

Provenance



Agenda

- What is Provenance?
- Types of Provenance
- Provenance Capture
- Provenance Representation
- Provenance Data x Data Provenance

QUICK REVIEW...

From in-vivo to in-silico



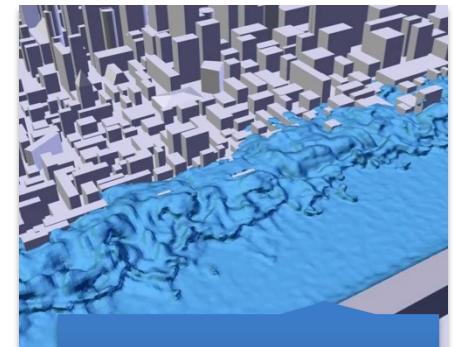
In-vivo



In-vitro



In-virtuo



In-silico



Travassos and Barros "Contributions of in virtuo and in silico experiments for the future of empirical studies in software engineering." WSESE 2003

Dealing with produced data



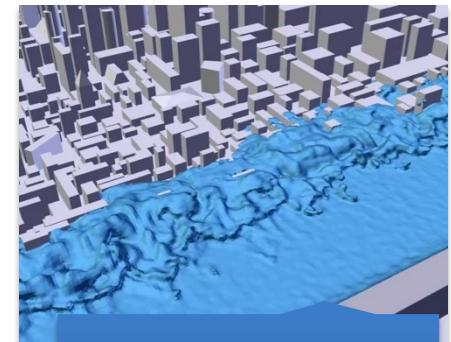
In-vivo



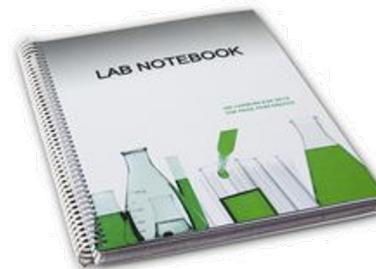
In-vitro



In-virtuo

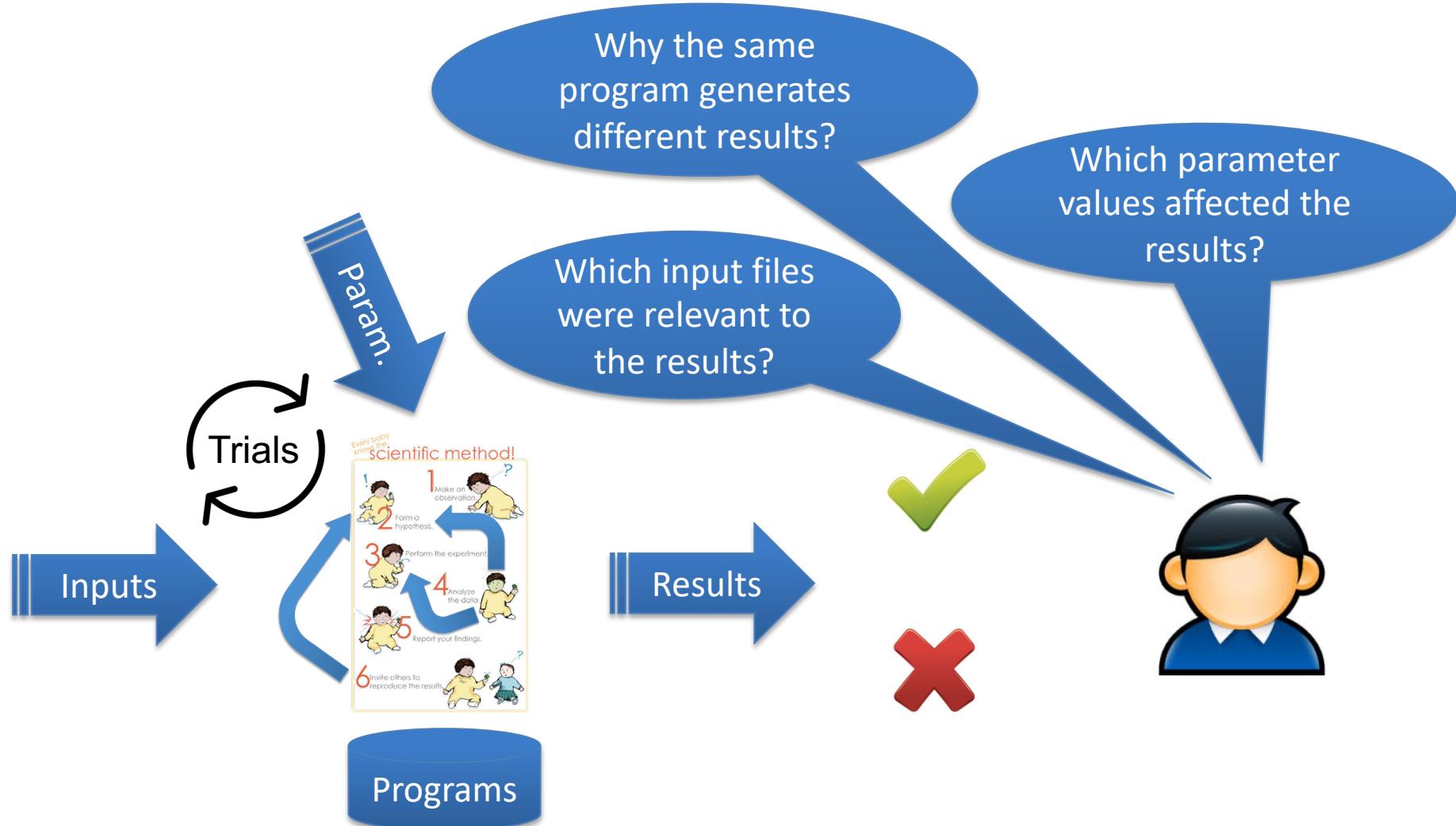


In-silico

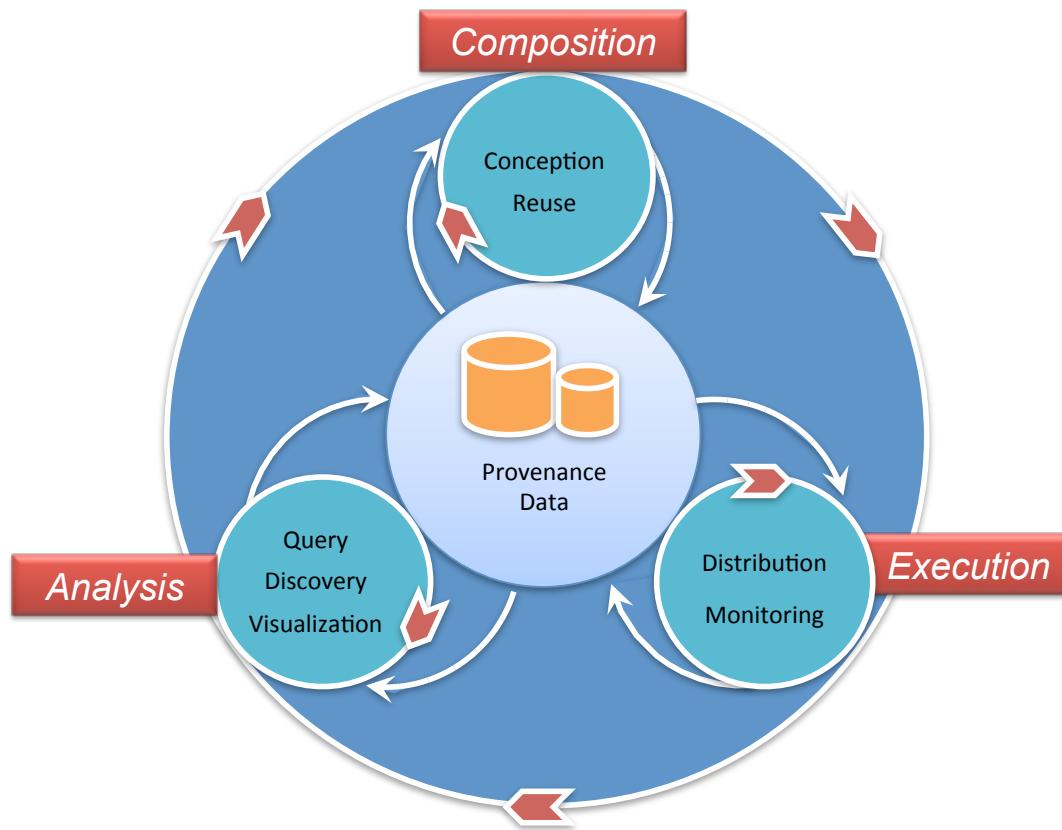


Travassos and Barros "Contributions of in virtuo and in silico experiments for the future of empirical studies in software engineering." WSESE 2003

How to reason from the data?



Provenance is the key!



Mattoso et al. "Towards supporting the life cycle of large scale scientific experiments." IJBPM 5(1) 2010

