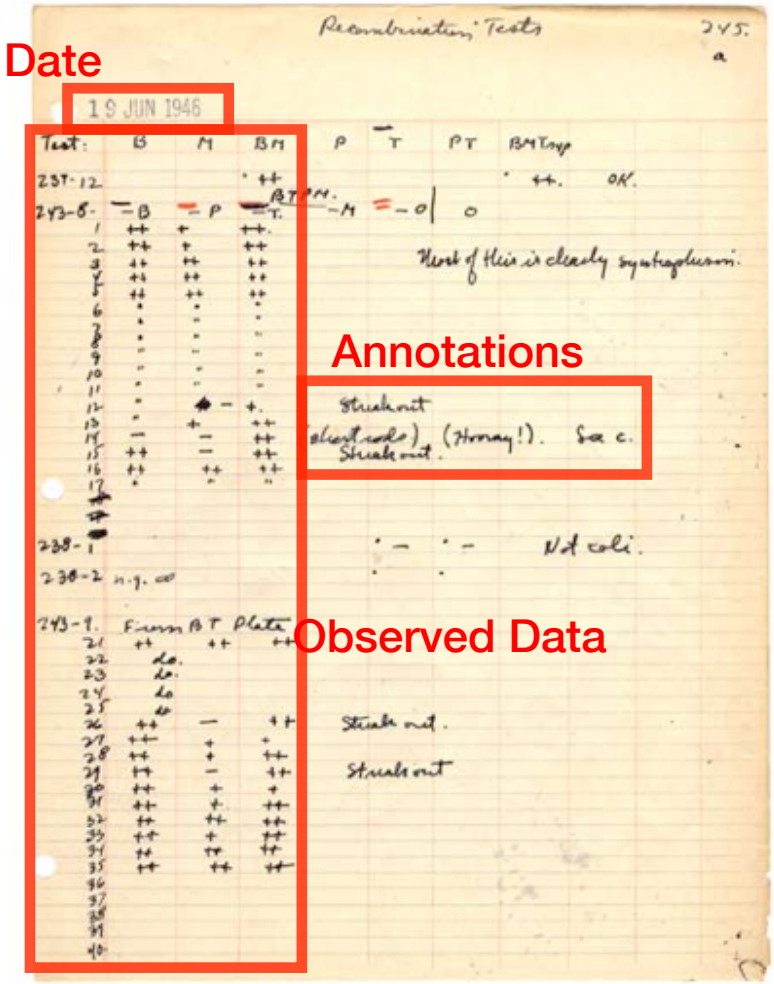
[Modified from Van Wijk, Vis 2005]

"Reflective thought requires the ability to store temporary results, to make inferences from stored knowledge, and to follow chains of reasoning backward and forward, sometimes backtracking when a promising line of thought proves to be unfruitful. The process takes time." – Donald A. Norman

[Courtesy of D. Koop, VisTrails]

