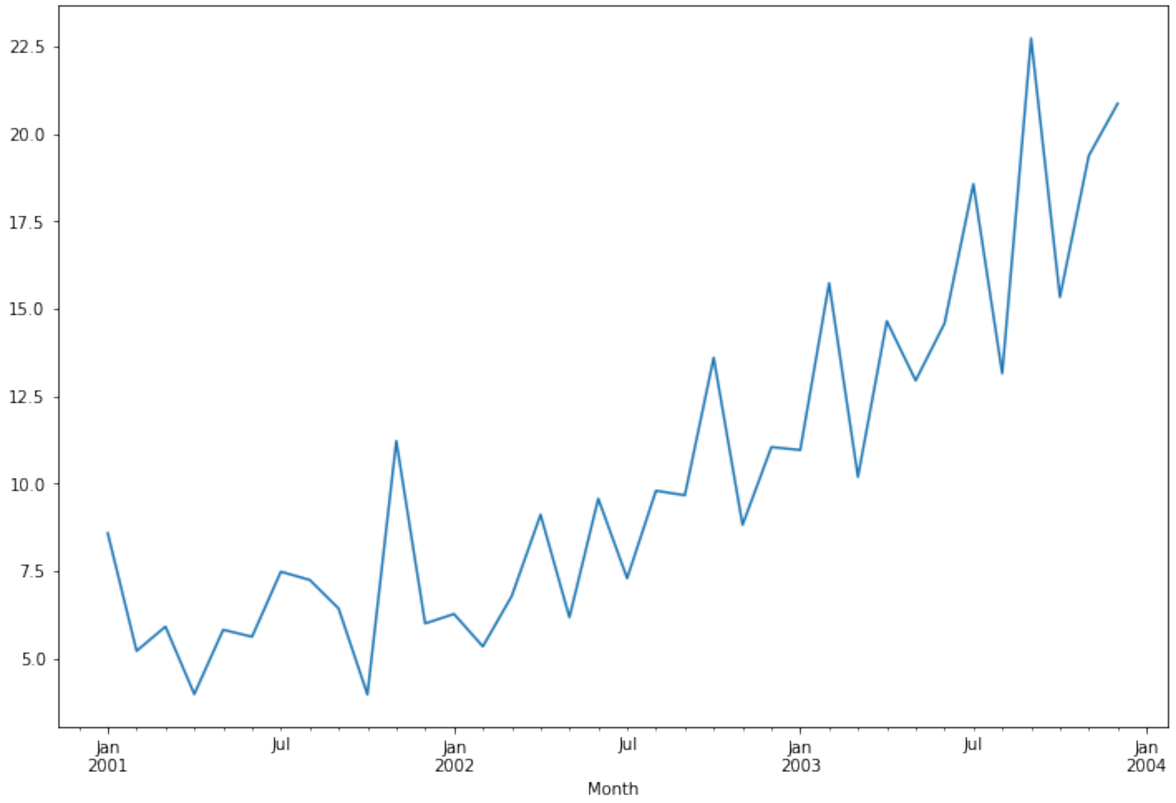
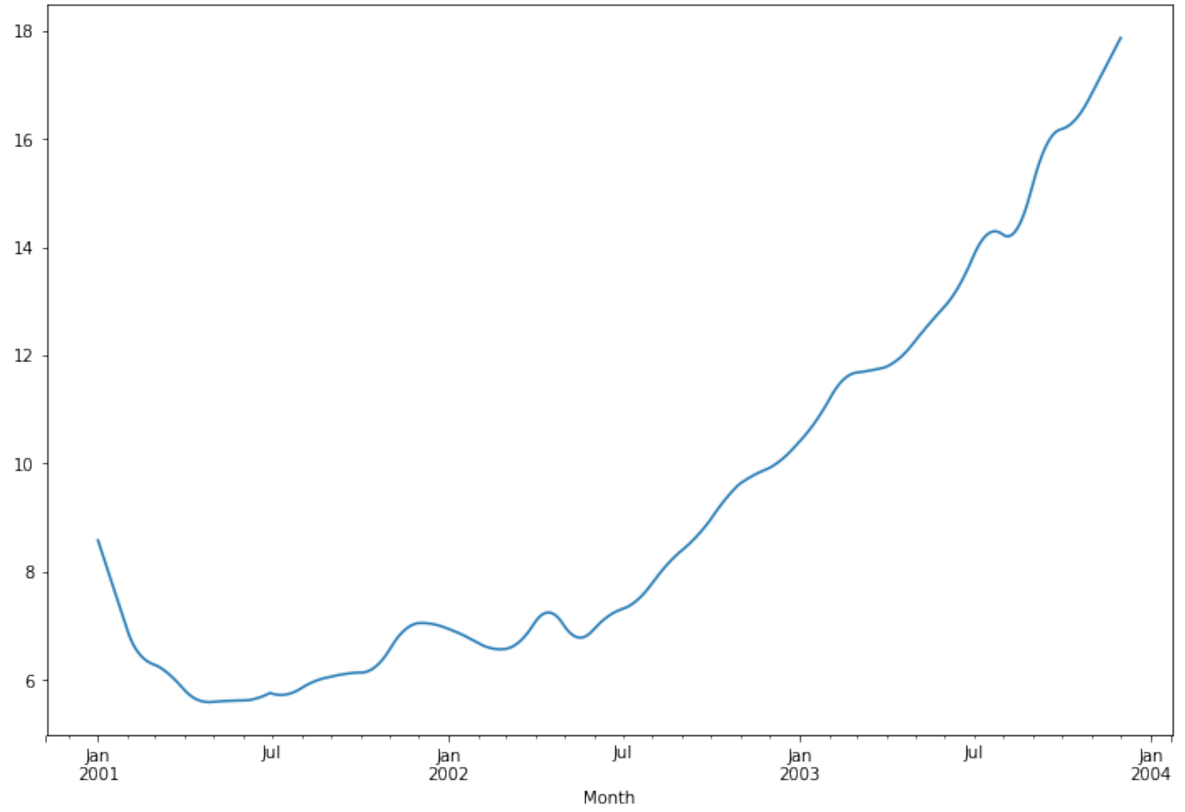
Advanced Data Management (CSCI 490/680)

Provenance & Reproducibility

Dr. David Koop

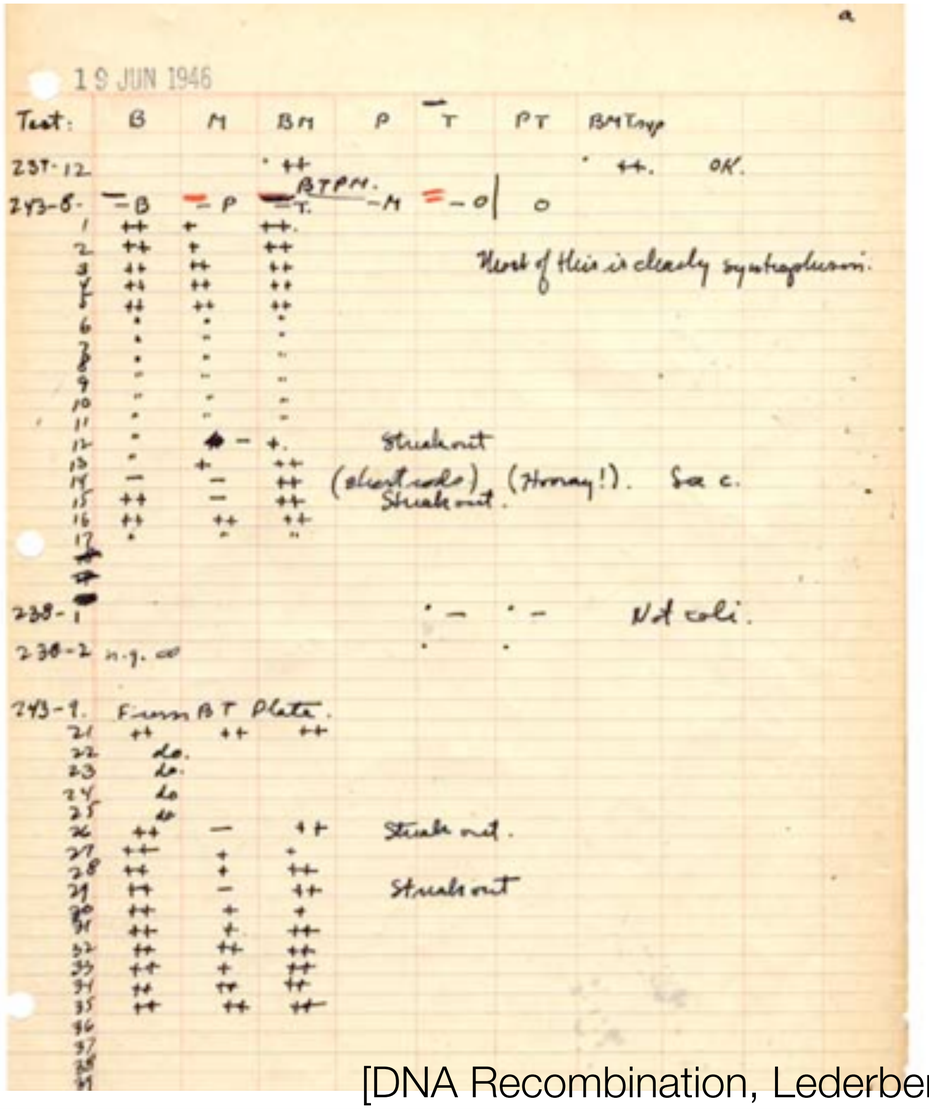
Sales Data and 180-Day Rolling Window





Provenance in Science

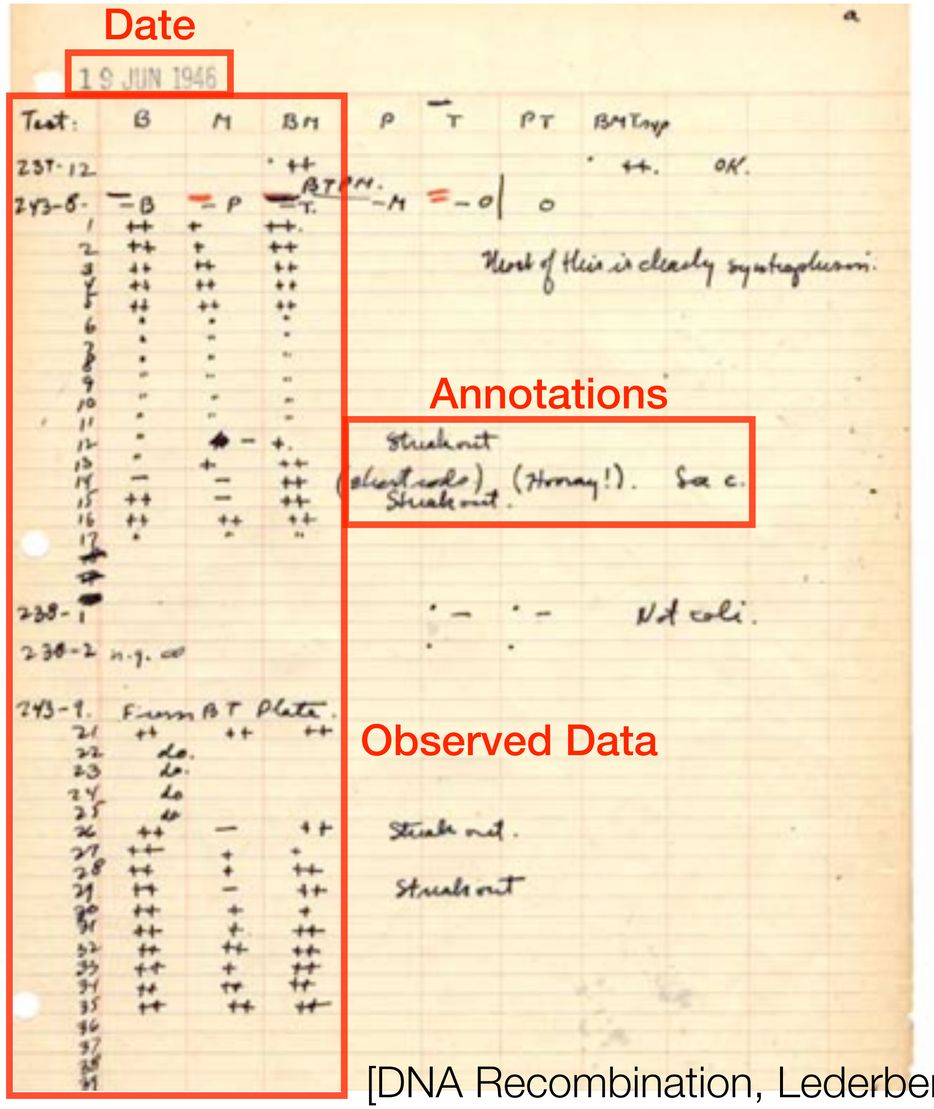
- Provenance: the lineage of data, a computation, or a visualization
- **Provenance is as (or more) important as the result!**
- Old solution:
  - Lab notebooks
- New problems:
  - Large volumes of data
  - Complex analyses
  - Writing notes doesn't scale