The Word Provenance

The word Provenance comes from the French Word
Provenir (to come from)

Provenance in the Dictionaries

“Origin; source.”

Merriam-Webster Online Dictionary

“The place of origin or earliest known **history** of something; origin; **derivation**; A record of **ownership** of a work of art or an antique, used as a guide to authenticity or quality”

Oxford Dictionary



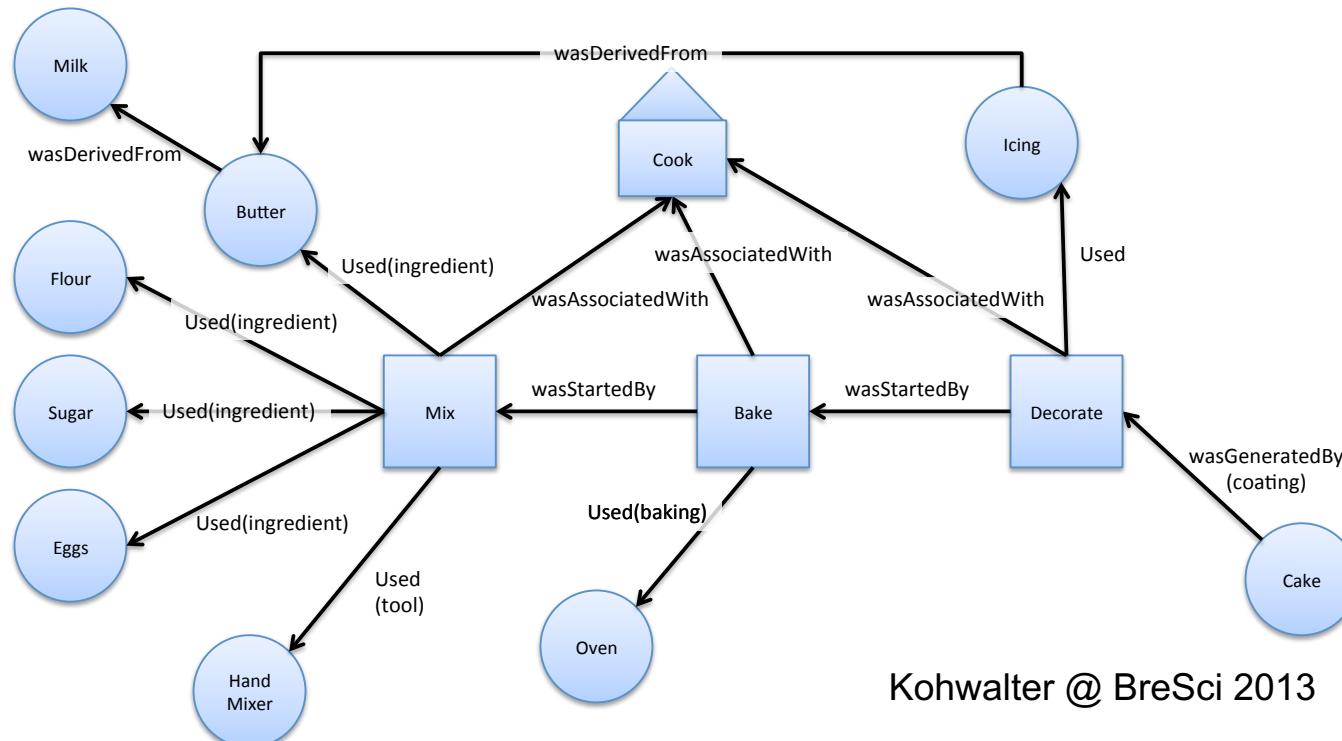
Diana and Actaeon by Titian has a full provenance covering its passage through several owners and four countries since it was painted for Philip II of Spain in the 1550s.



La Bella Principessa, a recent rediscovery said to be by Leonardo da Vinci, whose provenance is still the subject of research and controversy.

Source: <https://en.wikipedia.org/wiki/Provenance>

Example of Provenance



“Life can only be understood backwards; but it must be lived forwards.”