Provenance in Science

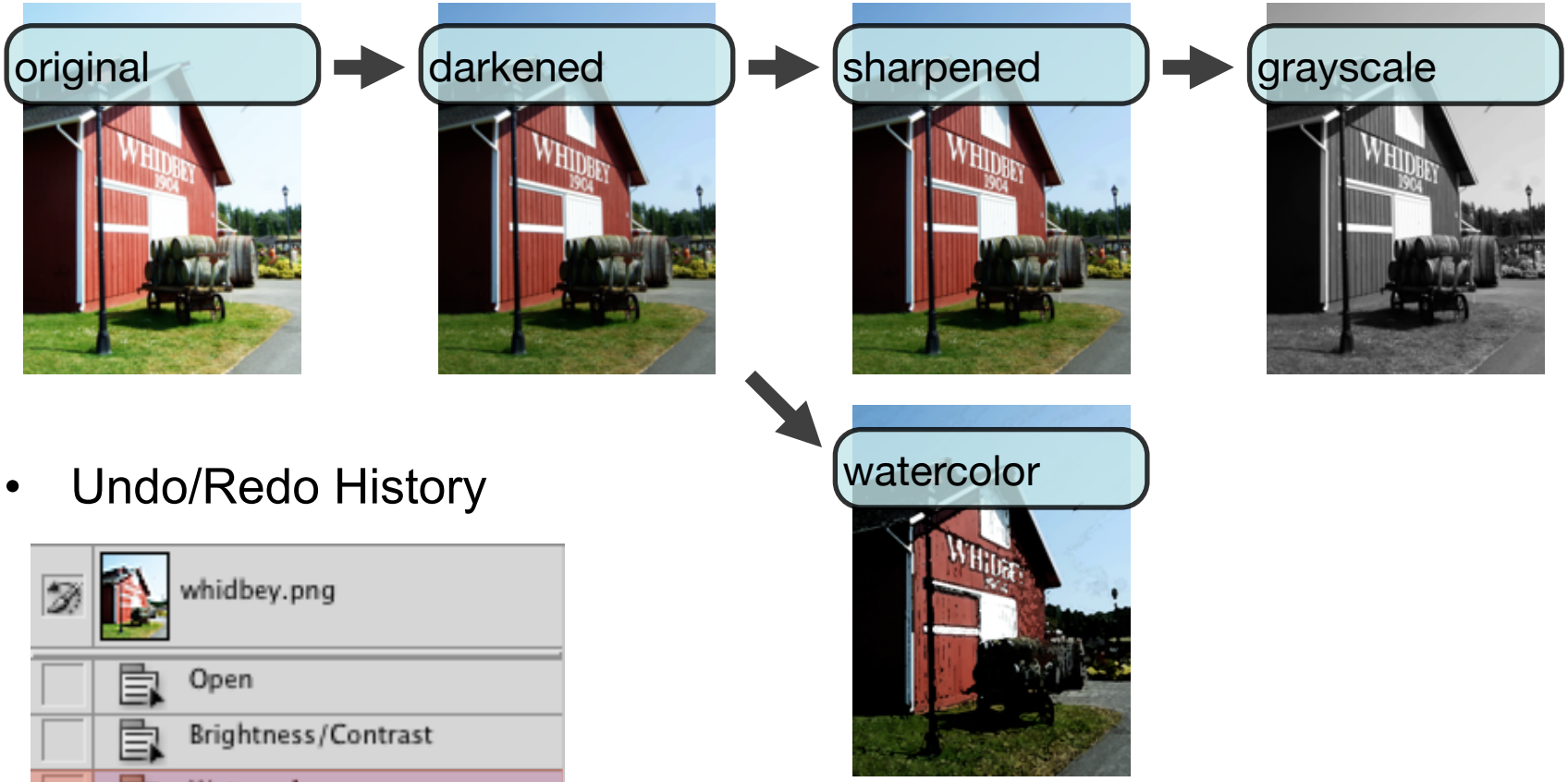
- Provenance: the **lineage** of data, a computation, or a visualization
- **Provenance is as (or more) important as the result!**
- Old solution:
  - Lab notebooks
- New problems:
  - Large volumes of data
  - Complex analyses
  - Writing notes doesn't scale



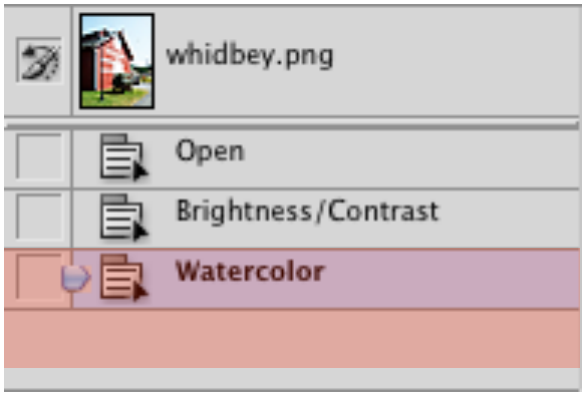
[DNA Recombination, Lederberg]

Simple Provenance Example: Photo Editing

- User Actions



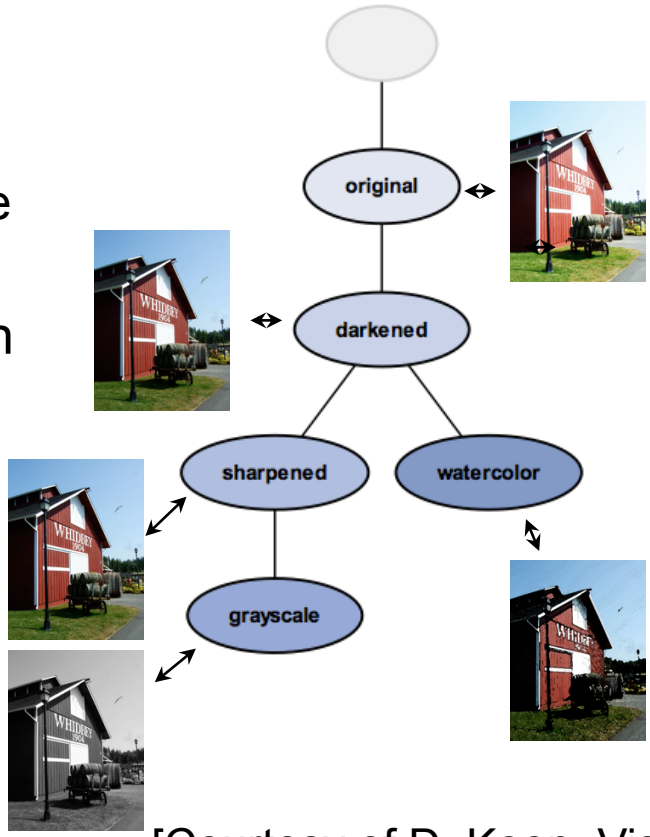
- Undo/Redo History



[Courtesy of D. Koop, VisTrails]