[DNA Recombination, Lederberg]

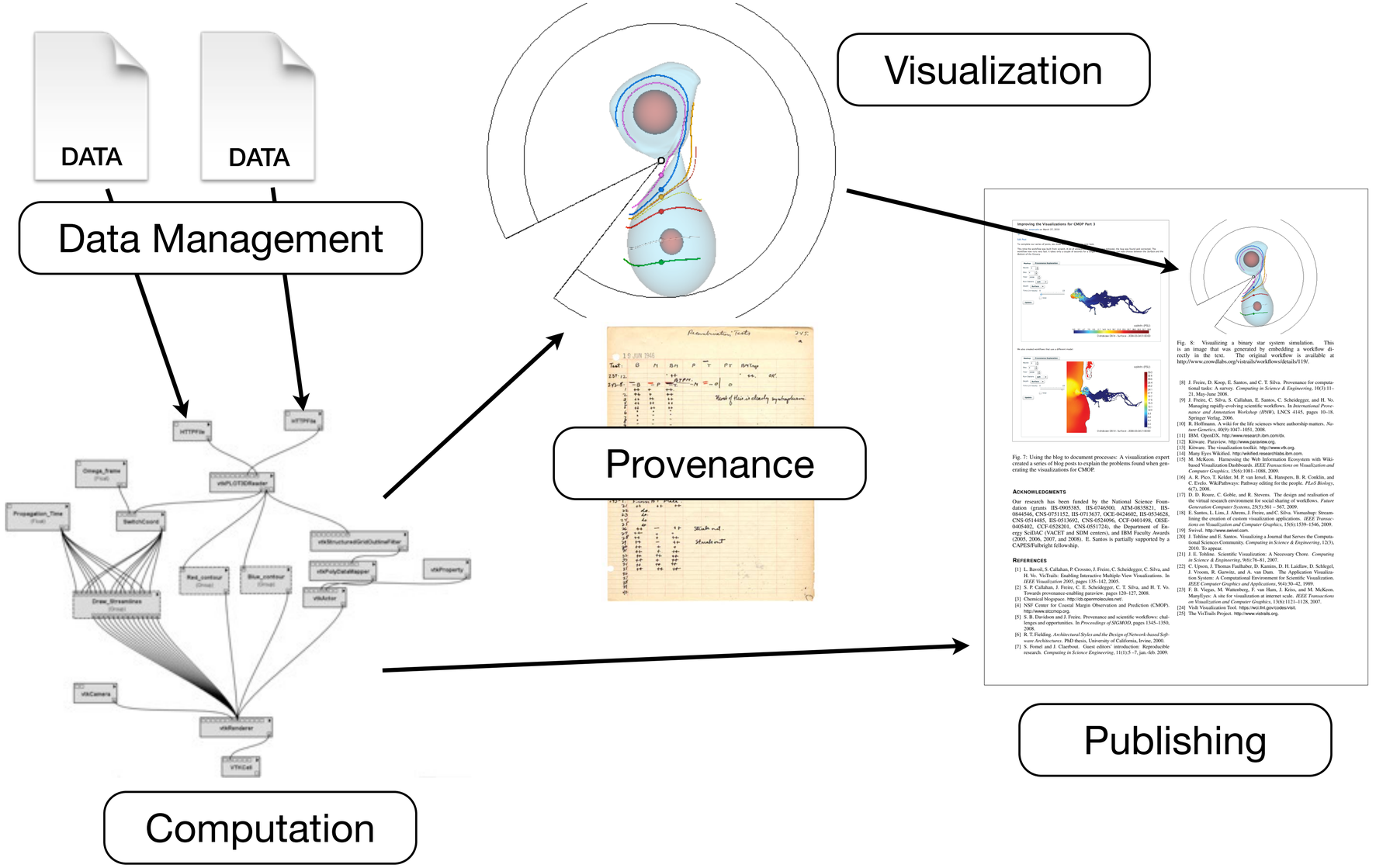
Provenance in Science

- 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

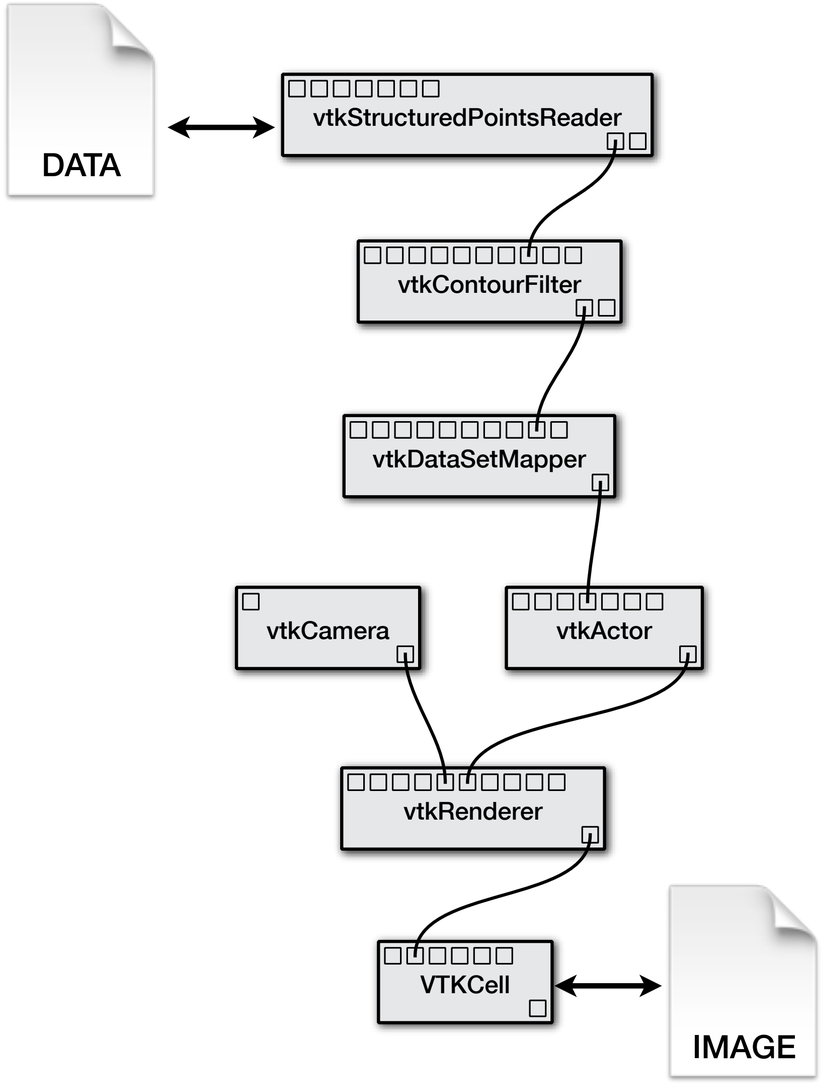


[DNA Recombination, Lederberg]

Provenance in Computational Science



Provenance Questions



- What process led to the output image?
- What input datasets contributed to the output image?
- What workflows create an isosurface with isovalue `57`?
- Who create this data product?
- When was this data file created?
- Why was `vtkCamera` used?
- Why do two output images differ?

Provenance & Causality

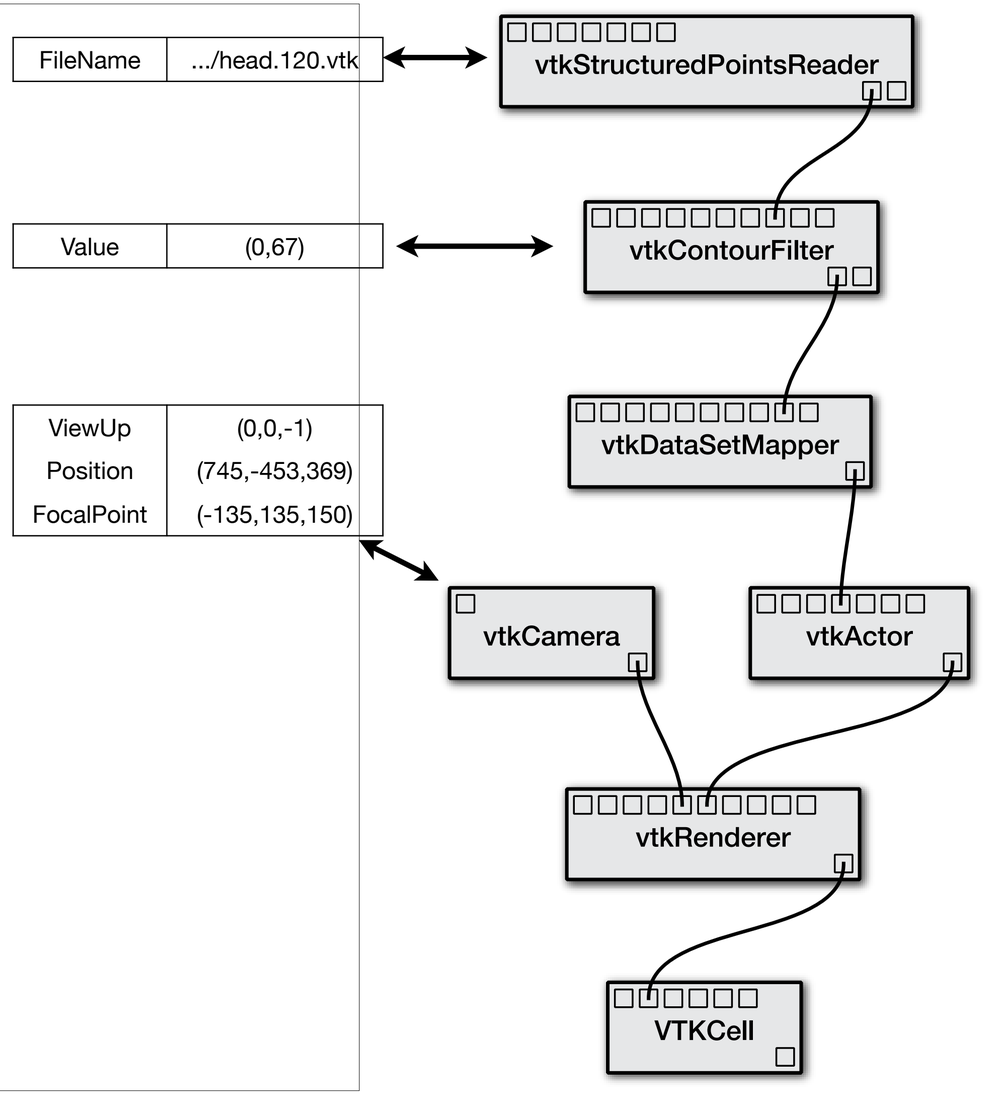
- Knowing what data/steps influenced other data/steps is important!
- Data dependencies: this output file depended on this input file
- Data-process dependencies: this output figure depended on these processes
- Causality can often be represented as a **graph** where connections represent dependencies

Provenance Capture Mechanisms

- **Workflow-based**: Since workflow execution is controlled, keep track of all the workflow modules, parameters, etc. as they are executed
- **Process-based**: Each process is required to write out its own provenance information (not centralized like workflow-based)
- **OS-based**: The OS or filesystem is modified so that any activity it does it monitored and the provenance subsystem organizes it
- Tradeoffs:
  - Workflow- and process-based have better abstraction
  - OS-based requires minimal user effort once installed and can capture "hidden dependencies"