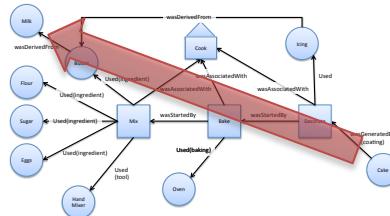
Søren Kierkegaard, 1843.

Provenance in E-Science

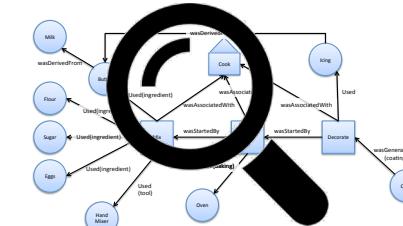
“The provenance (also referred to as the audit trail, lineage, and pedigree) of a data product contains **information about the process and data** used to derive the data product. It provides important documentation that is key to **preserving** the data, to **determining the data’s quality** and **authorship**, and to **reproduce** as well as **validate** the results. These are all important requirements of the **scientific process**.”

Davidson and Freire, SIGMOD 2008
(Tutorial)

Provenance is useful for...

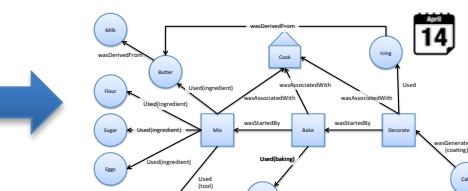
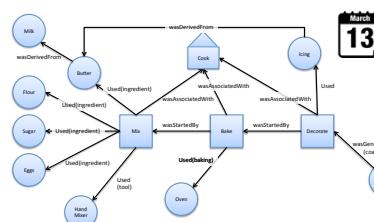


Interpreting and understanding results

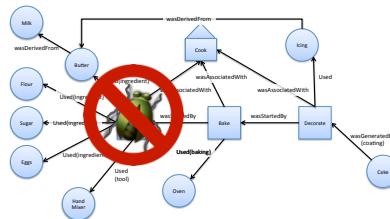


Auditing

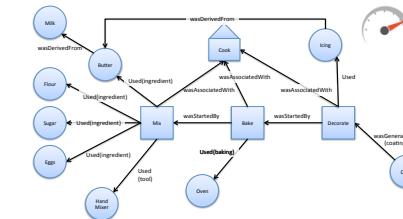
Reproducibility



Caching

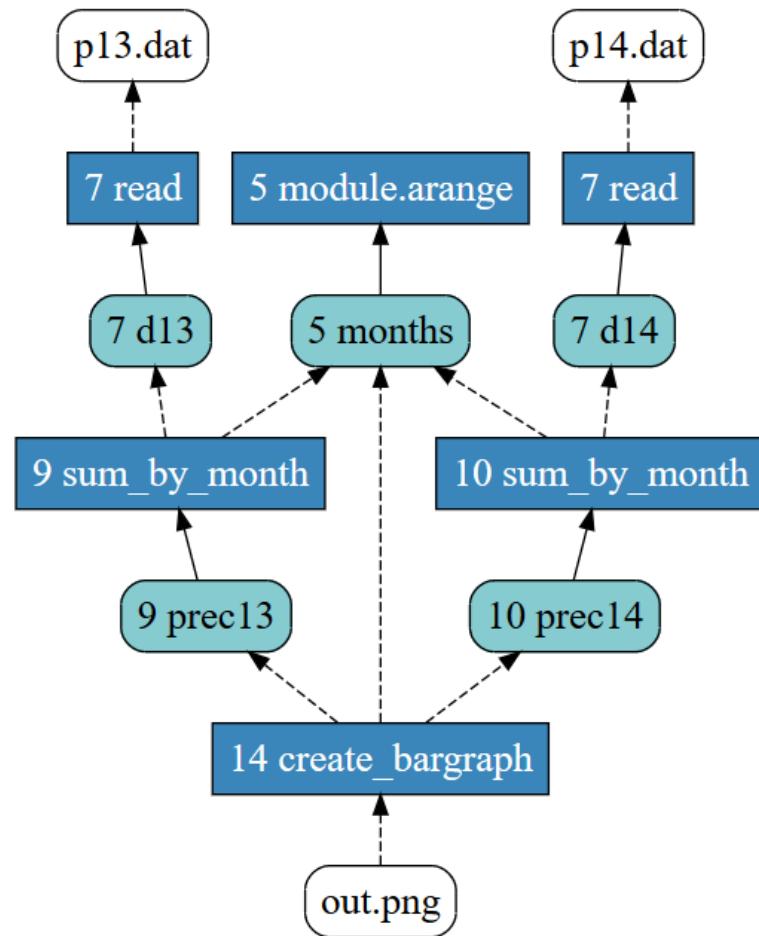
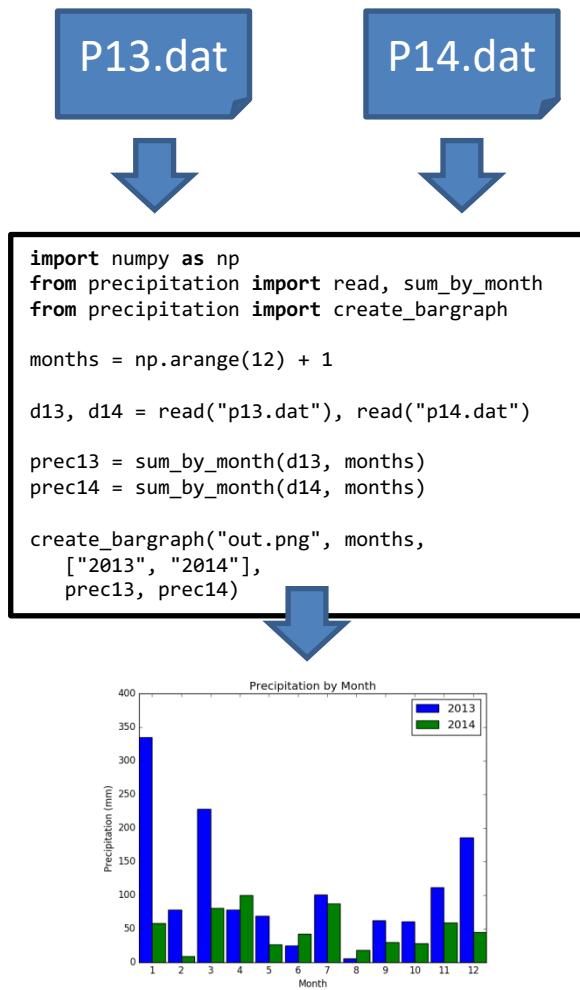


Debugging



Among other applications (attribution, reuse, trustworthiness, sharing, ...)