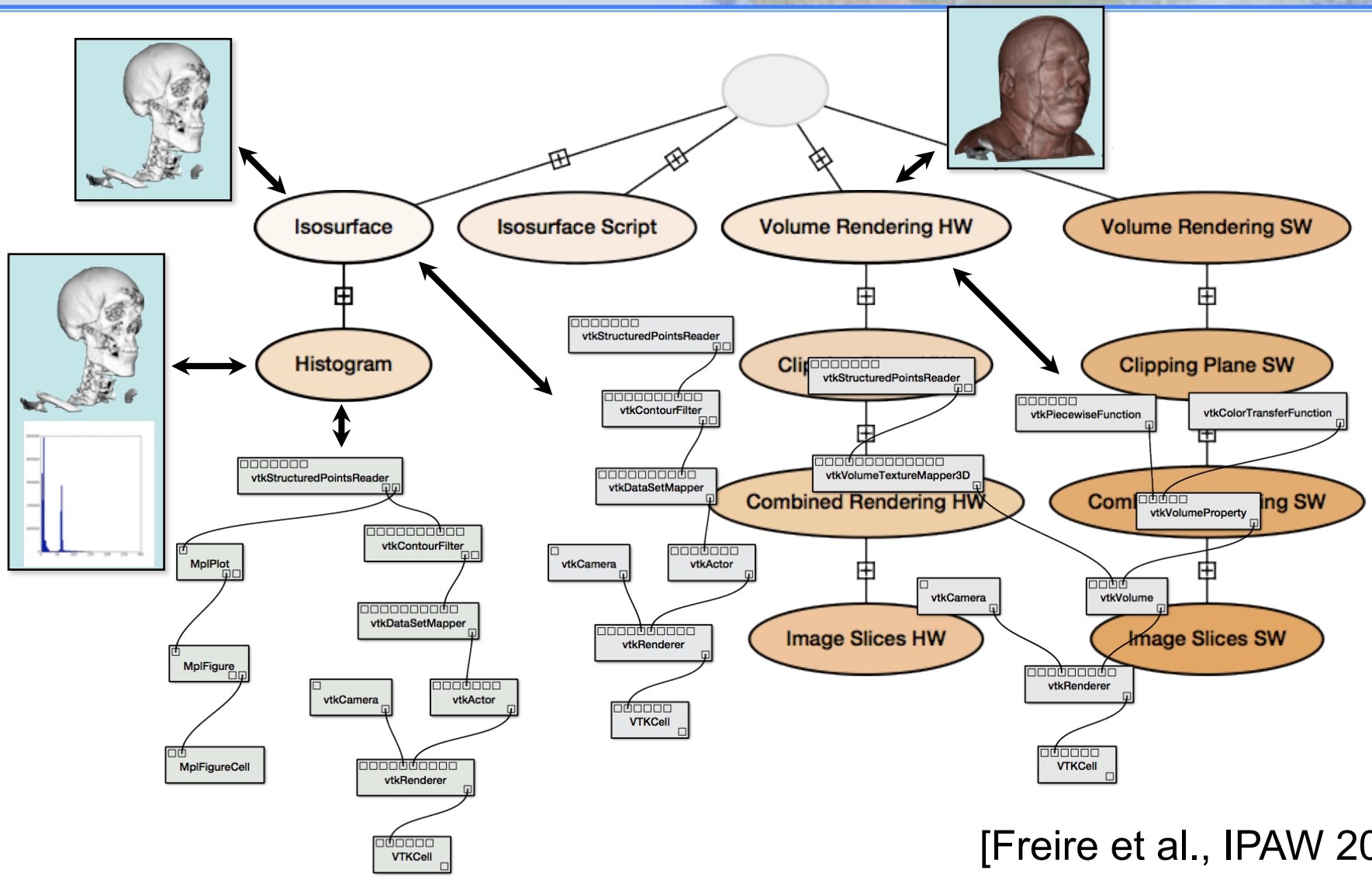
VisTrails Change-based Provenance

- Undo/redo stacks are linear!
- We lose history of exploration
- Old Solution: User saves files/state
- VisTrails Solution:
  - Automatically & transparently capture entire history as a tree
  - Users can tag or annotate each version
  - Users can go back to any version by selecting it in the tree



[Courtesy of D. Koop, VisTrails]

Change-based Provenance



[Freire et al., IPAW 2006]

NASA Earth Exchange

The VisTrails System

- Comprehensive **provenance** infrastructure for computational tasks
- Focus on **exploratory** tasks in collaborative environment
  - **Collaborative** design of scientific processes
  - Interactive parameter space exploration
- **Transparently** tracks provenance of the discovery process—from data acquisition to visualization
  - The **trail** followed as users generate and test hypotheses
  - Users can refer back to any point along this trail at any time
- Leverage provenance to **streamline exploration**

VisTrails: Specifications

- Open-source, freely downloadable system (www.vistrails.org)
- Multi-platform: users on Mac, Linux, and Windows
- User's guide, wiki, and mailing list
- Many users in different disciplines and countries
- You can try it **today** and be up and running in **minutes** (takes a little longer on some Linux systems because you have to build some dependencies)
- You can explore all examples from today's demo – they are part of the VisTrails distribution.

