Advanced Data Management (CSCI 640/490)

Provenance

Dr. David Koop
Sharing Data

• Required/encouraged by universities, funding agencies, publishers

• "Publications are arguments made by authors, and **data are the evidence** used to support the arguments." [C. L. Borgman]

• Questions:
  - How is data maintained? Who is responsible?
  - What is the process for curating data?
  - How long should data be kept?
  - How should data collection and curation be acknowledged?
Research Data Infrastructure Stakeholders

- Research Funding Agencies
- Individual Scientists and Scholars
  - Data collection/analysis, managing teams/technology
- Academic Institutions
  - Academic Leadership: Regulations, Governance, Financial Management
  - Research Computing
  - University Libraries: Maintain knowledge resources, provide access, steward
  - Schools and Departments
The DCC Curation Lifecycle Model

Full Lifecycle Actions
- Conceptualise
- Create or Receive
- Appraise and Select
- Ingest
- Preservation Action
- Store
- Access, Use and Reuse
- Transform
- Preserve

Sequential Actions
- Migrate
- Reappraise
- Community Watch and Participation
- Curate
- Preservation Planning
- Description
- Representation Information

Occasional Actions
- Dispose
- Reappraise
- Dispose

Data Curation Lifecycle

Data, any information in binary digital form, is at the centre of the Curation Lifecycle. This includes:
- Simple Digital Objects are discrete digital items; such as textual files, images or sound files, along with their related identifiers and metadata.
- Complex Digital Objects are discrete digital objects, made by combining a number of other digital objects, such as websites. Structured collections of records or data stored in a computer system.

Data (Digital Objects or Databases)

- Migrate data to a different format. This may be done to accord with the storage environment or to ensure the data's immunity from hardware or software obsolescence.
- Return data which fails validation procedures for further appraisal and reselection.
- Dispose of data, which has not been selected for long-term curation and preservation in accordance with documented policies, guidance or legal requirements. Typically data may be transferred to another archive, repository, data centre or other custodian. In some instances data is destroyed. The data's nature may, for legal reasons, necessitate secure destruction.

D. Koop, CSCI 640/490, Spring 2023
Sequential Actions in Data Curation

• Conceptualize: Plan creation of data—capture method and storage options.
• Create or Receive: Create/receive data and make sure metadata exists
• Appraise and Select: Evaluate data and select for long-term curation and preservation
• Ingest: Transfer data to an archive, repository, data centre or other custodian
• Preservation Action: Data cleaning, validation (ensure that data remains authentic, reliable and usable)
• Store: Store the data in a secure manner adhering to relevant standards
• Access, Use and Reuse: Make sure is accessible to users and reusers
• Transform: Create new data from the original (migrate formats, subsets, etc.)
FAIR Principles

- **Findable**: Metadata and data should be easy to find for both humans and computers.

- **Accessible**: Users need to know how data can be accessed, possibly including authentication and authorization.

- **Interoperable**: Can be integrated with other data, and can interoperate with applications or workflows for analysis, storage, and processing.

- **Reusable**: Optimize the reuse of data. Metadata and data should be well-described so they can be replicated and/or combined in different settings.
Findable: DataCite Workflow

1. Take a dataset
2. Describe it
   - Title
   - Authors
   - Year
   - Description
   - And others…
3. Assign a DOI
   - 10.1234/exampledata

Diagram:
- ENSO dataset from NST
- Graph showing data points

Prefix
Prefix
Prefix
Prefix

https://doi.org/10.5438/n138-z3mk
Accessible: DOI to Landing Page with Metadata

Metadata mark-up

Citation → PID resolution → Landing Page → web service → Data

Document citing the data

Repository housing the data

Data store

[M. Fenner et al., 2019]
Interoperable: Standard vocabularies
Reusable: Licensing