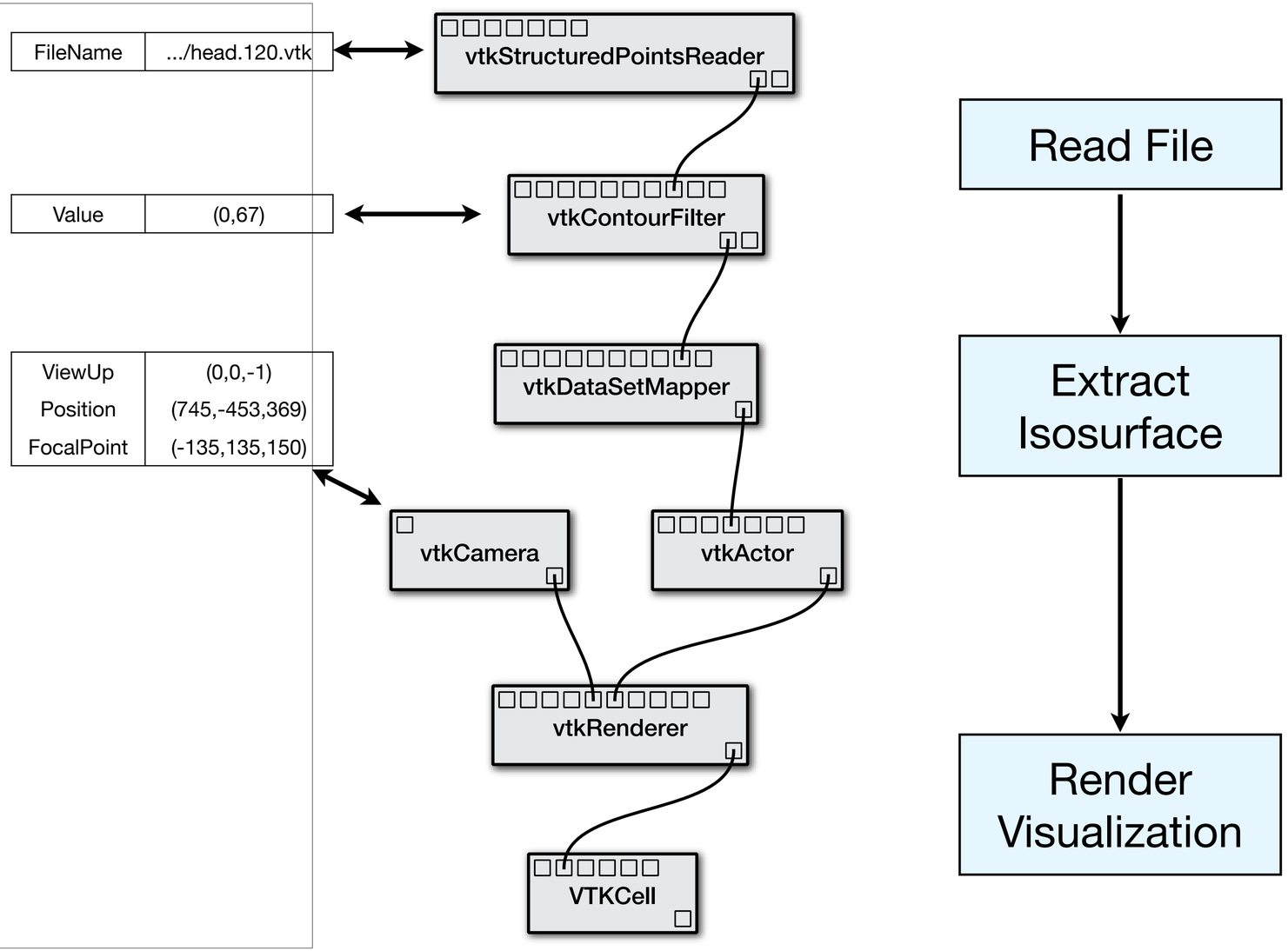
Abstraction: Script, Workflow, Abstract Workflow

```
data = vtk.vtkStructuredPointsReader()
data.SetFileName(../examples/data/head.120.vtk)

contour = vtk.vtkContourFilter()
contour.SetInput(data.GetOutput())
contour.SetValue(0, 67)

mapper = vtk.vtkPolyDataMapper()
mapper.SetInput(contour.GetOutput())
mapper.ScalarVisibilityOff()

actor = vtk.vtkActor()
actor.SetMapper(mapper)

cam = vtk.vtkCamera()
cam.SetViewUp(0,0,-1)
cam.SetPosition(745,-453,369)
cam.SetFocalPoint(135,135,150)
cam.ComputeViewPlaneNormal()

ren = vtk.vtkRenderer()
ren.AddActor(actor)
ren.SetActiveCamera(cam)
ren.ResetCamera()
renwin = vtk.vtkRenderWindow()
renwin.AddRenderer(ren)

style = vtk.vtkInteractorStyleTrackballCamera()
iren = vtk.vtkRenderWindowInteractor()
iren.SetRenderWindow(renwin)
iren.SetInteractorStyle(style)
iren.Initialize()
iren.Start()
```

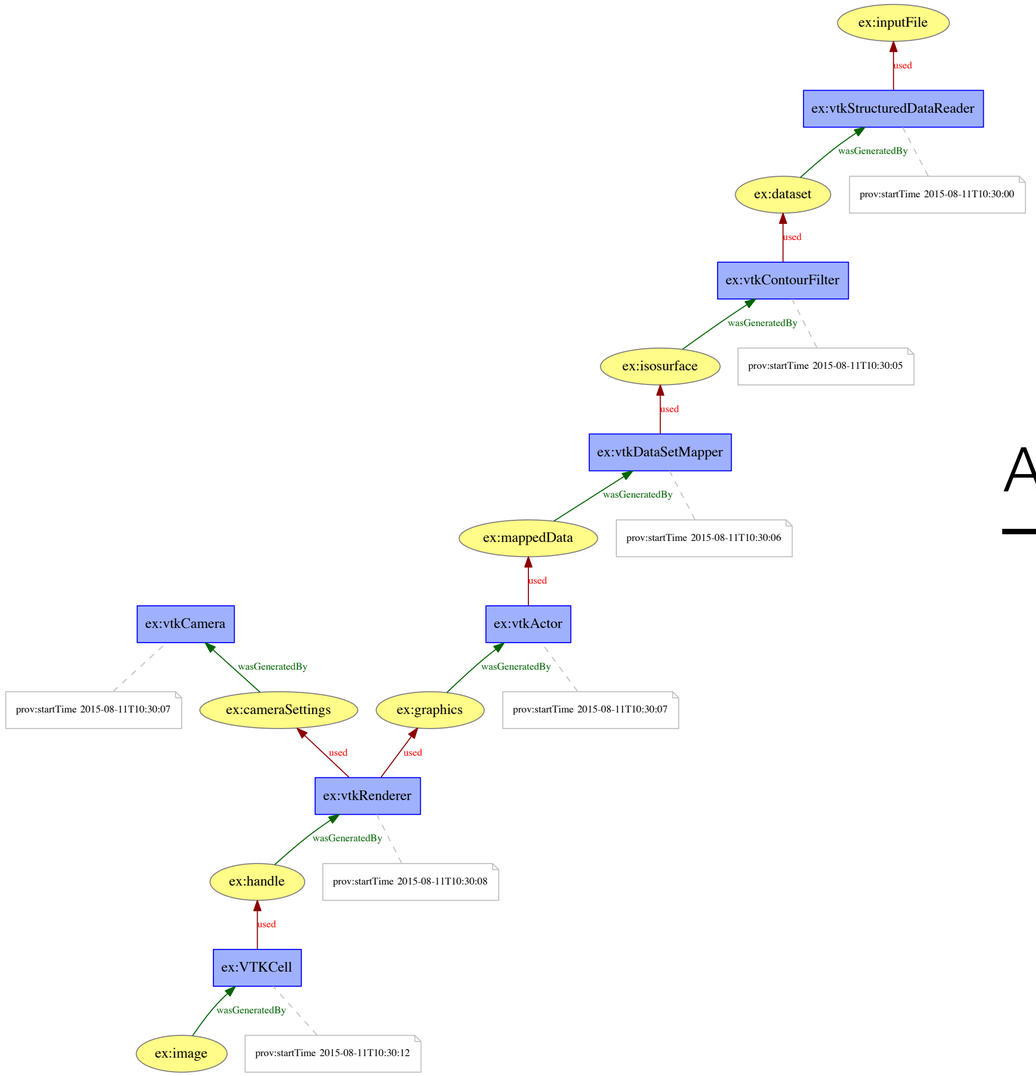


Abstraction: Script, Workflow, Abstract Workflow

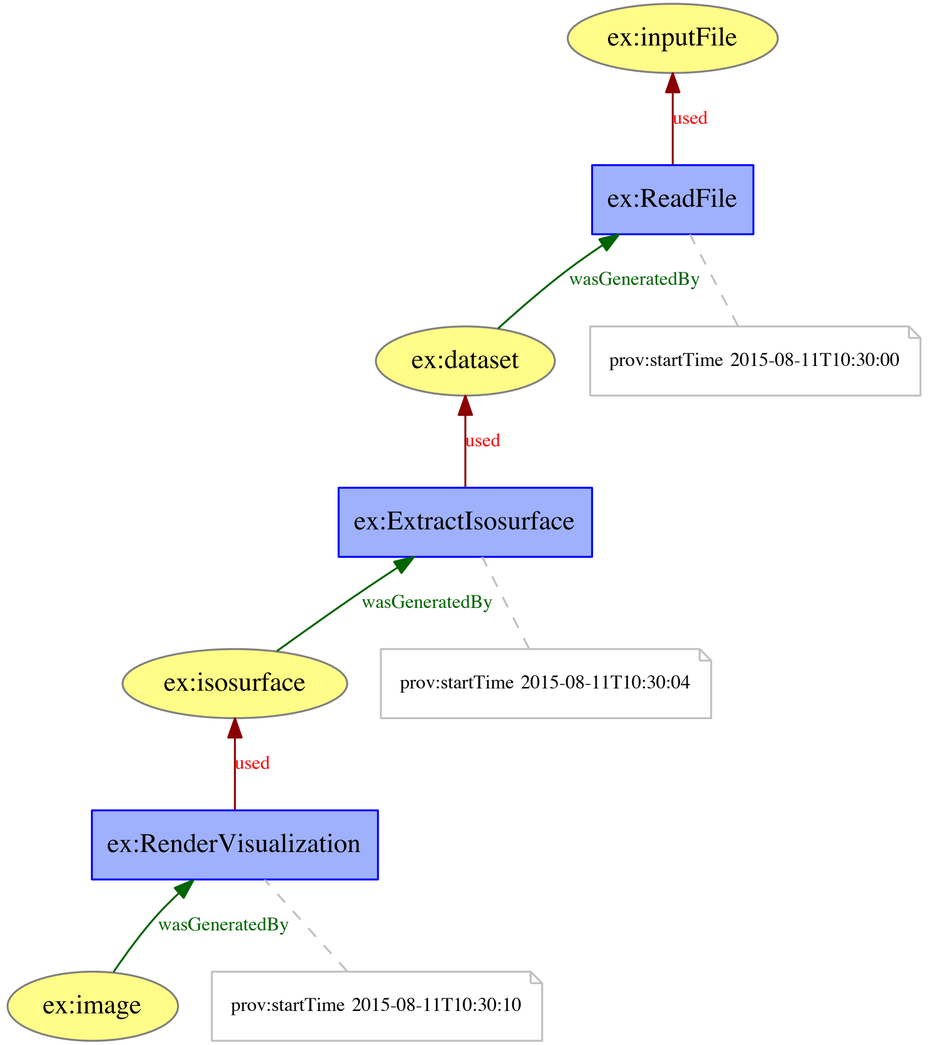
```
data = vtk.vtkStructuredPointsReader()
data.SetFileName(../examples/data/head.120.vtk)

contour = vtk.vtkContourFilter()
contour.SetInput(data.GetOutput())
contour.SetValue(0, 67)

mapper = vtk.vtkPolyDataMapper()
mapper.SetInput(contour.GetOutput())
mapper.ScalarVisibilityOff()

actor = vtk.vtkActor()
actor.SetMapper(mapper)

cam = vtk.vtkCamera()
cam.SetViewUp(0,0,-1)
cam.SetPosition(745,-453,369)
cam.SetFocalPoint(135,135,150)
cam.ComputeViewPlaneNormal()

ren = vtk.vtkRenderer()
ren.AddActor(actor)
ren.SetActiveCamera(cam)
ren.ResetCamera()
renwin = vtk.vtkRenderWindow()
renwin.AddRenderer(ren)

style = vtk.vtkInteractorStyleTrackballCamera()
iren = vtk.vtkRenderWindowInteractor()
iren.SetRenderWindow(renwin)
iren.SetInteractorStyle(style)
iren.Initialize()
iren.Start()
```



Abstraction: Provenance Views



Abstract



Assignment 5

- Chicago Bike Sharing Data
  - Spatial Analysis
  - Temporal Analysis
  - Graph Database (neo4j)