Now, let's try a demo

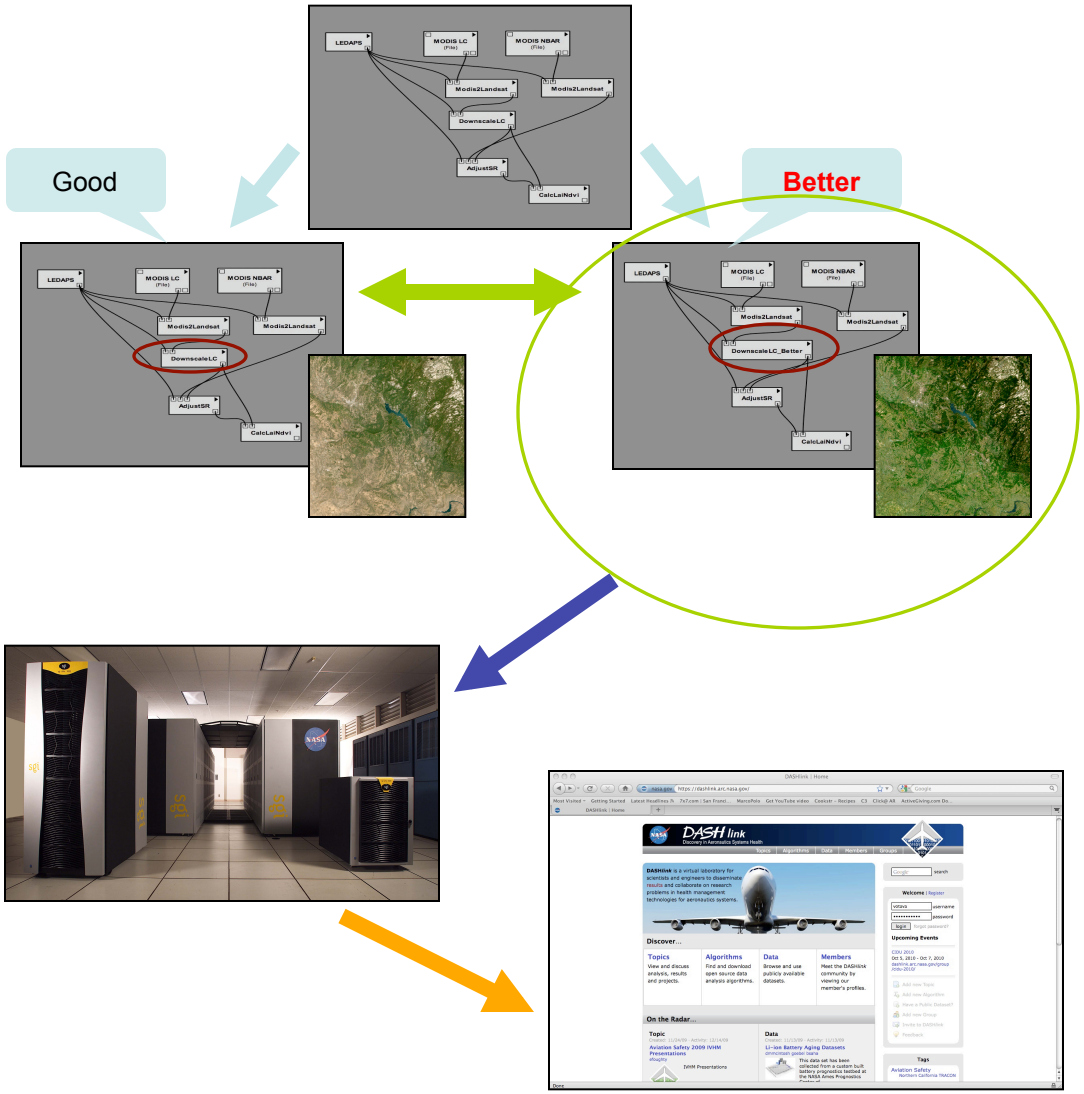
NEX Goal:
Collaborative Process Design and Knowledge Capture