• Citation of a dataset is expected as a scholarly norm, not by law
• CC0:
  - "I hereby waive all copyright and related or neighboring rights together with all associated claims and causes of action with respect to this work to the extent possible under the law"
• CC BY: license, not a waiver as CC0
  - "You must give appropriate credit, provide a link to the license, and indicate if changes were made."
• Data Use Agreements (DUA): Used when data are restricted due to proprietary or privacy concerns.
Reusable: Data Citation & Metrics

[H. Cousijn et al., 2019]
Assignment 5

• Divvy Bikes Data
• Spatial, Graph, and Temporal Data Processing
• Use pandas, geopandas, neo4j, (modin for extra credit)
geopandas example
Provenance
What actually happened in a computational experiment?
Provenance in Art

Rembrandt van Rijn
Dutch, 1606 - 1669
Self-Portrait, 1659
oil on canvas
Andrew W. Mellon Collection
1937.1.72

Provenance


[1] This early provenance is established by presence of a mezzotint after the portrait by R. Earlom (1743-1822), dated 1767. See John Charrington, A Catalogue of the Mezzotints After, or Said to Be After, Rembrandt, Cambridge, 1923, no. 49.

Associated Names

- Buccleuch, Henry, 3rd Duke of
- Buccleuch, John Charles, 7th Duke of
- Colnaghi & Co., Ltd., P. & D.
- Knoedler & Company, M.
- Mellon, Andrew W.
- Mellon Educational and Charitable Trust, The A.W.
- Montagu, and 4th Earl of Cardigan, George, 3rd Duke of
Provenance in Art

Rembrandt van Rijn
Dutch, 1606 - 1669
**Self-Portrait, 1659**
oil on canvas
Andrew W. Mellon Collection
1937.1.72

Provenance


[1] This early provenance is established by presence of a mezzotint after the portrait by R. Earlom (1743-1822), dated 1767. See John Charrington, A Catalogue of the Mezzotints After, or Said to Be After, Rembrandt, Cambridge, 1923, no. 49.

Associated Names

- Buccleuch, Henry, 3rd Duke of
- Buccleuch, John Charles, 7th Duke of
- Colnaghi & Co., Ltd., P. & D.
- Knoedler & Company, M.
- Mellon, Andrew W.
- Mellon Educational and Charitable Trust, The A.W.
- Montagu, and 4th Earl of Cardigan, George, 3rd Duke of
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

Computation

Data Management

Provenance

Visualization

Publishing
Evolution of Publication

• Publish paper
• Publish code
• Publish computational experiments/tests
• Publish provenance (what actually happens during your runs)
Galilei Conjugates of Topological Phases


Microsoft Research, Station Q, University of California, Santa Barbara, CA 93106, USA

Theoretische Physik, ETH Zurich, 8093 Zurich, Switzerland

(Dated: July 6, 2011)

Galilei conjugation relates unitary conformal field theories (CFTs) and topological quantum field theories (TQFTs) to non-unitary counterparts. Here we investigate Galois conjugates of quantum double models, such as the Levin-Wen model. While these Galilei conjugated Hamiltonians are typically non-Hermitian, we find that their ground state wave functions still obey a generalized version of the usual closed property (local operators do not act on the ground state manifold) and hence enjoy a generalized topological protection. The key question addressed in the paper is whether such non-unitary topological phases can ever act as the ground states of Hermitian Hamiltonians. Specific attempts at constructing Hermitian Hamiltonians with these ground states lead to a clash of the poles property and topological protection of the degenerate ground state. Beyond this we also find that Galilei conjugates of the Levin-Wen model, which arise as low energy models for a gapped 2D quantum mechanic, surprisingly have totally real spectrum, have been found to arise as low energy effective models for a gapped 2D quantum mechanics.

We reach this conclusion quite indirectly. Our main thrust is the investigation of Galilei conjugation in the simplest non-Abelian Levin-Wen model. This model, which is also called "DHLF" (a topological quantum field theory (TQFT) whose states are strings on a surface labeled by either a trivial or "Fibonacci" anyon. From this starting point, we give a rigorous argument that the "Fibonacci" ground state cannot be locally conjugated to the ground state of any topological phase, within a Hermitian model satisfying Lieb-Robinson (LR) bounds (which includes is not limited to gapped local and quasi-local Hamiltonians).