Provenance Storage

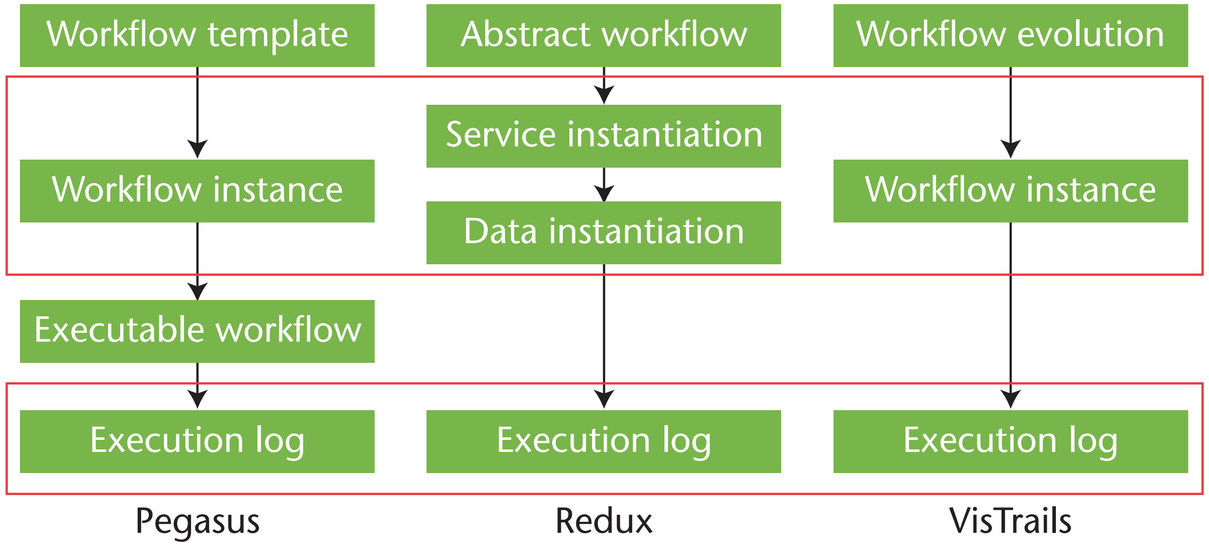
- Keeping provenance for each data item means lots of **repetition**
- Nested data storage also induces repetition
- Coarse provenance is naturally more compact, but how to decide what (not) to store?
- Repeated provenance is not uncommon:
  - Repeating the same computation with a different parameter
  - Creating a new computation that has a very similar structure to one that was run two weeks ago
- Provenance compression/factorization techniques (e.g. [Chapman et al., 2008], [Anand et al., 2009]) take advantage of that to reduce storage costs

Provenance Storage Formats

- Files, relational databases, XML databases, RDF (linked data)
- Log files are good for preserving data but can be bad to query or analyze
- Relational databases are great for column-specific queries but can be bad for dependency queries
- XML databases are more portable than relational databases but are usually less efficient for queries
- RDF triples are better for dependencies and integrating domain-specific knowledge but can be slower

Layered Provenance

- As with relational databases, want to normalize provenance to **minimize redundant information**
- Example: Don't store workflow specification each time that workflow is executed–store it once and reference it
- Also allow different layers for different aspects of provenance



[Freire et. al, 2008]

Provenance Models

- How provenance is represented (more abstract than the details of how it is actually stored)
- PROV (W3C Standard) has different storage backends for provenance but all of it conforms to the same model
- Model the objects involved and their relationships (e.g. activities, dependencies)
- Interoperability is a concern
  - Why? May use multiple tools/techniques to achieve a result, want to analyze the entire provenance chain

Prospective and Retrospective Provenance

- Prospective provenance is what was specified/intended
  - a workflow, script, list of steps
- Retrospective provenance is what actually happened
  - actual data, actual parameters, errors that occurred, timestamps, machine information
- **Do not need** prospective provenance to have retrospective provenance!
- Retrospective provenance is often the same type of information as prospective plus more
- Could have multiple retrospective provenance traces for one prospective provenance listing

Prospective and Retrospective Provenance

- **Example:** Baking a Cake
- Prospective Provenance (Recipe):
  1. Gather ingredients (3/4 cup butter, 3/4 cocoa, 3/4 cup flour, ...)
  2. Preheat oven to 350 degrees
  3. Grease cake pan
  4. Mix wet ingredients in large bowl
  5. Mix dry ingredients in a separate bowl
  6. Add dry mixture to wet mixture
  7. Pour batter into cake pan
  8. Put pan in the oven and bake for 30 minutes
  9. Take cake out of oven and let it cool

Prospective and Retrospective Provenance

- Retrospective Provenance (What actually happened)

1. **Went to store to buy butter**
2. Gathered ingredients (3/4 cup butter, 3/4 cocoa, **1 cup flour**, ...)
3. Greased cake pan
4. Preheated oven to 350 degrees
5. Mixed wet ingredients in large bowl
6. Mixed dry ingredients in a separate bowl
7. Added **wet** mixture to **dry** mixture
8. Poured batter into cake pan
9. Put pan in the oven and baked for **35 minutes**
10. Took cake out of oven and let it cool for **10 minutes**

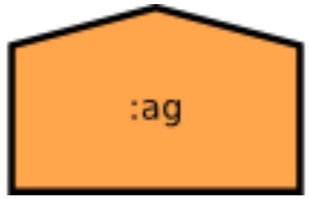
Provenance Model History

- Community organized provenance challenges (2006-2009)
- First Provenance Challenge assessed capabilities of systems
- Second Provenance Challenge examined interoperability
- Led to development of Open Provenance Model (OPM), (2007)
  - Sought to establish interchange format for provenance
- Further work led to PROV W3C Recommendations (2013)
  - Some confusion from name changes from OPM to PROV even though concepts are similar
  - Focus is on **model** not formats

PROV: Three Key Classes

An **entity** is a physical, digital, conceptual, or other kind of thing with some fixed aspects; entities may be real or imaginary.

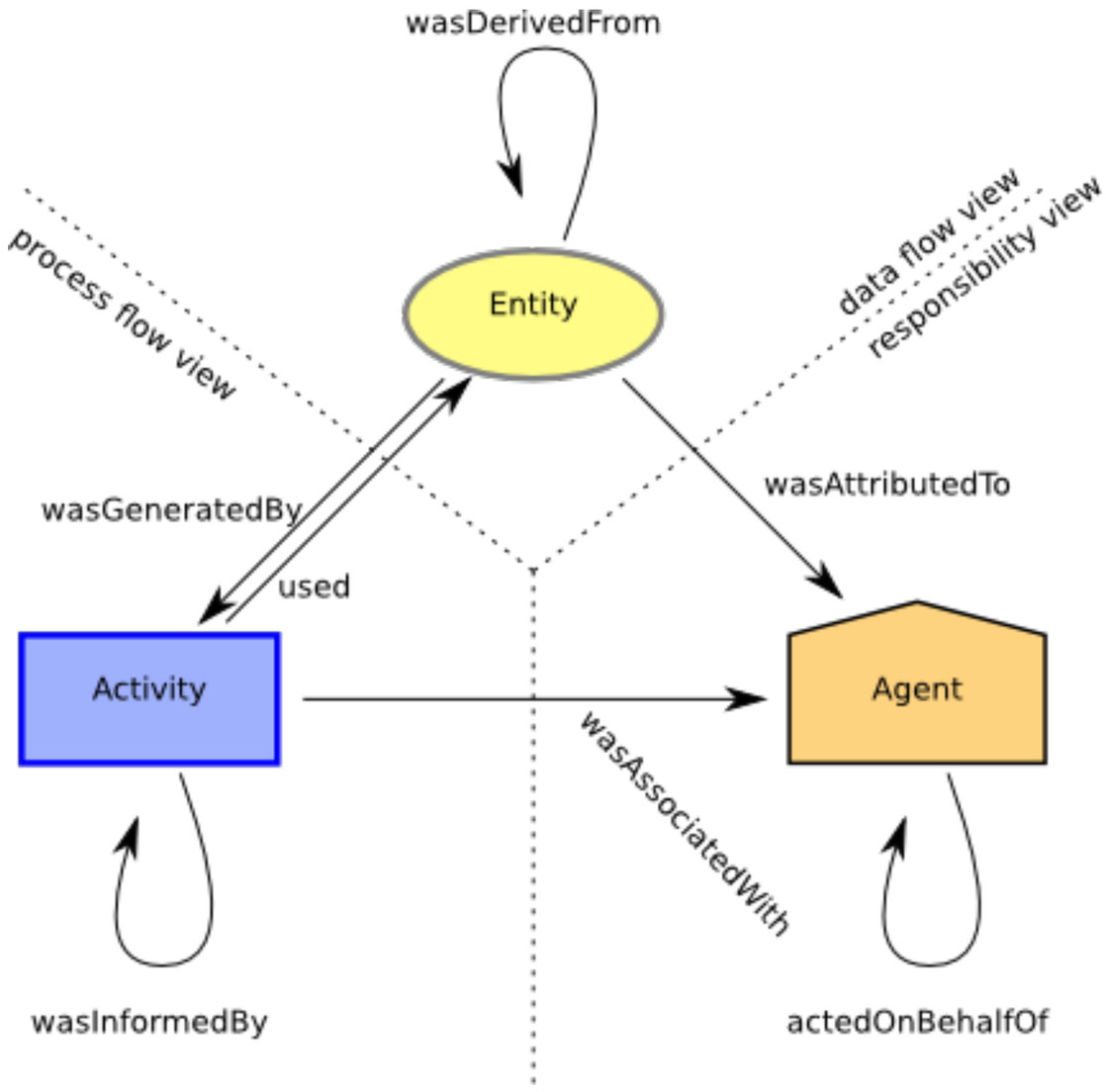
An **activity** is something that occurs over a period of time and acts upon or with entities; it may include consuming, processing, transforming, modifying, relocating, using, or generating entities.



An **agent** is something that bears some form of responsibility for an activity taking place, for the existence of an entity, or for another agent's activity.

[Moreau et al., 2014]

PROV: Three Views of Provenance



[Moreau et al., 2014]

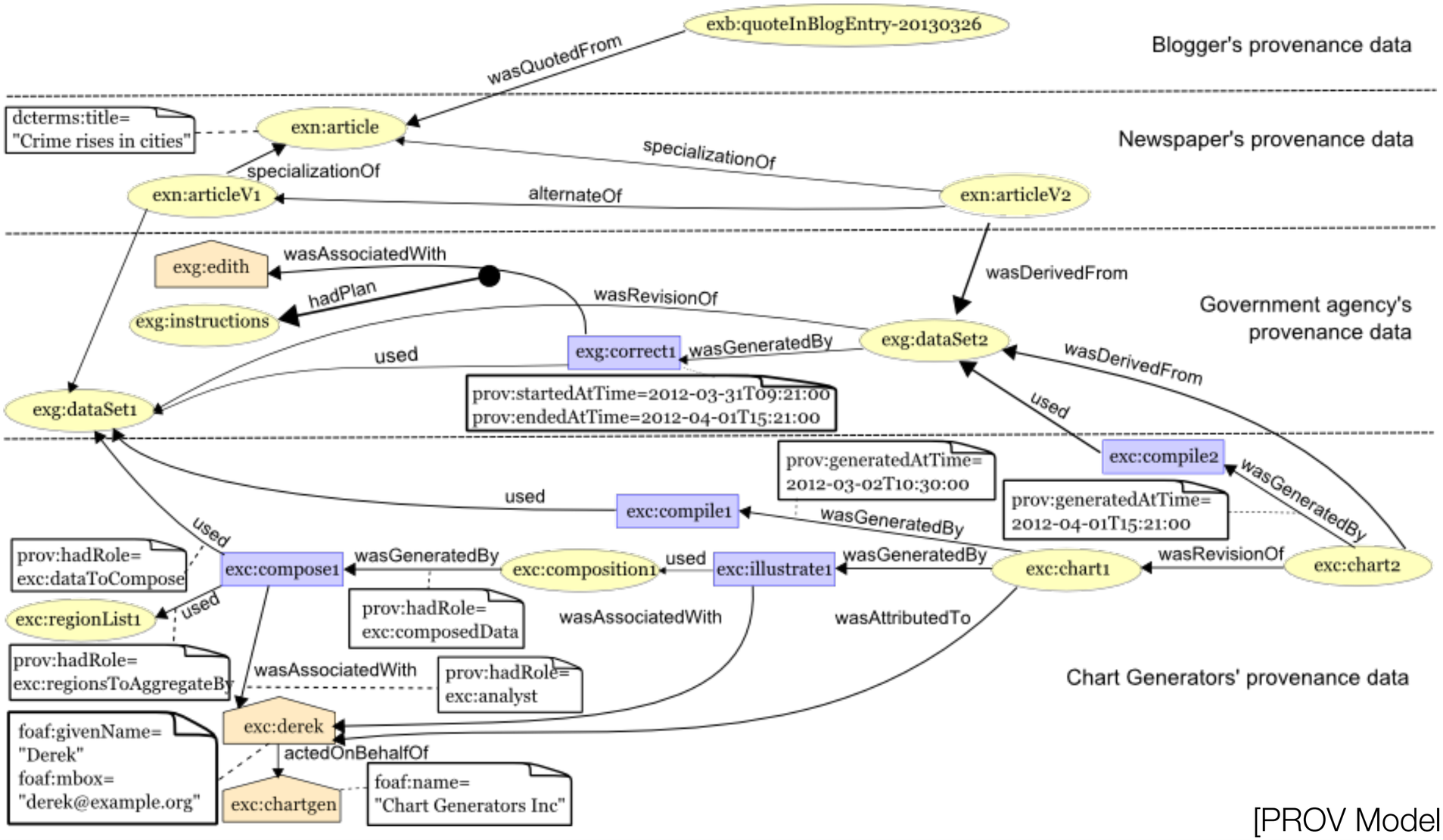
PROV Edges: Derivation

- Derivation Edges:
  - wasGeneratedBy: entity ⟶ activity
  - used: activity ⟶ entity

  - wasDerivedFrom: entity ⟶ entity

[PROV Model Primer, 2013]

PROV Example



[PROV Model Primer, 2013]

Querying Provenance

- Query methods are often tied to storage backend
- SQL, XQuery, Prolog, SPARQL, ...