Example



Provenance in Science is not new

- Scientists have been registering provenance for a long time

DNA Recombination Experiments

(J. Lederberg, 1946)

Recombination Tests

When

19 JUN 1946

Test:	B	M	BM	P	T	PT	BM Tmp
23T-12				+	++		
243-6-	=B	=P	=T.				
1	++	+	++				
2	++	+	++				
3	++	++	++				
4	++	++	++				
5	++	++	++				
6	+	+	++				
7	+	+	++				
8	+	+	++				
9	"	"	"				
10	"	"	"				
11	"	"	"				
12	-	# -	+				
13	"	+	++				
14	-	-	++				
15	++	-	++				
16	++	++	++				
17	+	+	++				
18	+	+	++				
19	+	+	++				
20	+	+	++				
21	++	++	++				
22	do.						
23	do.						
24	do.						
25	do.						
26	++	-	++				
27	++	+	+				
28	++	+	++				
29	++	-	++				
30	++	+	+				
31	++	+	++				
32	++	++	++				
33	++	+	++				
34	++	++	++				
35	++	++	++				
36							
37							
38							
39							
40							

Most of this is clearly synaptonemal.

Struck out
(short code). (Hooray!). See c.

Annotation

- - - - *Not coli.*

- - -

Struck out.

Struck out

Data

A STORY TOLD IN FILE NAMES:

Location: C:\user\research\data

Filename	Date Modified	Size	Type
data_2010.05.28_test.dat	3:37 PM 5/28/2010	420 KB	DAT file
data_2010.05.28_re-test.dat	4:29 PM 5/28/2010	421 KB	DAT file
data_2010.05.28_re-re-test.dat	5:43 PM 5/28/2010	420 KB	DAT file
data_2010.05.28_calibrate.dat	7:17 PM 5/28/2010	1,256 KB	DAT file
data_2010.05.28_huh??.dat	7:20 PM 5/28/2010	30 KB	DAT file
data_2010.05.28_WTF.dat	9:58 PM 5/28/2010	30 KB	DAT file
data_2010.05.29_aaarrgh.dat	12:37 AM 5/29/2010	30 KB	DAT file
data_2010.05.29_#\$@*!!?.dat	2:40 AM 5/29/2010	0 KB	DAT file
data_2010.05.29_crap.dat	3:22 AM 5/29/2010	437 KB	DAT file
data_2010.05.29_notbad.dat	4:16 AM 5/29/2010	670 KB	DAT file
data_2010.05.29_woohoo!!?.dat	4:47 AM 5/29/2010	1,349 KB	DAT file
data_2010.05.29_USETHISONE.dat	5:08 AM 5/29/2010	2,894 KB	DAT file
analysis_graphs.xls	7:13 AM 5/29/2010	455 KB	XLS file
ThesisOutline.doc	7:26 AM 5/29/2010	38 KB	DOC file
Notes_Meeting_with_ProfSmith.txt	11:38 AM 5/29/2010	1,673 KB	TXT file
JUNK...	2:45 PM 5/29/2010		Folder
data_2010.05.30_startingover.dat	8:37 AM 5/30/2010	420 KB	DAT file

Type: Ph.D Thesis Modified: too many times Copyright: Jorge Cham www.phdcomics.com



When

19 JUN 1946

Test:	B	M	BM	P	T	PT	BM Typ
23T-12				++			
243-6-	-B	-P	-BM ++	-M	= -O	0	++ OK.
1	++	+	++				
2	++	+	++				
3	++	++	++				
4	++	++	++				
5	++	++	++				
6	*	*	*				
7	*	*	"				
8	*	*	"				
9	"	"	"				
10	"	"	"				
11	"	"	"				
12	-	◆ -	+				Struck out
13	-	+	++				
14	-	-	++				(short rods) (Hooray!). See c.
15	++	-	++				Struck out
16	++	++	++				
17	*	*	"				
	◆						
	◆						
238-1				-	-		Not coli.
238-2	n.g. co			.	.		
243-9.	F from BT Plate.						

Annotation

Not Scalable!
Not Searchable!

A STORY TOLD IN FILE NAMES:

Filename	Date Modified	Size	Type
data_2010.05.28_test.dat	3:37 PM 5/28/2010	420 KB	DAT file
data_2010.05.28_re-test.dat	4:29 PM 5/28/2010	421 KB	DAT file
data_2010.05.28_re-re-test.dat	5:43 PM 5/28/2010	420 KB	DAT file
data_2010.05.28_calibrate.dat	7:17 PM 5/28/2010	1,256 KB	DAT file
data_2010.05.28_huh???.dat	7:20 PM 5/28/2010	30 KB	DAT file
data_2010.05.28_WTF.dat	9:58 PM 5/28/2010	30 KB	DAT file
data_2010.05.29_aaarrgh.dat	12:37 AM 5/29/2010	30 KB	DAT file
data_2010.05.29_#\$@*!!?.dat	2:40 AM 5/29/2010	0 KB	DAT file
data_2010.05.29_crap.dat	3:22 AM 5/29/2010	437 KB	DAT file
data_2010.05.29_notbad.dat	4:16 AM 5/29/2010	670 KB	DAT file
data_2010.05.29_woohoo!!.dat	4:47 AM 5/29/2010	1,349 KB	DAT file
data_2010.05.29_USETHISONE.dat	5:08 AM 5/29/2010	2,804 KB	DAT file

Metadata is on file names!
Not Scalable!
Not Searchable!