Process Design

Iterative Refinement/
Distributed Collaboration

Full-scale Execution

On-Line Result Sharing and
Process Re-use

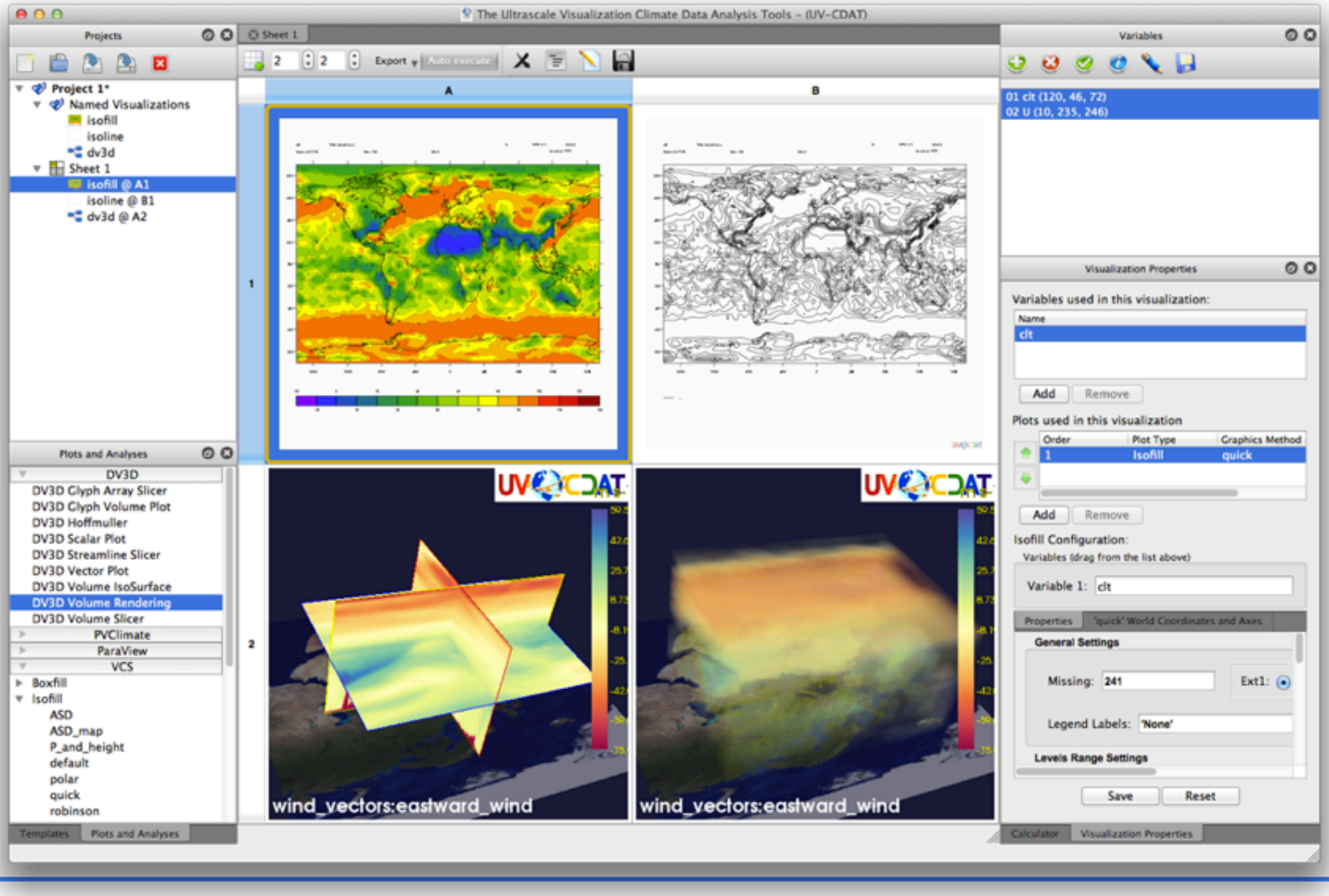


Additional NEX Integration

- Integration of VisTrails components with NEX semantic components
  - Bringing together provenance and semantics
- Developing workflow engine to interface with the Pleiades supercomputer
  - Running VisTrails on large scale problems, not just exploration any more