REDUX

```
SELECT Execution.ExecutableWorkflowId, Execution.ExecutionId, Event.EventId, ExecutableActivity.ExecutableActivityId
from Execution, Execution_Event, Event, ExecutableWorkflow_ExecutableActivity, ExecutableActivity,
    ExecutableActivity_Property_Value, Value, EventType as ET
where Execution.ExecutionId=Execution_Event.ExecutionId
and Execution_Event.EventId=Event.EventId
and ExecutableActivity.ExecutableActivityId=ExecutableActivity_Property_Value.ExecutableActivityId
and ExecutableActivity_Property_Value.ValueId=Value.ValueId and Value.Value=Cast('-m 12' as binary)
and ((CONVERT(DECIMAL, Event.Timestamp)+0)%7)=0 and Execution_Event.ExecutableWorkflow_ExecutableActivityId=
   ExecutableWorkflow_ExecutableActivity.ExecutableWorkflow_ExecutableActivityId
and ExecutableWorkflow_ExecutableActivity.ExecutableWorkflowId=Execution.ExecutableWorkflowId
and ExecutableWorkflow_ExecutableActivity.ExecutableActivityId=ExecutableActivity.ExecutableActivityId
and Event.EventTypeId=ET.EventTypeId and ET.EventTypeName='Activity Start';
```

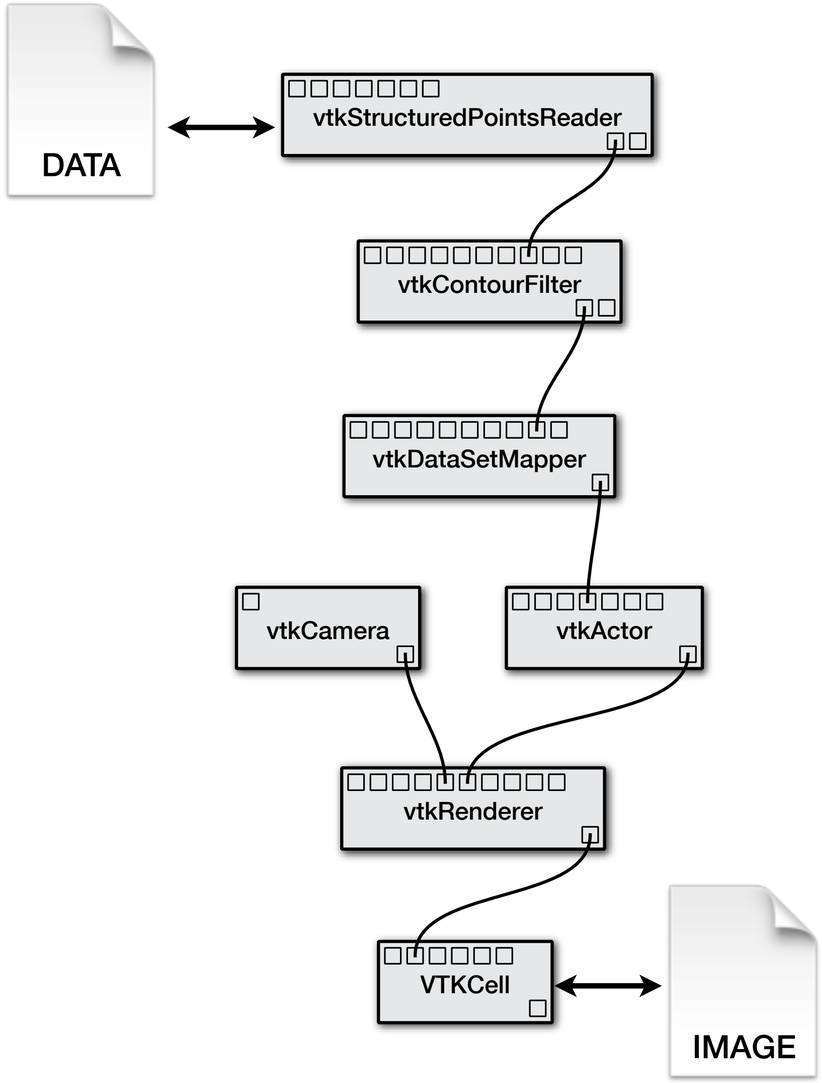
VisTrails

```
wf{*}: x where x.module='AlignWarp' and x.parameter('model')='12'
        and (log{x}: y where y.dayOfWeek='Monday')
```

MyGrid

```
SELECT ?p
where (?p <http://www.mygrid.org.uk/provenance#startTime> ?time) and (?time > date)
using ns for <http://www.mygrid.org.uk/provenance#> xsd for <http://www.w3.org/2001/XMLSchema#>

SELECT ?p
where <urn:lsid:www.mygrid.org.uk:experimentinstance:HXQOVQA2ZI0>
(?p <http://www.mygrid.org.uk/provenance#runsProcess> ?processname .
?p <http://www.mygrid.org.uk/provenance#processInput> ?inputParameter .
?inputParameter <ont:model> <ontology:twelfthOrder>)
using ns for <http://www.mygrid.org.uk/provenance#> ont for <http://www.mygrid.org.uk/ontology#>
```

Querying Provenance

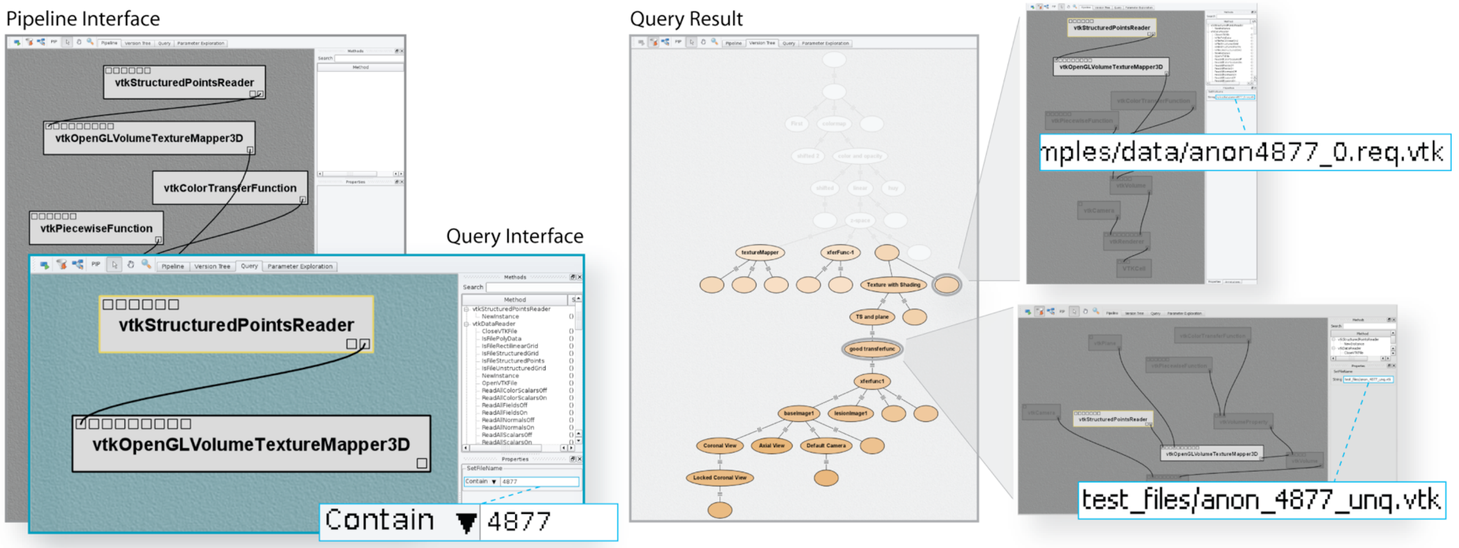


- *What process led to the output image?*
- *What input datasets contributed to the output image?*
- *What workflows include resampling and isosurfacing with isovalue* `57`*?*

- Graph traversal or graph patterns
  - How do we write such queries?

Querying Provenance by Example

- Provenance is represented as graphs: hard to specify queries using text!
- Querying workflows by example [Scheidegger et al., TVCG 2007; Beeri et al., VLDB 2006; Beeri et al. VLDB 2007]
  - WYSIWYQ -- What You See Is What You Query
  - Interface to create workflow is same as to query



Stronger Links Between Provenance and Data

```
<workflow_exec id="1">
  <m_exec id="5"
          name="vtkStructuredDataReader"
          package="edu.utah.sci.vistrails
          version="5.6.0">
    <param id="2" name="SetFile"
           value="/MyData/05-12-sc2.dat"/>
  </m_exec>
  <m_exec id="6"
          name="vtkContourFilter"
          package="edu.utah.sci.vistrails.vtk"
          version="5.6.0">
    <param id="3" name="SetValue"
           value="[1, 57]"/>
    <param id="4" name="ComputeScalarsOn"
           value="True"/>
  </m_exec>

  ...

  <m_exec id="11"
          name="FileSink"
          package="edu.utah.sci.vistrails.basic"
          version="1.5">
    <param id="15" name="path"
           value="/home/a/results/23.out"/>
  </m_exec>
```