In relativistic case there can be some exponentially small "leakage" outside the light-cone. The LR velocity is set by microscopic details of the Hamiltonian, such as the interaction strength and range. Combining the LR bounds with the spectral gap enables us to prove locality of various correlation and response functions. We will prove a Hamiltonian is a Lieb-Robinson bound, if and only if it does not have a gap.

We work primarily with a single example, but it should be clear that the concept of Galilei conjugation can be widely applied to TQFTs. The essential idea is to relate quantum systems by a Galilean transformation. The LR velocity is set by microscopic details of the Hamiltonian, such as the interaction strength and range. Combining the LR bounds with the spectral gap enables us to prove locality of various correlation and response functions. We will prove a Hamiltonian is a Lieb-Robinson bound, if and only if it does not have a gap.

Our method is not restricted to Galilei conjugated DHLF and its factors FH and FLF, but can be generalized to infinitely many non-unitary TQFTs, showing that they will not arise as low energy model for a gapped 2D quantum mechanical system.
Benefits of Provenance-Rich Publications

• Produce more knowledge—not just text
• Allow scientists to stand on the shoulders of giants (and their own)
• Science can move faster!
• Higher-quality publications
• Authors will be more careful
• Many eyes to check results
• Describe more of the discovery process: people only describe successes, can we learn from mistakes?
• Expose users to different techniques and tools: expedite their training; and potentially reduce their time to insight
Provenance Definitions

• Dictionary: "the source or origin of an object; its history and pedigree; a record of the ultimate derivation and passage of an item through its various owners."

• Focus on causality—the sequence of steps that detail how a result was generated and/or derivation—what data a result depended on

• Provenance itself is data, this list of steps along with metadata for each step: when it occurred, who initiated it, notes about it

• Can be used to preserve information about an experiment and to answer many questions
Workflows

- Abstract computation
- Computational modules connected through input and output ports
- Data flows along the connections

DATA → vtkStructuredPointsReader → vtkContourFilter → vtkDataSetMapper → vtkCamera → vtkActor → vtkRenderer → VTKCell → IMAGE
Provenance Graph
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?
Questions about Provenance

• How does one capture provenance?
• How does one manage provenance for later use?
• How do we answer questions about our provenance?
• How do we use provenance for good?
Provenance Management

• Provenance can be generated from tasks/programs/scripts/etc.

• Properties of provenance are related to the computational model
  - a specific application with a graphical interface
  - a script that automates the use of several command-line tools
  - a scientific workflow that combines several tools
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
User-defined provenance

- Goal: capture lots of provenance automatically based on what steps are executed
- Problem: not everything can be captured automatically
- Annotations offer ability to keep notes about processes
- Users might also specify known causal links that cannot be automatically determined (e.g. a step depends on three system files that were not specified as inputs in the workflow)
Provenance Management

• What is needed to capture, store, and use provenance?
  1. Capture mechanism
  2. Model for representing provenance
  3. Tools to store, query, and analyze provenance
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".
Provenance Granularity

• How detailed should our provenance be?
  - **Coarse**: "This program ran with inputs x, y, z and produced outputs a, b, c"
  - **Fine**: "Input x was read into register 4, input y was read in register 5, add operation was performed using registers 4 and 5, …"

• More queries are possible with fine-grained provenance, but…
  - Storage concerns
  - Performance concerns

• Abstraction can help here
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)

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

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

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

style = vtk.vtkInteractorStyleTrackballCamera()
iren = vtk.vtkRenderWindowInteractor()
iren.SetRenderWindow(renwin)
iren.SetInteractorStyle(style)
iren.Initialize()
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
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 $\rightarrow$ activity
  - used: activity $\rightarrow$ entity
  - wasDerivedFrom: entity $\rightarrow$ entity
Querying Provenance