Example of VisTrails Integration: UV-CDAT

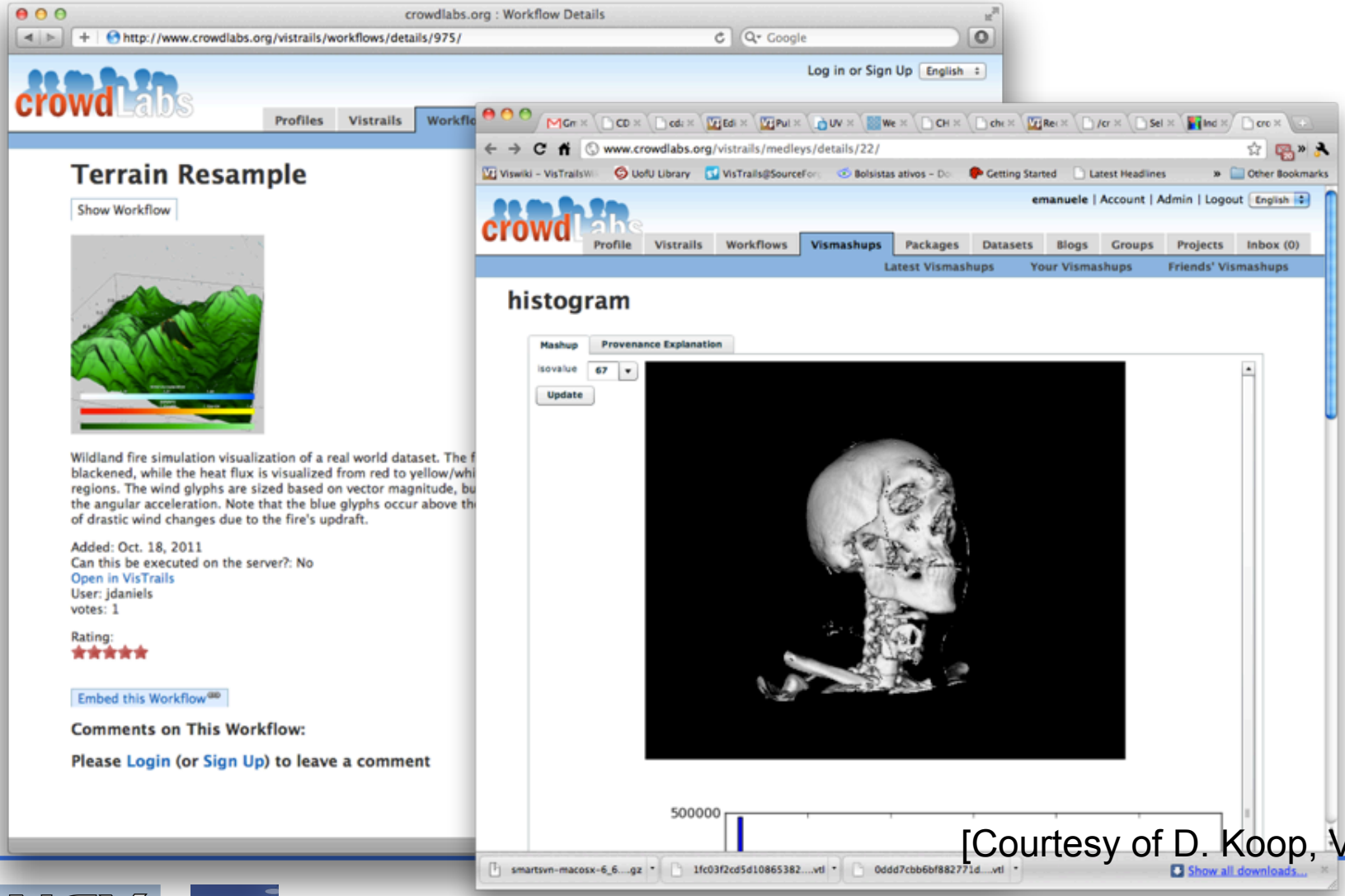
Ultrascale Visualization – Climate Data Analysis Tools (Lawrence Livermore)



NASA Earth Exchange

VisTrails Collaboration Site: crowdLabs

Planned integration with the NEX portal



[Courtesy of D. Koop, VisTrails]

NASA Earth Exchange

Executable Papers

The ALPS project release 2.0:
Open source software for strongly correlated systems

http://adsabs.harvard.edu/abs/2011arXiv1101.2646B

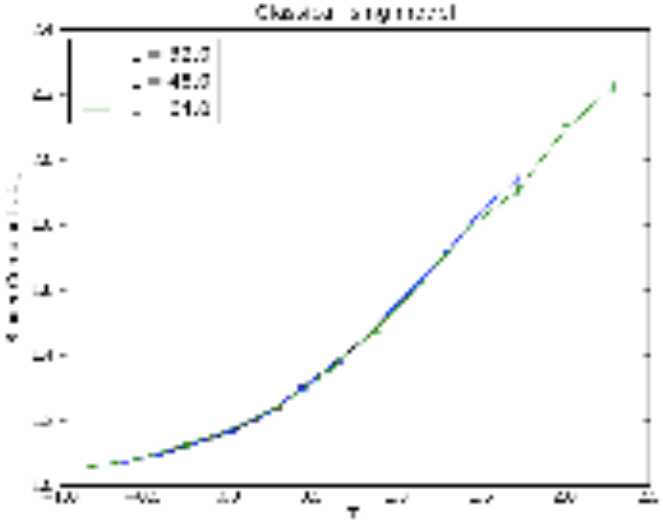


Figure 3. In this example we show a data collapse of the Binder Cumulant in the classical Ising model...