Placing order into the chaos

- In the past years, several tools appeared aiming at helping scientists conduct their experiments
- Several of them **automatically** capture different **types of provenance**

Types of Provenance

- **Prospective Provenance**
 - it captures a computational task's **specification** (whether it's a **script** or a **workflow**) and corresponds to the steps (or recipe) that must be followed to generate a data product or class of data products
- **Retrospective Provenance**
 - captures the **steps executed** as well as **information about the environment** used to derive a specific data product
 - it's a detailed log of a computational task's execution and the **data** (input/output) involved in the execution

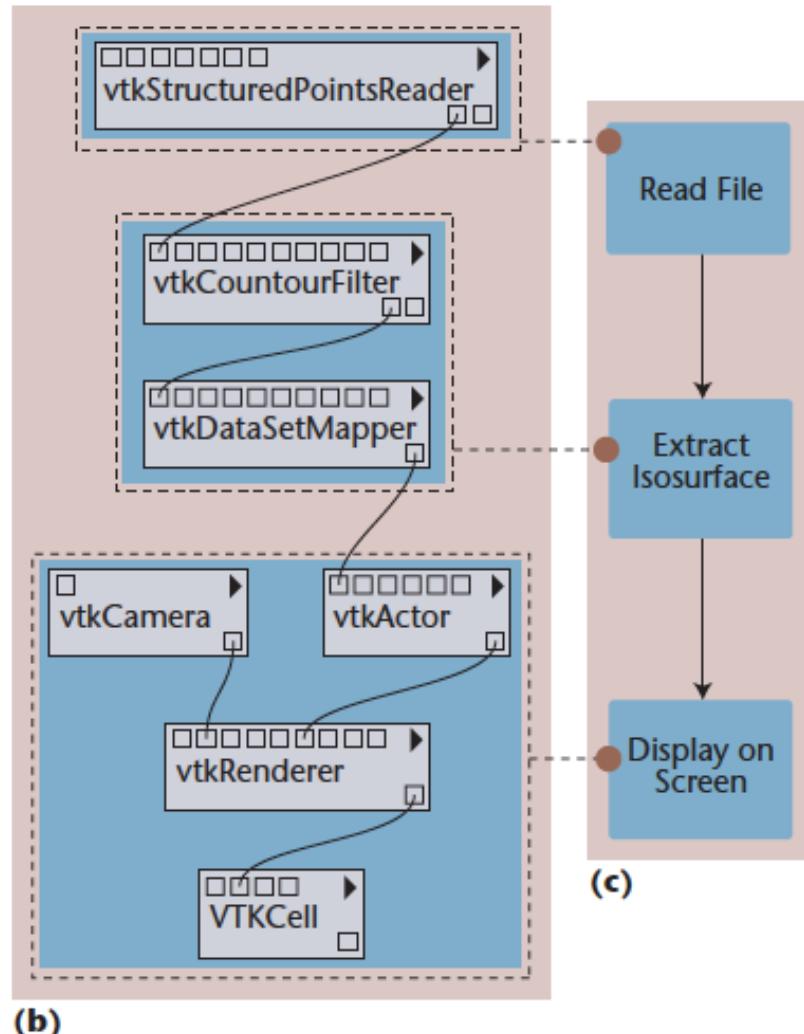
Scripts and Workflows

- **Scripts** and **Workflows** are usually the ways in which scientists materialize their experiments
- Some scientists use also **regular programming languages** such as Fortran or C
- Some popular scripting language for science are Python, R and MatLab (among others)
- Popular workflow systems include Swift, Vistrails, Kepler, Taverna, SciCumulus (among others)

```

1 import vtk
2
3 data = vtk.vtkStructuredPointsReader()
4 data.SetFileName("../examples/data/head.120.vtk")
5
6 contour = vtk.vtkContourFilter()
7 contour.SetInput(0, data.GetOutput())
8 contour.SetValue(0, 67)
9
10 mapper = vtk.vtkPolyDataMapper()
11 mapper.SetInput(contour.GetOutput())
12 mapper.ScalarVisibilityOff()
13
14 actor = vtk.vtkActor()
15 actor.SetMapper(mapper)
16
17 cam = vtk.vtkCamera()
18 cam.SetViewUp(0,0,-1)
19 cam.SetPosition(745,-453,369)
20 cam.SetFocalPoint(135,135,150)
21 cam.ComputeViewPlaneNormal()
22
23 ren = vtk.vtkRenderer()
24 ren.AddActor(actor)
25 ren.SetActiveCamera(cam)
26 ren.ResetCamera()
27
28 renwin = vtk.vtkRenderWindow()
29 renwin.AddRenderer(ren)
30
31 style = vtk.vtkInteractorStyleTrackballCamera()
32 iren = vtk.vtkRenderWindowInteractor()
33 iren.SetRenderWindow(renwin)
34 iren.SetInteractorStyle(style)
35 iren.Initialize()
36 iren.Start()

```



Source: Freire et al., 2008. Provenance for Computational Tasks: A Survey.

Extending the Types of Provenance to Accommodate Scripts

Prospective Provenance
(workflow structure)

Retrospective Provenance

Workflows
Scripts

Definition Provenance
(script structure, including function definitions, their arguments, and function calls)

Deployment Provenance
(execution environment, including info about the OS, environment variables, libraries in which the script depends on)

Execution Provenance
(function activations, argument values, return values, variable values)

Source: Freire et al., 2008. Provenance for Computational Tasks: A Survey.
Murta et al., 2014. noWorkflow: Capturing and Analyzing Provenance of Scripts

Managing Provenance

- A Provenance Management System consists of three main components
 - A capture mechanism
 - A representation model
 - An infrastructure (for storage, access and queries)

Managing Provenance

- A Provenance Management System consists of three main components
 - A **capture mechanism**
 - A representation model
 - An infrastructure (for storage, access and queries)

Capture Mechanisms

- A provenance capture mechanism needs access to a computational task's relevant details, such as its steps, execution information, and user-specified annotations
- Such mechanisms can be classified in three types
 - Operating System (OS)-based
 - Workflow-based
 - Process-based

OS-Based

- OS-based capture systems rely on the Operating System functionalities to capture provenance
 - (+) No modification to the experiment script or program is needed
 - (-) Provenance is captured in very fine grain
 - (-) Only retrospective provenance is captured

Source: Frew, J., Metzger, D., Slaughter, P., 2007. Automatic capture and reconstruction of computational Provenance.