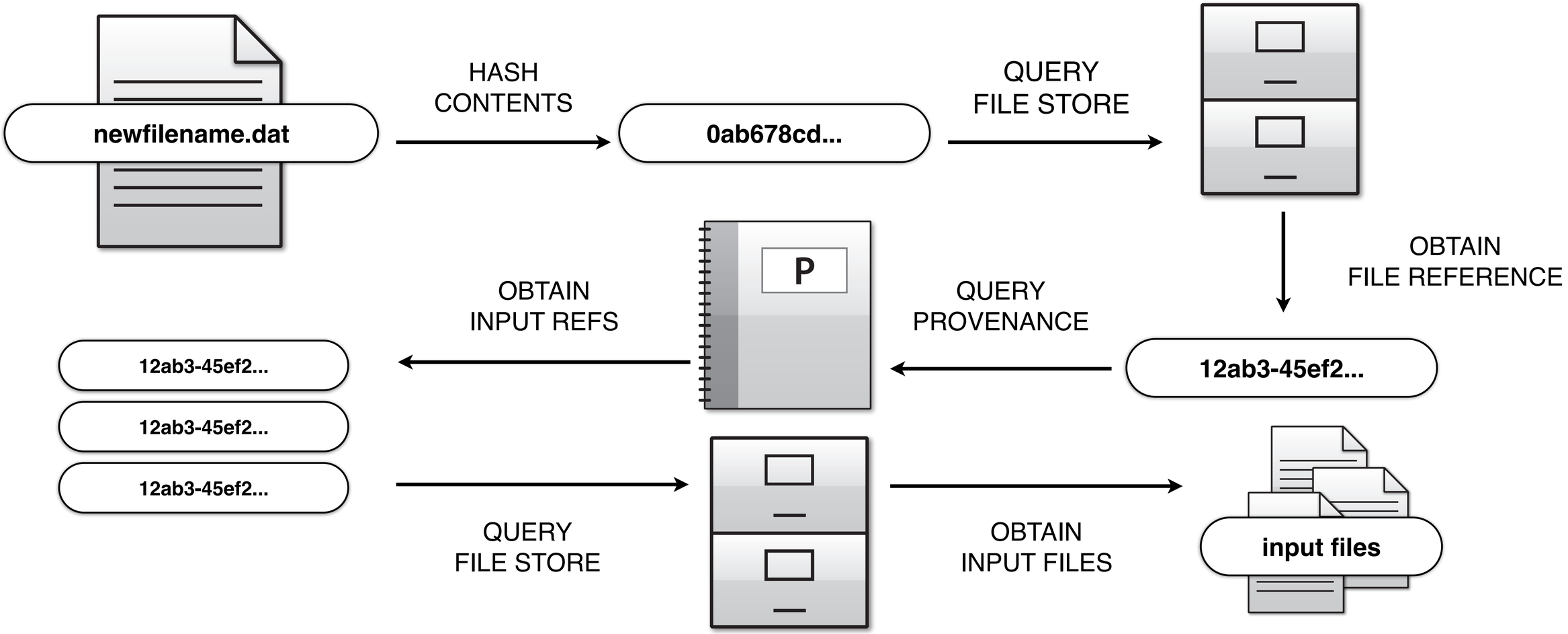
FILE NOT FOUND

FILE NOT FOUND

- Filenames are often the mode of identification in data exploration
- We might also use URIs or access curated data stores
  - Always expected for exploratory tasks?
  - What happens if offline?
- Solution:
  - Managed store for data associated with computations
  - Improved data identification
  - Automatic versioning

[Koop et. al, 2010]

Provenance from Data



[Koop et. al, 2010]

Provenance-Enabled Systems

Table 1. Provenance-enabled systems.

System	Capture mechanism	Prospective provenance	Retrospective provenance	Workflow evolution
REDUX	Workflow-based	Relational	Relational	No
Swift	Workflow-based	SwiftScript	Relational	No
VisTrails	Workflow-based	XML and relational	Relational	Yes
Karma	Workflow- and process-based	Business Process Execution Language	XML	No
Kepler	Workflow-based	MoML	MoML variation	Under development
Taverna	Workflow-based	Scufl	RDF	Under development
Pegasus	Workflow-based	OWL	Relational	No
PASS	OS-based	N/A	Relational	No
ES3	OS-based	N/A	XML	No
PASOA/PreServ	Process-based	N/A	XML	No

[Freire et. al, 2008]

Provenance-Enabled Systems

Table 1. Provenanc			
System	**Storage**	**Query support**	**Available as open source?**
REDUX	Relational database management system (RDBMS)	SQL	No
Swift	RDBMS	SQL	Yes
VisTrails	RDBMS and files	Visual query by example, specialized language	Yes
Karma	RDBMS	Proprietary API	Yes
Kepler	Files; RDBMS planned	Under development	Yes
Taverna	RDBMS	SPARQL	Yes
Pegasus	RDBMS	SPARQL for metadata and workflow; SQL for execution log	Yes
PASS	Berkeley DB	nq (proprietary query tool)	No
ES3	XML database	XQuery	No
PASOA/PreServ	Filesystem, Berkeley DB	XQuery, Java query API	Yes

[Freire et. al, 2008]

Provenance-Enabled Systems

Table 1. Provenanc

System	Storage	Query support	Available as open source?
REDUX	Relational database management system (RDBMS)	SQL	No
Swift	RDBMS	SQ	Yes
VisTrails	RDBMS and files	Visual query by example, specialized la	Yes
Karma	RDBMS	Proprietary API	Yes
Kepler	Files; RDBMS planned	Under development	Yes
Taverna	RDBMS	SP	Yes
Pegasus	RDBMS	SPARQL for metadata and workflow; SQL for execution log	Yes
PASS	Berkeley DB	nq (proprietary query tool)	No
ES3	XML database	XQuery	No
PASOA/PreServ	Filesystem, Berkeley DB	XQuery, Java query API	Yes

[Freire et. al, 2008]

Today: Two types of provenance

- Database Provenance
- Evolution Provenance

Database Provenance

- Motivation: Data warehouses and curated databases
  - Lots of work
  - Provenance helps check correctness
  - Adds value to data by how it was obtained
- Three Types:
  - Why (Lineage): Associate each tuple t present in the output of a query with a set of tuples present in the input
  - How: Not just existence but routes from tuples to output (multiple contrib.'s)
  - Where: Location where data is copied from (may have choice of different tables)

[Cheney et al., 2007]

Provenance in Databases

A. Amarilli

Why Provenance

Agencies

	name	based_in	phone
t_1:	BayTours	San Francisco	415-1200
t_2:	HarborCruz	Santa Cruz	831-3000

ExternalTours

	name	destination	type	price
t_3:	BayTours	San Francisco	cable car	$50
t_4:	BayTours	Santa Cruz	bus	$100
t_5:	BayTours	Santa Cruz	boat	$250
t_6:	BayTours	Monterey	boat	$400
t_7:	HarborCruz	Monterey	boat	$200
t_8:	HarborCruz	Carmel	train	$90

```
Q1:
SELECT a.name, a.phone
FROM Agencies a, ExternalTours e
WHERE a.name = e.name AND e.type='boat'
```

Result of Q_1:

name	phone
BayTours	415-1200
HarborCruz	831-3000

- Lineage of `(HarborCruz, 831-3000)`: `{Agencies(t2),ExternalTours(t7)}`
- Lineage of `(BayTours, 415-1200)`: `{Agencies(t1), ExternalTours(t5,t6)}`
- This is not really precise because we don't need both `t5` and `t6`—only one is ok

[Cheney et al., 2007]

How Provenance

Agencies

	name	based_in	phone
t_1:	BayTours	San Francisco	415-1200
t_2:	HarborCruz	Santa Cruz	831-3000

ExternalTours

	name	destination	type	price
t_3:	BayTours	San Francisco	cable car	$50
t_4:	BayTours	Santa Cruz	bus	$100
t_5:	BayTours	Santa Cruz	boat	$250
t_6:	BayTours	Monterey	boat	$400
t_7:	HarborCruz	Monterey	boat	$200
t_8:	HarborCruz	Carmel	train	$90

Q_2:

```
SELECT   e.destination, a.phone
FROM     Agencies a,
         (SELECT name,
              based_in AS destination
         FROM Agencies a
         UNION
         SELECT name, destination
         FROM ExternalTours ) e
WHERE    a.name = e.name
```

Result of Q_2:

destination	phone	
San Francisco	415-1200	$t_1 \cdot (t_1 + t_3)$
Santa Cruz	831-3000	t_2^2
Santa Cruz	415-1200	$t_1 \cdot (t_4 + t_5)$
Monterey	415-1200	$t_1 \cdot t_6$
Monterey	831-3000	$t_1 \cdot t_7$
Carmel	831-3000	$t_1 \cdot t_8$

- How provenance gives more detail about how the tuples provide witnesses to the result
- Prov of `(San Francisco, 415-1200)`: `{{t1}, {t1,t3}}`
- `t1` contributes **twice**
- Uses provenance semirings (the "polynomial" shown on the right)

[Cheney et al., 2007]

Where Provenance

Agencies

	name	based_in	phone
t_1:	BayTours	San Francisco	415-1200
t_2:	HarborCruz	Santa Cruz	831-3000

ExternalTours

	name	destination	type	price
t_3:	BayTours	San Francisco	cable car	$50
t_4:	BayTours	Santa Cruz	bus	$100
t_5:	BayTours	Santa Cruz	boat	$250
t_6:	BayTours	Monterey	boat	$400
t_7:	HarborCruz	Monterey	boat	$200
t_8:	HarborCruz	Carmel	train	$90

Q_1:
SELECT a.name, a.phone
FROM Agencies a, ExternalTours e
WHERE a.name = e.name
AND e.type='boat'

Q'_1:
SELECT e.name, a.phone
FROM Agencies a, ExternalTours e
WHERE a.name = e.name
AND e.type='boat'

Result of Q_1:

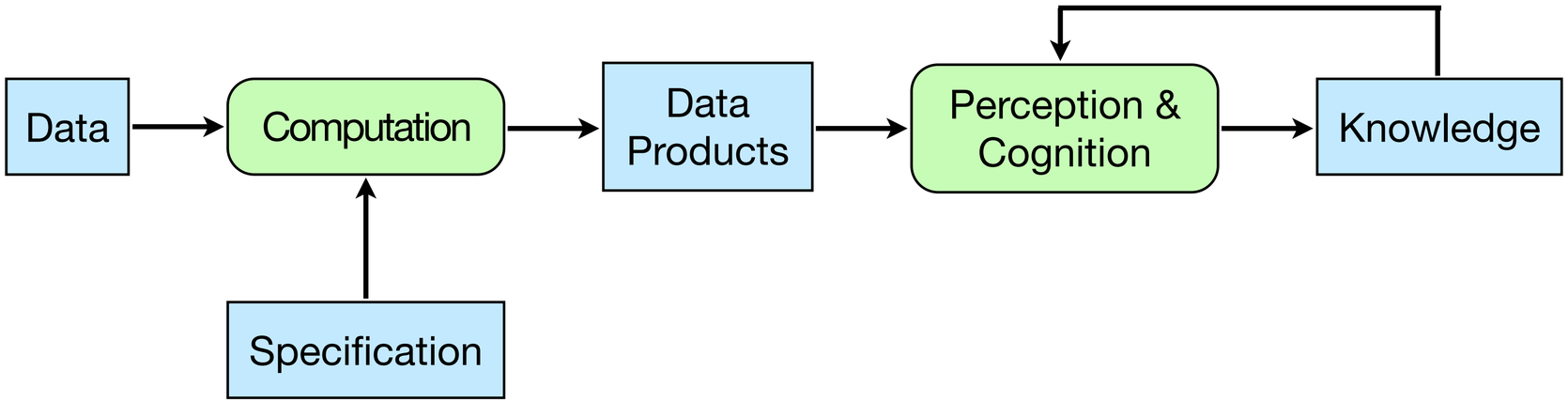
name	phone
BayTours	415-1200
HarborCruz	831-3000

- Where provenance traces to specific locations, not the tuple values
- Q and Q' give the same result but the name comes from different places
- Prov of `HarborCruz` in second output: `(t2, name)`
- Important in annotation-propogation

[Cheney et al., 2007]

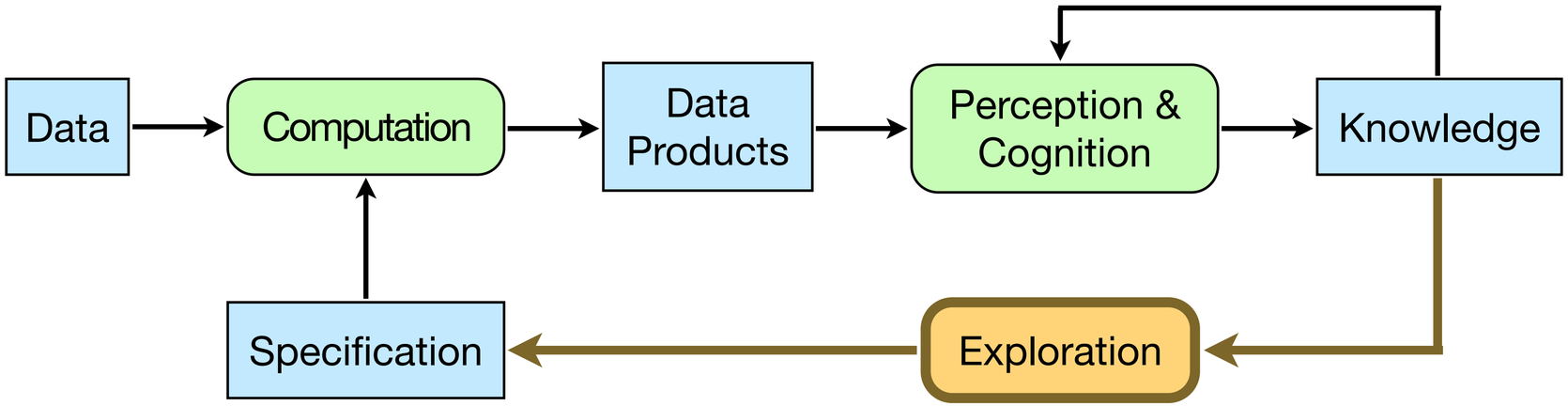
Evolution Provenance

Data Exploration



[Modified from Van Wijk, Vis 2005]

