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
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
Provenance in Computational Science

Data Management

Visualization

Provenance

Publishing
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 \texttt{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 is 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

```python
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

```python
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)

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

ren = vtk.vtkRenderer()
ren.AddActor(actor)
ren.SetActiveCamera(kам)
ren.ResetCamera()

renwin = vtk.vtkRenderWindow()
renwin.AddRenderer(ren)

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

Diagram:
- Read File
- Extract Isosurface
- Render Visualization
Abstraction: Provenance Views
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

![Provenance Diagram]

[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
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]
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, 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_Name=Event.Property_Name and 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;
```
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

• 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

newfilename.dat ➔ HASH CONTENTS ➔ 0ab678cd... ➔ QUERY FILE STORE ➔ OBTAIN FILE REFERENCE

12ab3-45ef2... ➔ OBTAIN INPUT REFS ➔ QUERY PROVENANCE ➔ 12ab3-45ef2... ➔ OBTAIN INPUT FILES ➔ input files

[Koop et. al, 2010]
Provenance-Enabled Systems

Table 1. Provenance-enabled systems.

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

[Freire et al., 2008]
## Provenance-Enabled Systems

### Table 1. Provenance

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

[Freire et. al, 2008]
Provenance-Enabled Systems

Table 1. Provenance

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

[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

<table>
<thead>
<tr>
<th>name</th>
<th>based_in</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayTours</td>
<td>San Francisco</td>
<td>415-1200</td>
</tr>
<tr>
<td>HarborCruz</td>
<td>Santa Cruz</td>
<td>831-3000</td>
</tr>
</tbody>
</table>

#### ExternalTours

<table>
<thead>
<tr>
<th>name</th>
<th>destination</th>
<th>type</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayTours</td>
<td>San Francisco</td>
<td>cable car</td>
<td>$50</td>
</tr>
<tr>
<td>BayTours</td>
<td>Santa Cruz</td>
<td>bus</td>
<td>$100</td>
</tr>
<tr>
<td>BayTours</td>
<td>Santa Cruz</td>
<td>boat</td>
<td>$250</td>
</tr>
<tr>
<td>BayTours</td>
<td>Monterey</td>
<td>boat</td>
<td>$400</td>
</tr>
<tr>
<td>HarborCruz</td>
<td>Monterey</td>
<td>boat</td>
<td>$200</td>
</tr>
<tr>
<td>HarborCruz</td>
<td>Carmel</td>
<td>train</td>
<td>$90</td>
</tr>
</tbody>
</table>

Q1:

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

**Result of Q1:**

<table>
<thead>
<tr>
<th>name</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayTours</td>
<td>415-1200</td>
</tr>
<tr>
<td>HarborCruz</td>
<td>831-3000</td>
</tr>
</tbody>
</table>

#### Lineage

- **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

1.1 Why, How and Where: An Overview

**Agencies**

<table>
<thead>
<tr>
<th>name</th>
<th>based_in</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1:</td>
<td>BayTours</td>
<td>San Francisco</td>
</tr>
<tr>
<td>t2:</td>
<td>HarborCruz</td>
<td>Santa Cruz</td>
</tr>
</tbody>
</table>

**ExternalTours**

<table>
<thead>
<tr>
<th>name</th>
<th>destination</th>
<th>type</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>t3:</td>
<td>BayTours</td>
<td>San Francisco</td>
<td>cable car</td>
</tr>
<tr>
<td>t4:</td>
<td>BayTours</td>
<td>Santa Cruz</td>
<td>bus</td>
</tr>
<tr>
<td>t5:</td>
<td>BayTours</td>
<td>Santa Cruz</td>
<td>boat</td>
</tr>
<tr>
<td>t6:</td>
<td>BayTours</td>
<td>Monterey</td>
<td>boat</td>
</tr>
<tr>
<td>t7:</td>
<td>HarborCruz</td>
<td>Monterey</td>
<td>boat</td>
</tr>
<tr>
<td>t8:</td>
<td>HarborCruz</td>
<td>Carmel</td>
<td>train</td>
</tr>
</tbody>
</table>

**Q2:**

```
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 Q2:**

<table>
<thead>
<tr>
<th>destination</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>San Francisco</td>
<td>415-1200</td>
</tr>
<tr>
<td>Santa Cruz</td>
<td>831-3000</td>
</tr>
<tr>
<td>San Francisco</td>
<td>415-1200</td>
</tr>
<tr>
<td>Santa Cruz</td>
<td>415-1200</td>
</tr>
<tr>
<td>Monterey</td>
<td>415-1200</td>
</tr>
<tr>
<td>Monterey</td>
<td>831-3000</td>
</tr>
<tr>
<td>Carmel</td>
<td>831-3000</td>
</tr>
</tbody>
</table>

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

[Cheney et al., 2007]
Where Provenance

<table>
<thead>
<tr>
<th>Agencies</th>
<th>name</th>
<th>based_in</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1:</td>
<td>BayTours</td>
<td>San Francisco</td>
<td>415-1200</td>
</tr>
<tr>
<td>t2:</td>
<td>HarborCruz</td>
<td>Santa Cruz</td>
<td>831-3000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ExternalTours</th>
<th>name</th>
<th>destination</th>
<th>type</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>t3:</td>
<td>BayTours</td>
<td>San Francisco</td>
<td>cable car</td>
<td>$50</td>
</tr>
<tr>
<td>t4:</td>
<td>BayTours</td>
<td>Santa Cruz</td>
<td>bus</td>
<td>$100</td>
</tr>
<tr>
<td>t5:</td>
<td>BayTours</td>
<td>Santa Cruz</td>
<td>boat</td>
<td>$250</td>
</tr>
<tr>
<td>t6:</td>
<td>BayTours</td>
<td>Monterey</td>
<td>boat</td>
<td>$400</td>
</tr>
<tr>
<td>t7:</td>
<td>HarborCruz</td>
<td>Monterey</td>
<td>boat</td>
<td>$200</td>
</tr>
<tr>
<td>t8:</td>
<td>HarborCruz</td>
<td>Carmel</td>
<td>train</td>
<td>$90</td>
</tr>
</tbody>
</table>

\[ Q_1: \]
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 \( Q_1: \):
<table>
<thead>
<tr>
<th>name</th>
<th>phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayTours</td>
<td>415-1200</td>
</tr>
<tr>
<td>HarborCruz</td>
<td>831-3000</td>
</tr>
</tbody>
</table>

- 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 \( \text{HarborCruz} \) in second output: \((t2, \text{name})\)
- Important in annotation-propogation

[Cheney et al., 2007]
Evolution Provenance
Data Exploration

[Diagram showing the process of data exploration from data to knowledge, with steps including computation, data products, perception & cognition, and specification.]

[Modified from Van Wijk, Vis 2005]
Data Exploration

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

[Modified from Van Wijk, Vis 2005]
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

original → darkened → sharpened → grayscale

• 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

• 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
delete module “GMapCell”
dele module “CellLocation”
delete module “ProjectTable”
delete module “SelectFromTable”

... 

add module “SelectFromTable”
add parameter “float_expr” to “SelectFromTable” with value “latitude > 40.6”
delete parameter “float_expr” from “SelectFromTable”
add parameter “float_expr” to “SelectFromTable” with value “latitude > 40.7”
delete parameter “float_expr” from “SelectFromTable”
add parameter “float_expr” to “SelectFromTable” with value “latitude > 40.8”

...
Execution Provenance