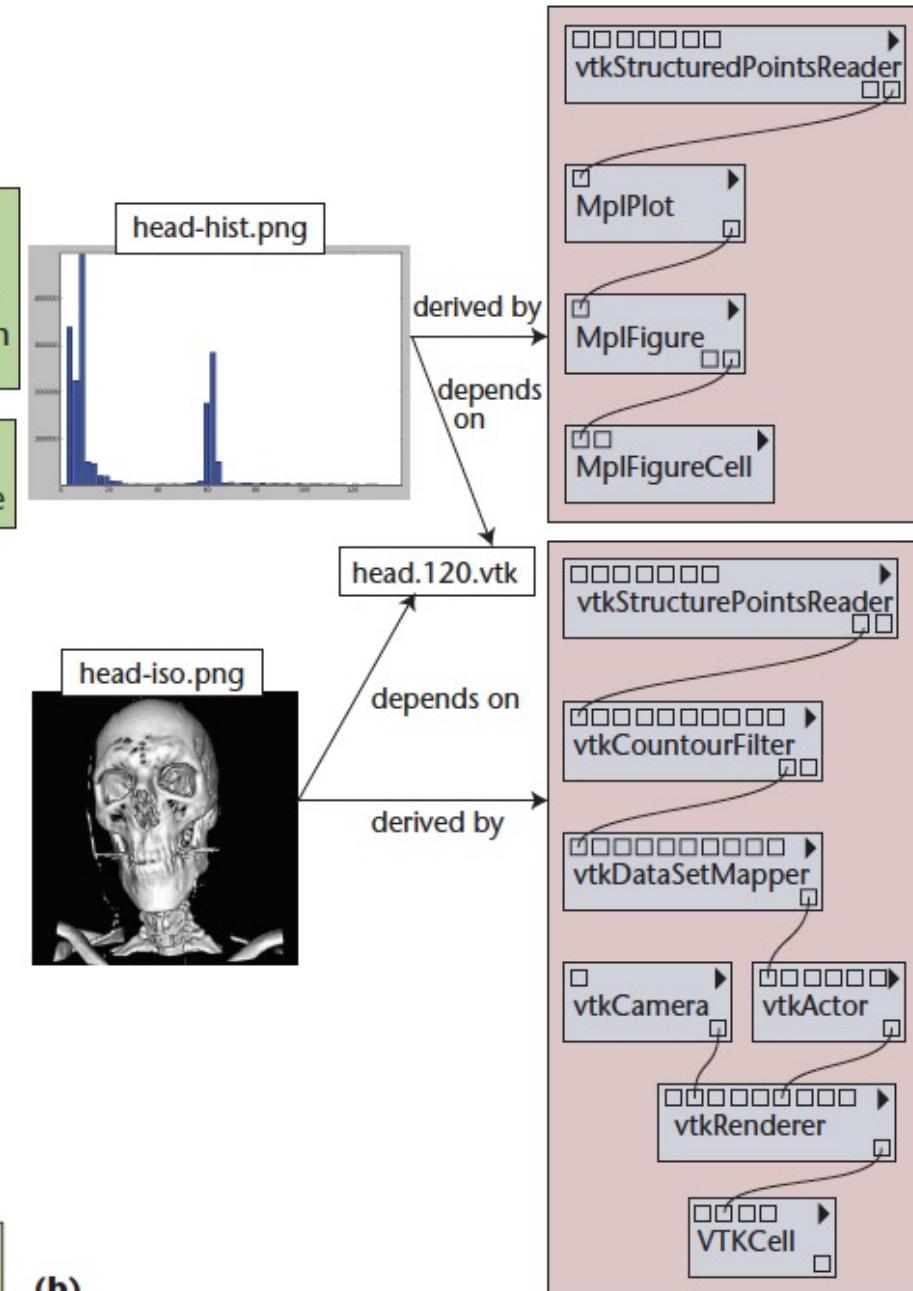
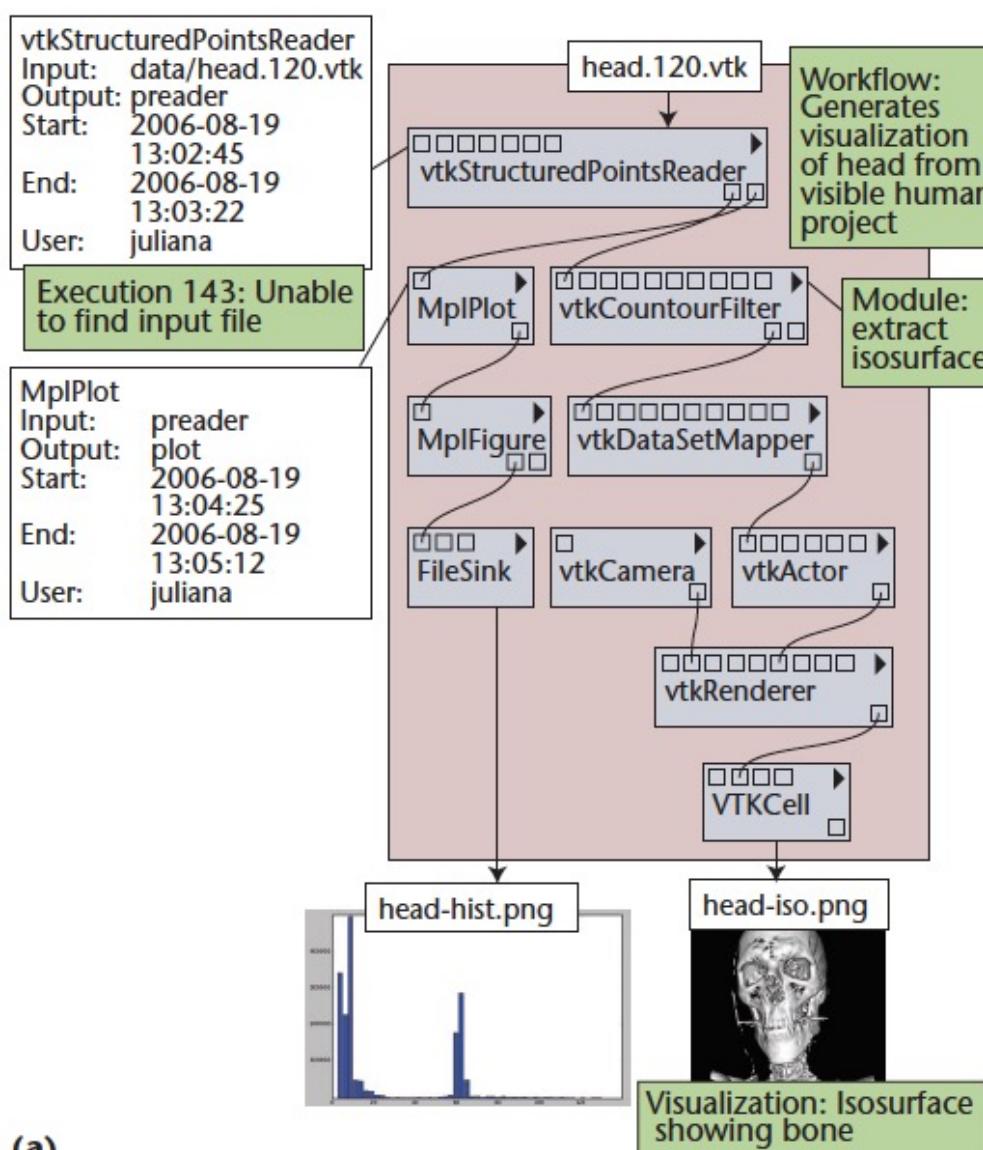
Example of Provenance Captured by an OS-Based System (ES3)

```
...
<exec time="20060908T211419.747511Z" routine="/AIR5.2.5/bin/align_warp"
pid="22636">
  <io>
    <pipe read="true" id="std-in"/>
    <pipe write="true" id="std-out"/>
    <pipe write="true" id="std-err"/>
    <file read="true">/etc/ld.so.cache</file>
    <file read="true">/lib/libm.so.6</file>
    <file read="true">/lib/libc.so.6</file>
    <file read="true">/home/es3/provenance_challenge/anatomy1.hdr</file>
    <file read="true">/home/es3/provenance_challenge/reference.hdr</file>
  </io>
...