Data Exploration



[Modified from Van Wijk, Vis 2005]

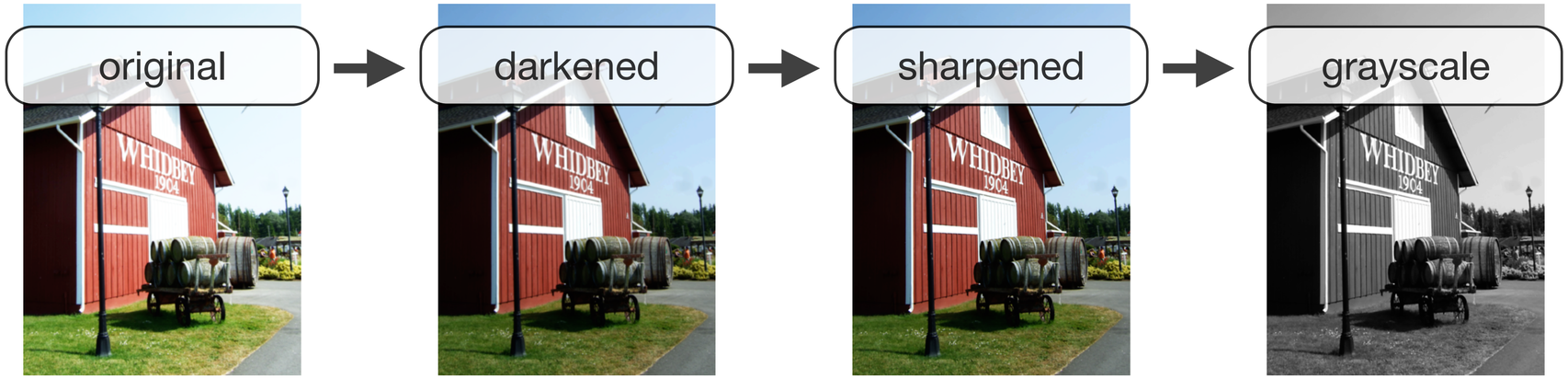
- Data analysis and visualization are iterative processes
- In exploratory tasks, change is the norm!

Exploration and Creativity Support

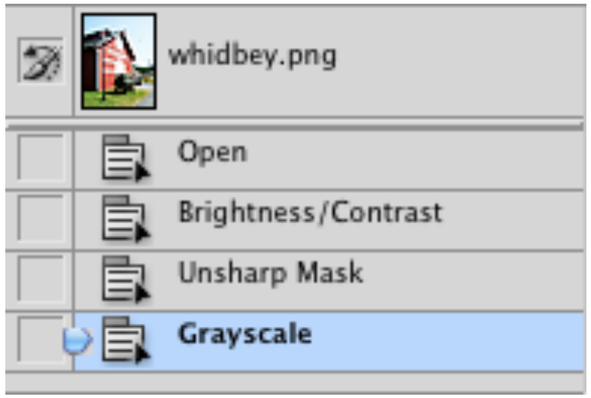
- Reasoning is key to the exploratory processes
- “Reflective reasoning requires the ability to store temporary results, to make inferences from stored knowledge, and to follow chains of reasoning backward and forward, sometimes backtracking when a promising line of thought proves to be unfruitful. …the process is slow and laborious” —Donald A. Norman
- Need external aids—tools to facilitate this process
  - "Creativity support tools" —Ben Shneiderman
- Need aid from people—collaboration

