Provenance-enabled Automatic Data Publication

James Frew, Greg Janée, and Peter Slaughter
Earth Research Institute, University of California, Santa Barbara
Unrecognised Leonardo da Vinci portrait revealed by his fingerprint
At the toolbar (menu, whatever) associated with a document there is a button marked "Oh, yeah?". You press it when you lose that feeling of trust. It says to the Web, "so how do I know I can trust this information?"

... The result of pressing on the "Oh, yeah?" button is either a list of assumptions on which the trust is based, or of course an error message indicating either that a signature has failed, or that the system couldn't find a path of trust from you to the page.

— Tim Berners-Lee (1995)
Provenance: “working’ definition

- Provenance of a resource is a record that describes entities and processes involved in producing and delivering or otherwise influencing that resource. Provenance provides a critical foundation for assessing authenticity, enabling trust, and allowing reproducibility. Provenance assertions are a form of contextual metadata and can themselves become important records with their own provenance.

—W3C Provenance Incubator Group (2010)
from 50 Kft to 0.5 ft
mosaic.sh:
  mosaicFn="MOD09GA.A2008019.sn.005.hdf"  mrtmosaic -i tile.lis -o $mosaicFn resample -p MRT.prm -g MRT.log

tile.lis:
  MOD09GA.A2008019.h08v04.005.2008022125449.hdf
  MOD09GA.A2008019.h08v05.005.2008022134646.hdf
  MOD09GA.A2008019.h09v04.005.2008022151755.hdf

MRT.prm:
  INPUT_FILENAME=./MOD09GA.A2008019.sn.005.hdf
  SPATIAL_SUBSET_TYPE=INPUT_LAT_LONG
  SPATIAL_SUBSET_UL_CORNER=(41.5000 -122.4000)
  SPATIAL_SUBSET_LR_CORNER=(35.0000 -117.6000)
  OUTPUT_FILENAME=MOD09GA.A2008019.sn_cal-aea.005.Refl.hdf
  RESAMPLING_TYPE=NN  OUTPUT_PROJECTION_TYPE=AEA  DATUM=WGS84
  OUTPUT_PROJECTION_PARAMETERS=(0.0 0.0 34.00 40.50 -120.00 \ 0.00 0.00 -40000000.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00)
  OUTPUT_PIXEL_SIZE=500  SPECTRAL_SUBSET=(0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0)
provenance in ES3

• input file(s) → process → output file(s)
• collected automatically by tracing
  - process creation
  - program execution
  - filesystem I/O
ES3 architecture

Collector / Data Submission
- Plugin 1
- Plugin 2
- ... (Ellipsis)
- Plugin i

Logger
- Log Files
  - Transmitter
    - XML
      - Web Interface
        - Provenance Store
          - Database

User / Data Request
- XML
  - Database
    - XML / GRAPHML

Core / Data Storage
<ES3Request type="storeTransformation">
  <transformation>
    <timestamp type="execution">20080610T181515Z</timestamp>
    <provenance>
      <link>
        <type>1/0</type>
        <fromUuid>7af82a69-fa7a-4aec-abdf-eb009f5e2cab</fromUuid>
      </link>
    </provenance>
    <collection>/default</collection>
    <workflowUuid>b2189b33-349c-434d-bf73-3f8817dccbd5</workflowUuid>
    <containsWorkflowUuid>2c4310db-4949-4fab-a82e-1282432257c3</containsWorkflowUuid>
    <uuid>197dc9ee-3dbf-447b-871a-e11a0288a7ba</uuid>
    <name>./mosaic.sh</name>
  </transformation>
</ES3Request>
mosaic.sh:
    mosaicFn="MOD09GA.A2008019.sn.005.hdf"
    mrtmosaic -i tile.lis -o $mosaicFn
    resample -p MRT.prm -g MRT.log

tile.lis:
    MOD09GA.A2008019.h08v04.005.2008022125449.hdf
    MOD09GA.A2008019.h08v05.005.2008022134646.hdf
    MOD09GA.A2008019.h09v04.005.2008022151755.hdf

MRT.prm:
    INPUT_FILENAME=
    SPATIAL_SUBSET_I
    SPATIAL_SUBSET_U
    SPATIAL_SUBSET_I
    OUTPUT_FILENAME=
    RESAMPLING_TYPE=
    OUTPUT_PROJECSTIC
    DATUM=WGS84
    OUTPUT_PROJECTIC
    0.00 0.00 -400
    OUTPUT_PIXEL_SIZE
    SPECTRAL_SUBSET=

    MOD09GA.A2008019.h08v04.005.2008022125449.hdf
    MOD09GA.A2008019.h08v05.005.2008022134646.hdf
    MOD09GA.A2008019.h09v04.005.2008022151755.hdf
    /mrtmosaic
    resample.log
    tmpEi6Z73
    resample
    /mrtmosaic
    /resample
    MRT.log
    MOD09GA.A2008019.sn.cal-aea.005.Refl.hdf
    spheroid.txt
    datum.txt
    MRT.prm
data publication

• evaluate object’s antecedents against publication assertions

• if antecedents justify publication, then object is publishable
“publish” tool

- retrieve object’s provenance
- traverse depth-first
- foreach antecedent
  - automatically endorse if assertion valid
  - else manually endorse
- save endorsements in provenance graph
automatic endorsement

• filename patterns
  - if matches a *glob* expression

• version control
  - if == a committed version in a repository

• transitivity
  - if all antecedents are endorsed
manual endorsement

• endorse
  - optional comment

• ignore
  - object is irrelevant

• skip
  - punt for now
example

- ocean color algorithm
example

• ocean color algorithm

• provenance captured by ES3; rendered as dataflow graph

• now, let’s publish→
**Endorsed by:** glob rule: */data/* .bz2

**Annotation:** SeaWiFS reprocessing 5.2

**Assertion check:** file (at time referenced) was unchanged since rule creation (failed assertion may indicate annotation is incorrect)
**Endorsed by:** glob rule: /usr/bin/*

**Annotation:** operating system tool

**Assertion check:** (none)
**Endorsed by:** version control system rule

**Annotation:** GSMS

**Assertion check:** file (at time referenced) corresponded to committed version (failed assertion may indicate uncommitted code was used)
**Endorsed by**: glob rule: /itt/idl/*

**Annotation**: IDL 8.0

**Assertion check**: file (at time referenced) was unchanged since rule creation (failed assertion may indicate annotation is incorrect)
Endorsed by: transitivity
issues & next steps

- granularity
  - read/write file → provenance graph cycle

- compilation
  - versioned source vs. executed binary

- distributed version control
  - single “version” in multiple changesets
Thanks!