```

Workflow-based

- Workflow-based capture mechanisms are either attached or integrated into a Workflow System
-
- (+) Capturing provenance is straightforward
 - (+) It is able to capture both prospective and retrospective provenance
 - (+) They connect prospective and retrospective provenance
 - (-) Experiment needs to run as a workflow



Source: Freire et al., 2008. Provenance for Computational Tasks: A Survey.

Scripts?

- The classification of workflow-based capture was made in a time where there was very little support for capturing provenance of scripts (2008)
- Today, there are several tools that capture provenance from scripts in a similar way the workflow-based approaches work – transparently and with little to no effort

Process-Based

- Process-based mechanisms require each activity to capture its own provenance
 - (+) It works on any workflow system
 - (-) It requires instrumentation
 - (-) Only retrospective provenance is captured, unless the prospective provenance is captured during the instrumentation process

Example of Instrumentation

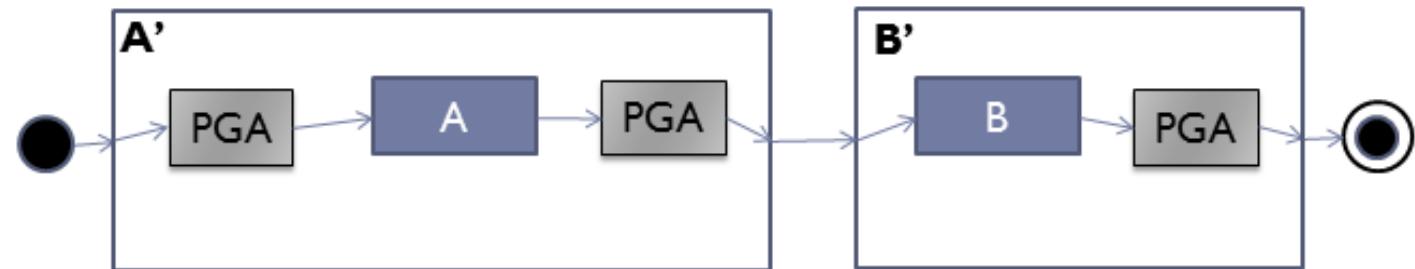
I - Original workflow



II - Instrumenting workflow



III - Wrapping activities



IV - Adapted workflow

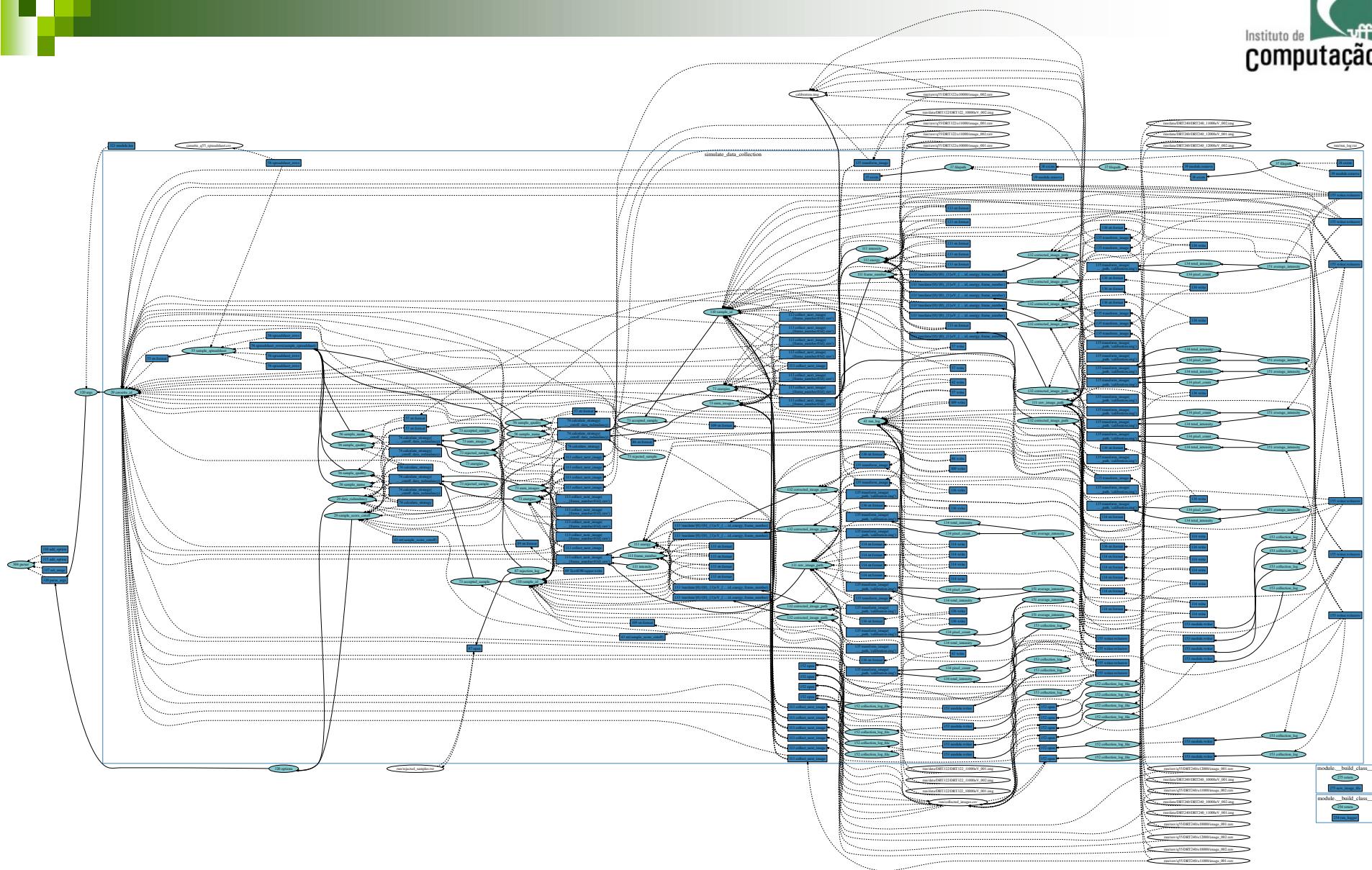


PGA: Provenance Gathering Activity

Source: Marinho et al., 2012. ProvManager: A Provenance Management System for Scientific Workflows.