Change-based Provenance: Photo Editing

- User Actions

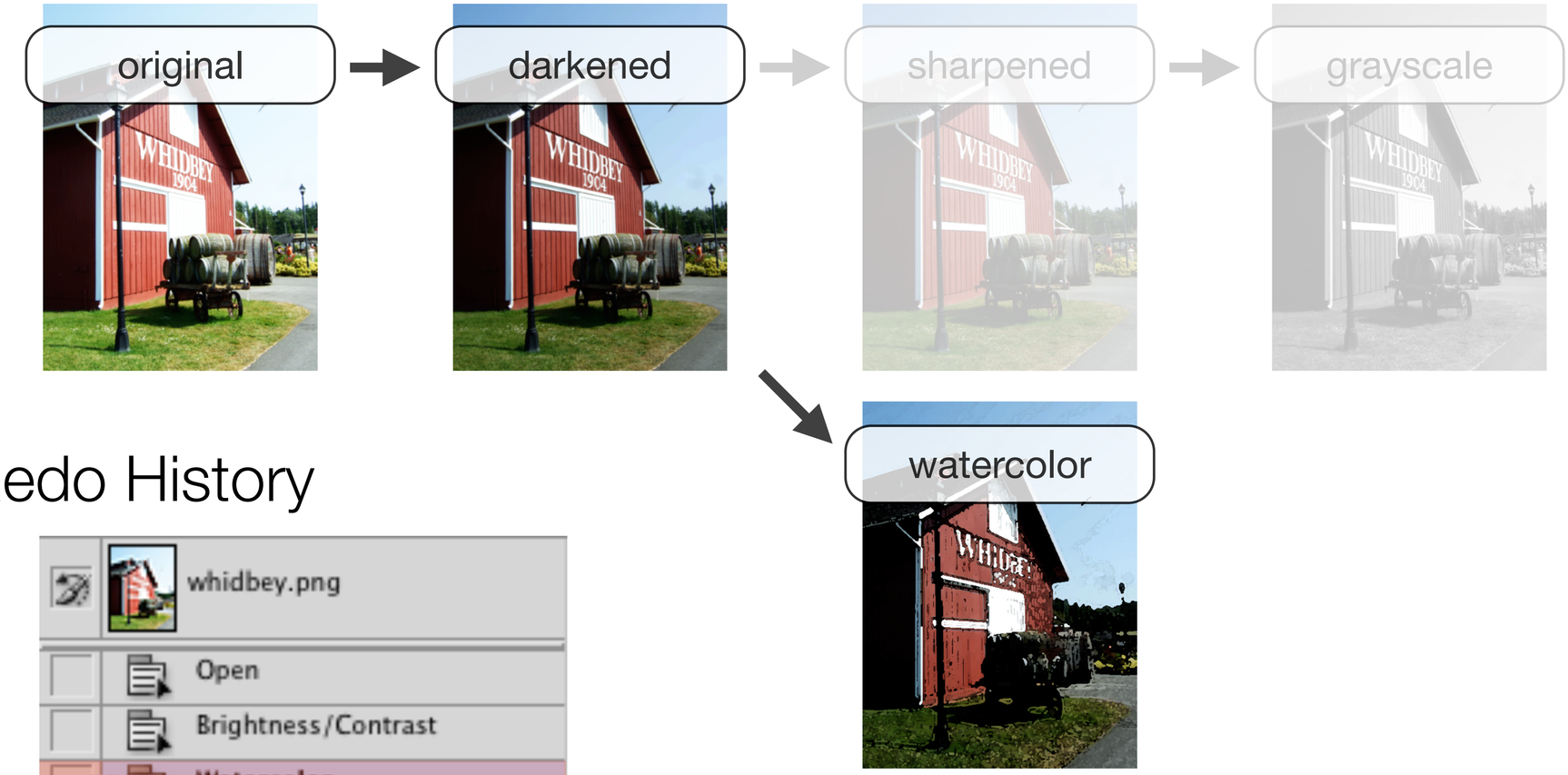


- Undo/Redo History



Change-based Provenance: Photo Editing

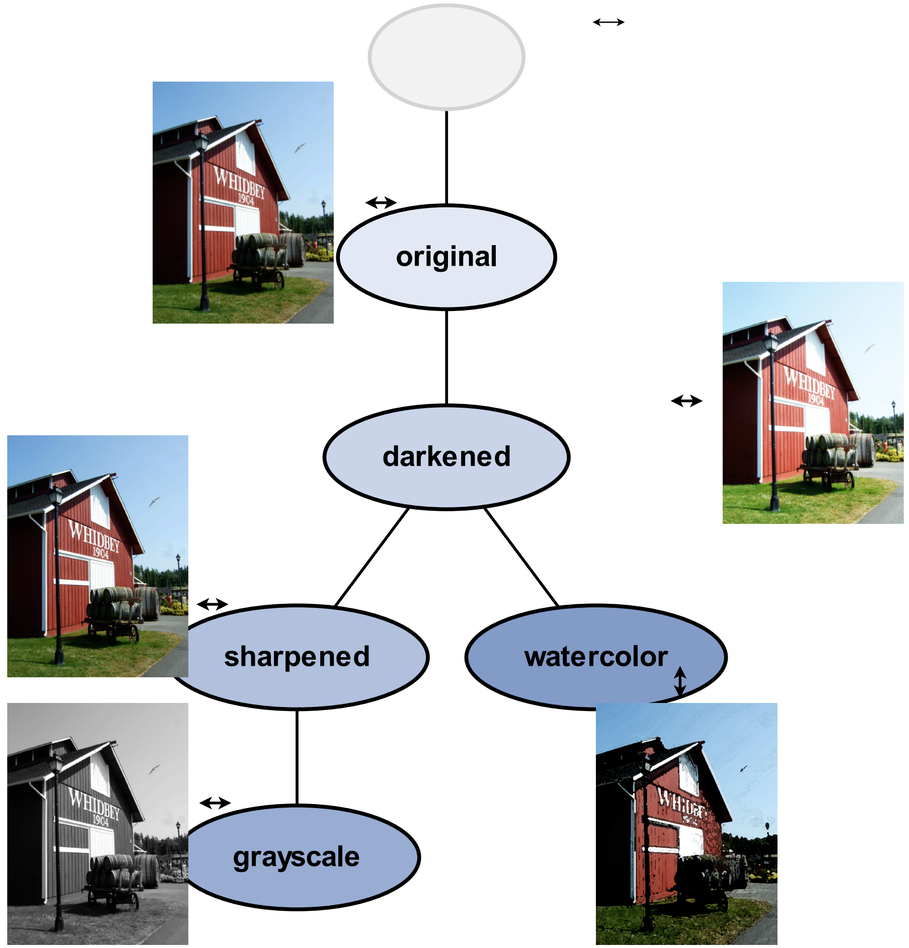
- User Actions



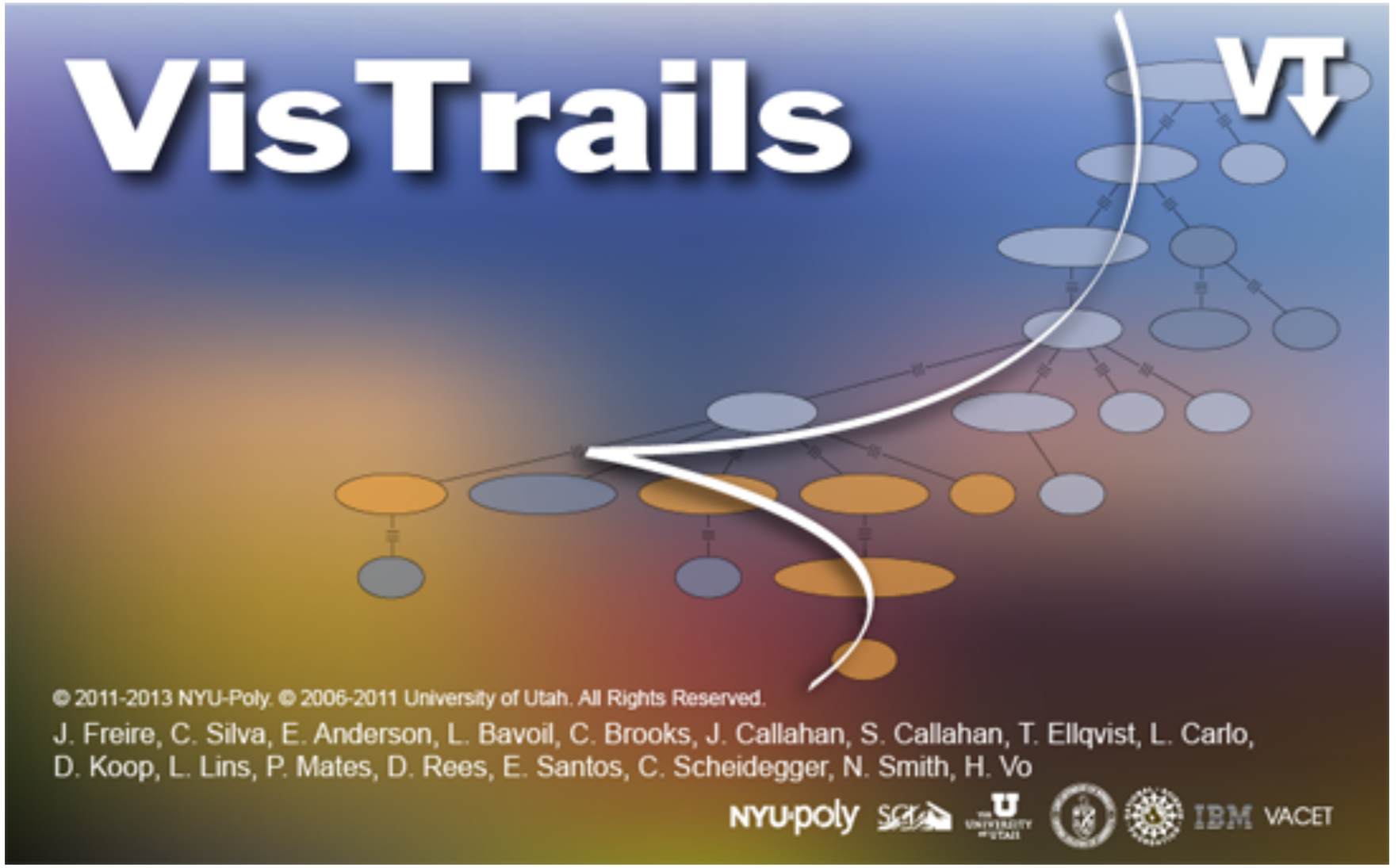
- Undo/Redo History

Version Trees

- 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



VisTrails


VisTrails
VT
© 2011-2013 NYU-Poly. © 2006-2011 University of Utah. All Rights Reserved.
J. Freire, C. Silva, E. Anderson, L. Bavoil, C. Brooks, J. Callahan, S. Callahan, T. Ellqvist, L. Carlo,
D. Koop, L. Lins, P. Mates, D. Rees, E. Santos, C. Scheidegger, N. Smith, H. Vo
NYU-poly
IBM
VACET

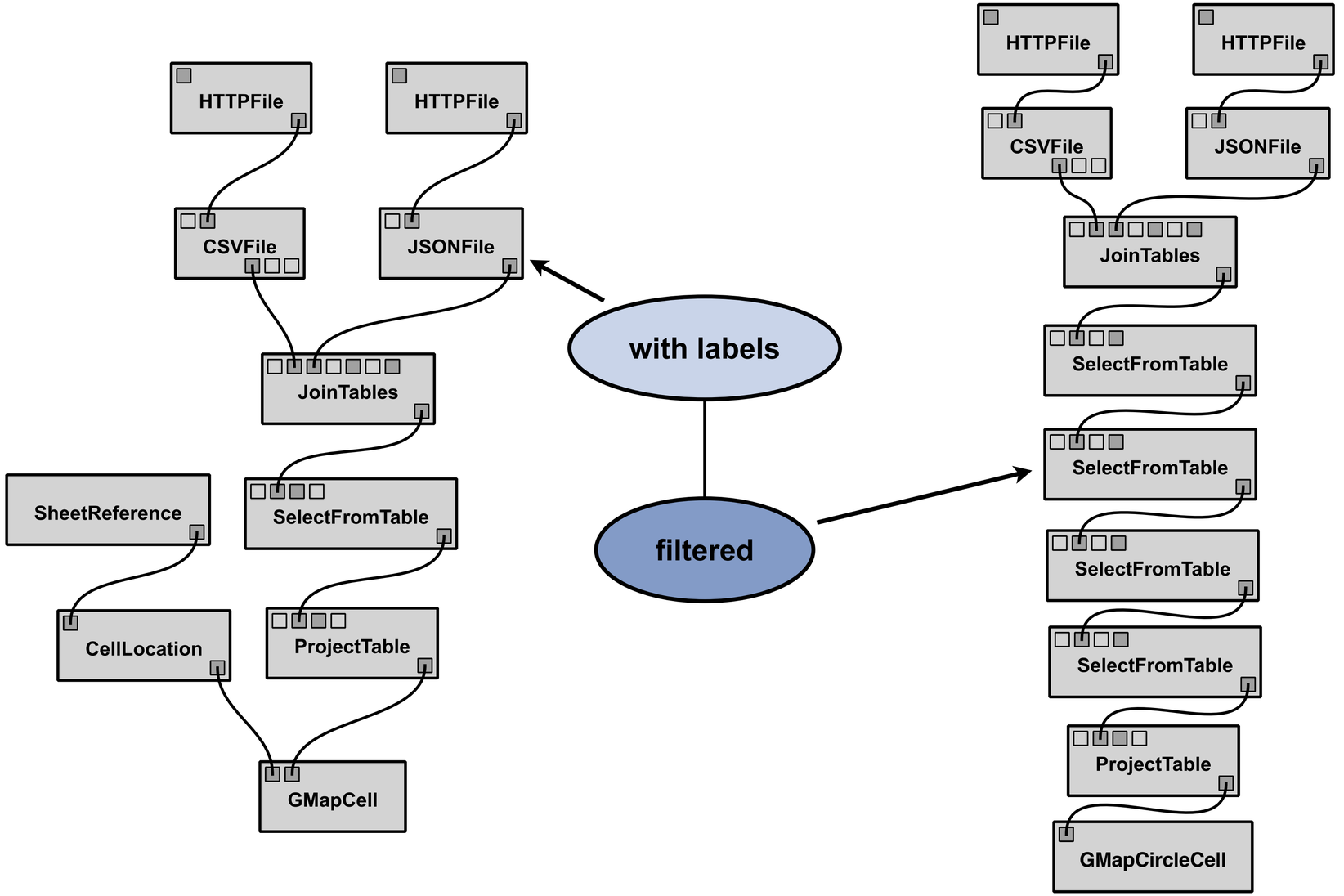
VisTrails

- Comprehensive provenance infrastructure for computational tasks
- Focus on exploratory tasks such as simulation, visualization, and data analysis
- 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
- Focus on usability—build tools for scientists

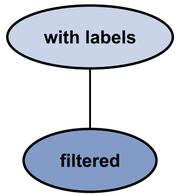
Discussion

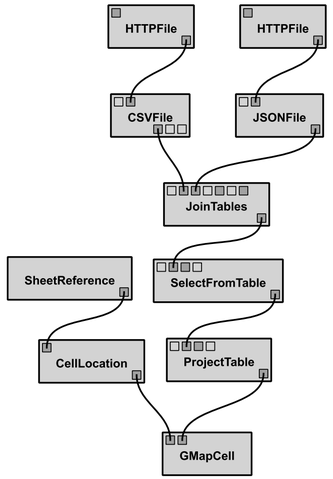
- Reproducibility in VisTrails

Workflow Evolution Provenance

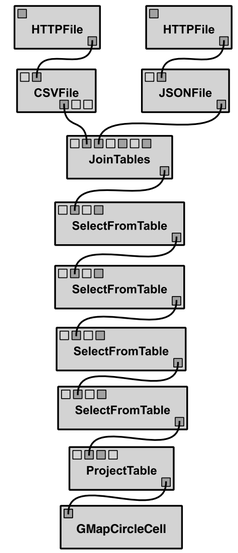


Workflow Evolution Provenance

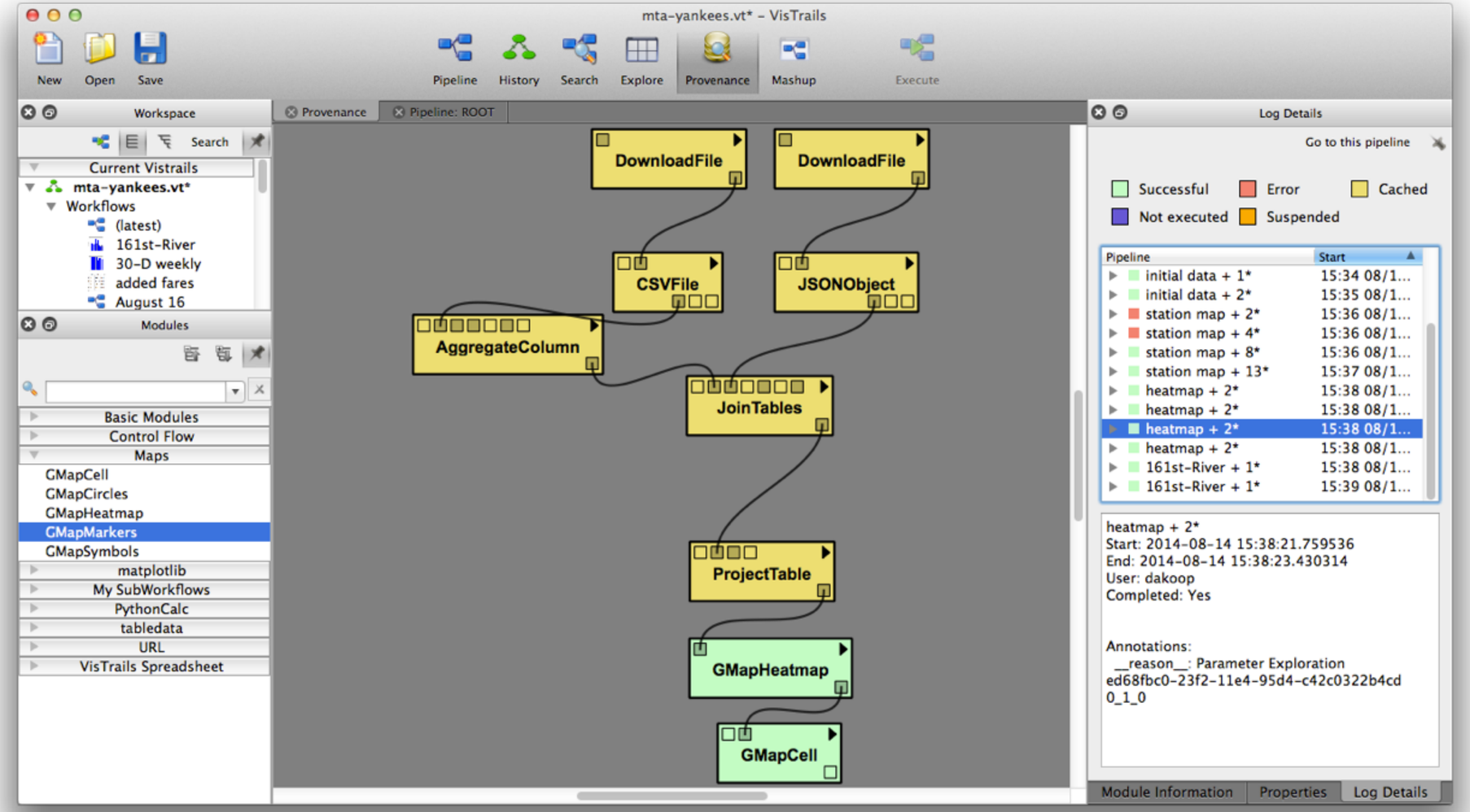




delete module “GMapCell”
delete module “CellLocation”
delete module “ProjectTable”
delete module “SelectFromTable”
...
add module “SelectFromTable”
add parameter “float_expr” to “SelectFromTable” with value “latitutde > 40.6”
delete parameter “float_expr” from “SelectFromTable”
add parameter “float_expr” to “SelectFromTable” with value “latitutde > 40.7”
delete parameter “float_expr” from “SelectFromTable”
add parameter “float_expr” to “SelectFromTable” with value “latitutde > 40.8”
...



Execution Provenance



Execution Provenance

```
<module id="12" name="vtkDataSetReader"
        start_time="2010-02-19 11:01:05"
        end_time="2010-02-19 11:01:07">
  <annotation key="hash"
            value="c54bea63cb7d912a43ce"/>
</module>
<module id="13" name="vtkContourFilter"
        start_time="2010-02-19 11:01:07"
        end_time="2010-02-19 11:01:08"/>
<module id="15" name="vtkDataSetMapper"
        start_time="2010-02-19 11:01:09"
        end_time="2010-02-19 11:01:12"/>
<module id="16" name="vtkActor"
        start_time="2010-02-19 11:01:12"
        end_time="2010-02-19 11:01:13"/>
<module id="17" name="vtkCamera"
        start_time="2010-02-19 11:01:13"
        end_time="2010-02-19 11:01:14"/>
<module id="18" name="vtkRenderer"
        start_time="2010-02-19 11:01:14"
        end_time="2010-02-19 11:01:14"/>
...
```