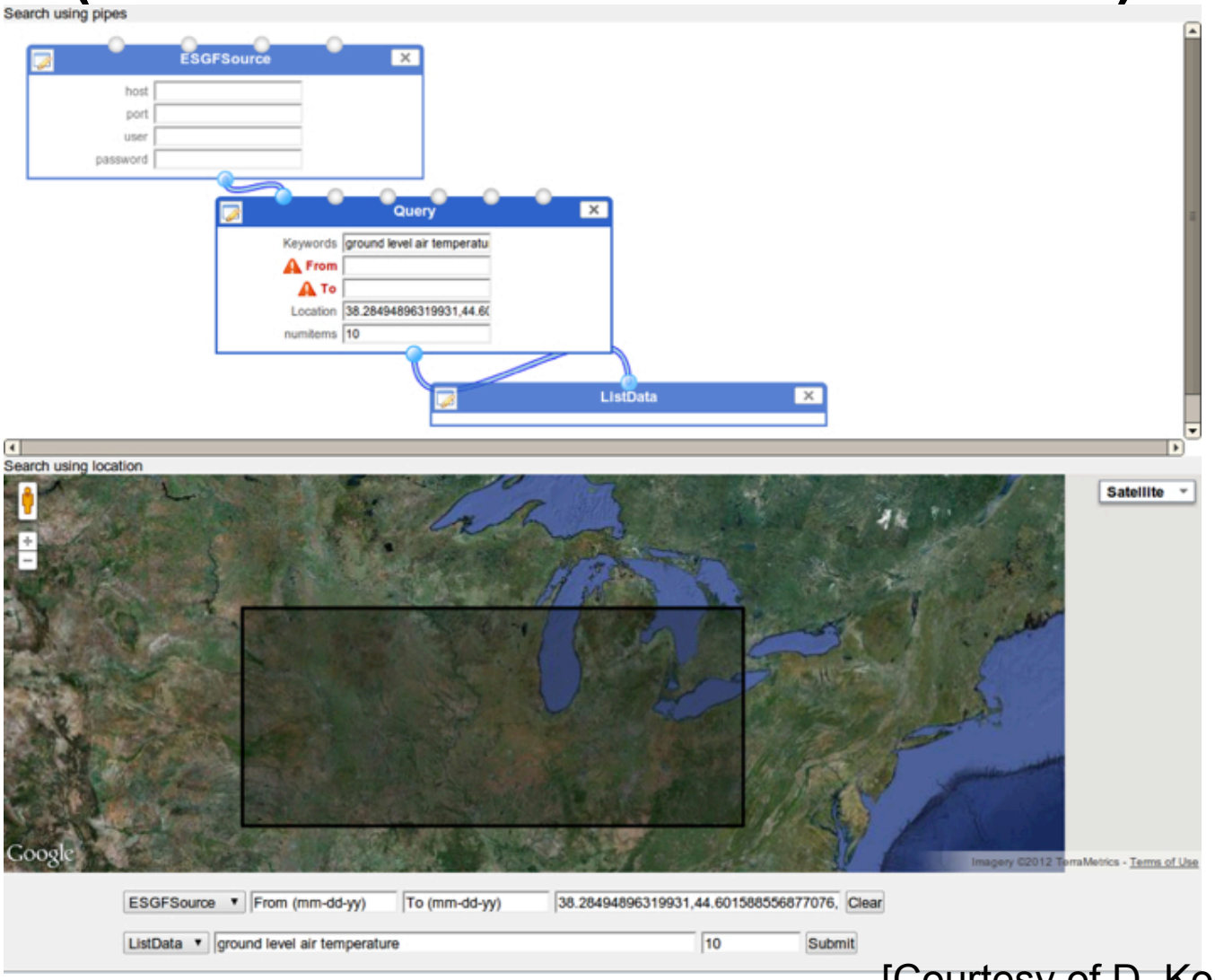
LaTeXcommand used to embed plot:

```
\vistrail[wfid=935,
  buildalways=false]{width=1.0\linewidth}
```

[Courtesy of D. Koop, VisTrails]