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

**REDUX**

```sql
SELECT Execution.ExecutableWorkflowId, Execution.ExecutionId, Event.EventId, ExecutableActivity.ExecutableActivityId
from Execution, Execution_Event, Event, ExecutableWorkflow_ExecutableActivity, ExecutableActivity,
  ExecutableActivity_Property_Value, Event.Property, Event.Type as ET
where Execution.ExecutionId = Execution_Event.ExecutionId
  and Event.EventId = Event.EventId
  and ExecutableActivity.ExecutableActivityId = ExecutableActivity_Property_Value.ExecutableActivityId
  and (CONVERT(DECIMAL, Event.Timestamp) = 0) and Execution_Event.ExecutableWorkflow_ExecutableActivityId =
  ExecutableWorkflow_ExecutableActivity.ExecutableWorkflow_ExecutableActivityId
  and ExecutableWorkflow_ExecutableActivity.ExecutableActivityId = ExecutableActivity.ExecutableActivityId
and Event.EventTypeId = ET.EventTypeId and ET.EventName = 'Activity Start';
```

**VisTrails**

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

**MyGrid**

```sql
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:HXQOVQAZZ10>
  ?inputParameter <ont:mode> <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

- 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

12ab3-45ef2... → OBTAIN INPUT REFS → QUERY PROVENANCE

0ab678cd... → FILE STORE

12ab3-45ef2... → OBTAIN FILE REFERENCE

QUERY FILE STORE

OBTAIN INPUT FILES

12ab3-45ef2...

input files

[Koop et al, 2010]
# 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-Enabled Systems

<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 filesystems</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</td>
<td>Yes</td>
</tr>
<tr>
<td>Pegasus</td>
<td>RDBMS</td>
<td>SPARQL for metadata and workflow;</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SQL for execution log</td>
<td></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

<table>
<thead>
<tr>
<th>Agencies</th>
<th>ExternalTours</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>destination</td>
</tr>
<tr>
<td>t1:</td>
<td>San Francisco</td>
</tr>
<tr>
<td>t2:</td>
<td>Santa Cruz</td>
</tr>
<tr>
<td>t3:</td>
<td>Santa Cruz</td>
</tr>
<tr>
<td>t4:</td>
<td>Monterey</td>
</tr>
<tr>
<td>t5:</td>
<td>Monterey</td>
</tr>
<tr>
<td>t6:</td>
<td>Carmel</td>
</tr>
</tbody>
</table>

Q1:
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 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 \(t_5\) and \(t_6\)—only one is ok

[Cheney et al., 2007]
How Provenance

- 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 \textit{twice}
- Uses provenance semirings (the "polynomial" shown on the right)

---

D. Koop, CSCI 640/490, Spring 2023
Where Provenance

Intuitively, the two source tuples witness the existence of the tuple of shown above on the right. According to Cui et al., the lineage of the out-

where Agencies( of the two source tuples justify the existence of the HarborCruz tuple.

\[ \text{Fig. 1.1 Our example database: an online travel portal.} \]

Tours, 415-1200) is the union of the lineage of the intermediate subset of the input database records that is su-
tuples are part of the witness since they do not contribute to the Har-

borCruz output tuple according to

\[ \text{In other words, the source tuples } t_1 : (\text{BayTours, San Francisco, 415-1200}), t_2 : (\text{HarborCruz, Santa Cruz, 831-3000}) \text{ of the two source tuples justify the existence of the HarborCruz tuple.} \]

The result of \( Q \) executed on our example database in Figure 1.1 is

\[ \text{Result of } Q_1: \]

<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>

\[ \text{WHERE a.name } = \text{ e.name AND e.type } = \text{ 'boat'} \]

> [Cheney et al., 2007]

- 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: 
  \( (t_2, \text{name}) \)
- Important in annotation-propogation
Evolution Provenance
Data Exploration

[Modified from Van Wijk, Vis 2005]
Data Exploration

- 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

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
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
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 “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
Execution Provenance

```xml
<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"/>
...
```
Capturing and querying fine-grained provenance of preprocessing pipelines in data science

A. Chapman, P. Missier, L. Lauro, R. Torlone