NASA Earth Exchange

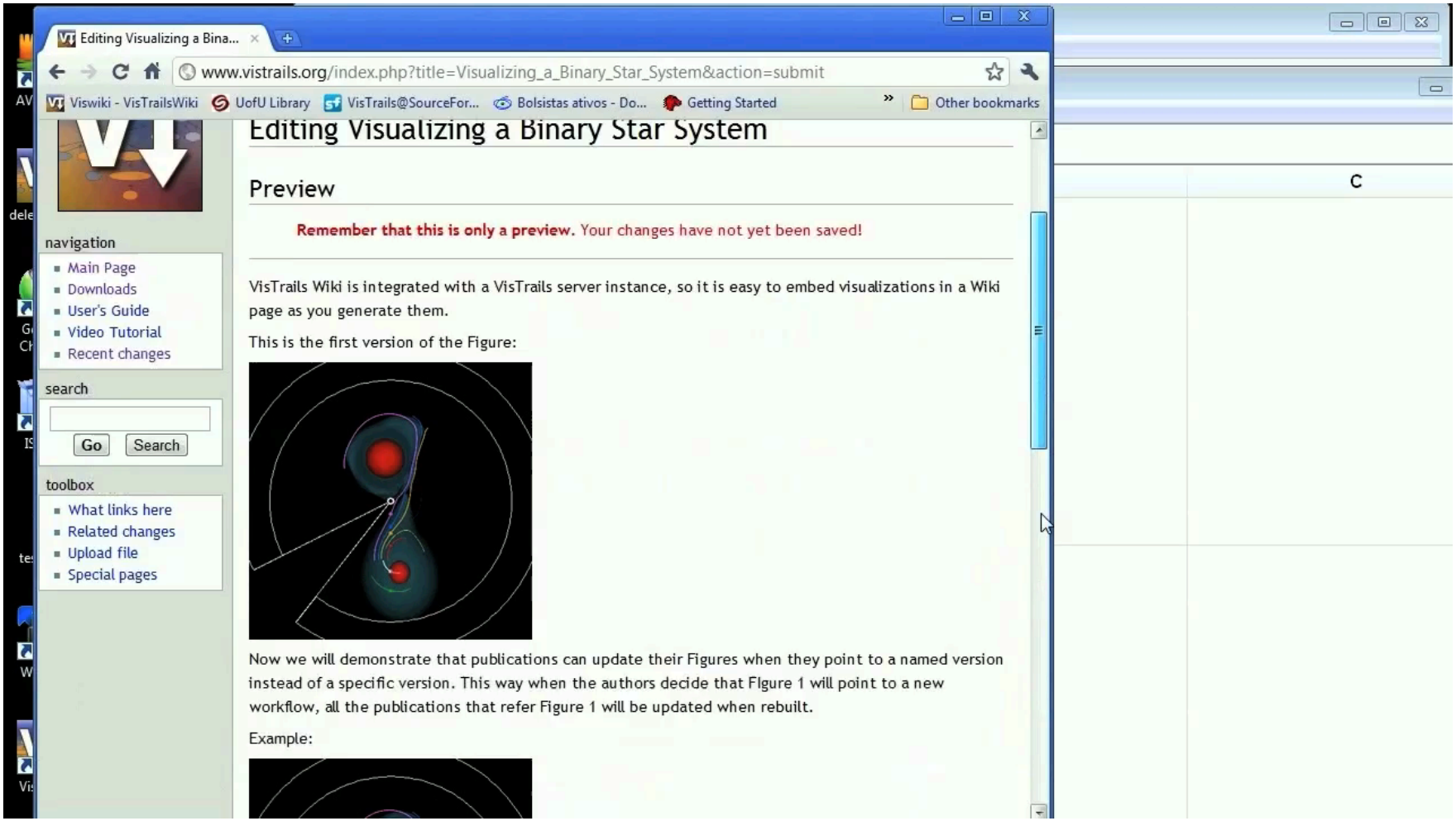
Future Work: ClimatePipes (web-based interface to VisTrails)



[Courtesy of D. Koop, VisTrails]

NASA Earth Exchange

Web-based Publishing and Interaction



[Courtesy of D. Koop, VisTrails]

Scripting Support

```
import vt_setup

import core.api
vt = core.api.get_api()
text = vt.get_package("org.vistrails.text")
basic = vt.get_package("edu.utah.sci.vistrails.basic")
m1 = text.TextFile(value="/vistrails/climate/a.txt")
m2 = text.TextFile(value="/vistrails/climate/b.txt")
m3 = text.MergeFil
m3.addFile(m1.valu
m3.addFile = m2.va
m4 = basic.FileSin
m4.overwrite = Tru
m4.outputPath = "/vistrails/climate/c.txt"
m4.file = m3.outputFile
vt.tag_version("Merge2")
vt.execute()
vt.save_vistrail("/vistrails/climate/helloworld2.vt")
```

Provenance and Structure Automatic!

[Courtesy of D. Koop, VisTrails]

Final Thoughts on Reproducibility

- Workflow is just one of the components
  - It's a good start, but it is not enough
- Need to capture system information (OS, database, libraries) and configuration (versions, configuration options, …)
- Virtual machines play an important role
- Great reproducibility info

http://vgc.poly.edu/~juliana/talks/reproducibility-tutorial2012.pdf

Thank You

http://www.vistrails.org/