Granularity

- Different systems capture provenance at different granularity
- There is a trade-off that needs to be taken into account regarding granularity
- **Coarse-grain** provenance requires less storage space, but is less detailed
- **Fine-grain** provenance requires lots of storage space, but can be filtered to remove unwanted information during the analysis phase



Source: Pimentel et al., 2016 . Yin & Yang: Demonstrating Complementary Provenance from noWorkflow & YesWorkflow.

Managing Provenance

- A Provenance Management System consists of three main components
 - A capture mechanism
 - A **representation model**
 - An infrastructure (for storage, access and queries)

Provenance Representation

- In 2006, the scientific community started an effort to understand how provenance was being represented and dealt with by different tools
- This was conducted in the form of what they called **Provenance Challenges**

See <https://openprovenance.org/provenance-challenge/WebHome.html> for details on the Provenance Challenges

1st Provenance Challenge (2006)

- The 1st provenance challenge aimed to establish an **understanding of the capabilities of available provenance-related systems** and, in particular, the following details.
 - The **representations** that systems use to document details of processes that have occurred
 - The **capabilities** of each system **in answering provenance-related queries**
 - What each system considers to be within scope of the topic of provenance (regardless of whether the system can yet achieve all problems in that scope)

2nd Provenance Challenge (2007)

- The first challenge showed that there are **multiple levels of granularity/types of provenance** that may be relevant at different times or in different parts of a process
- Thus, the focus of the 2nd Challenge was **understanding interoperability**

Open Provenance Model

- The result of the first two Provenance Challenges was the **Open Provenance Model** (OPM)

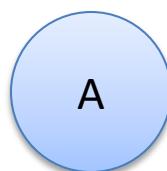
OPM

- OPM represents **retrospective provenance**
- It **never** represents things that will happen in the future (prospective provenance)

OPM Basic Concepts

- Provenance is represented by a graph
- Entities

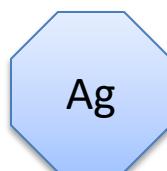
– **Artifact**



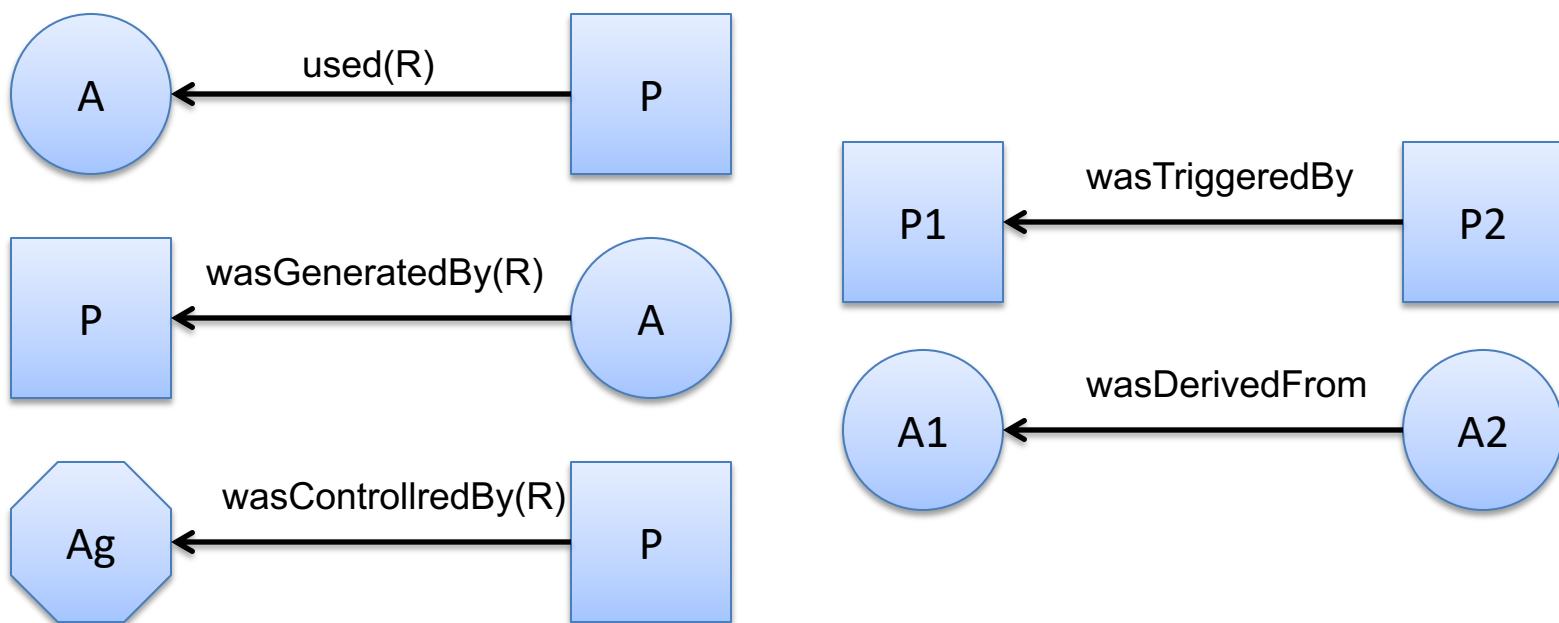
– **Process**



– **Agent**



OPM Causality Relations



3rd Provenance Challenge (2009)

- The main goal of the 3rd Provenance Challenge was to **evaluate OPM**, identifying weakness and strengths
- The result of this challenge was a set of proposed changes to the OPM specification

4th Provenance Challenge

- Interrupted due to the launch of **a new provenance model – PROV**

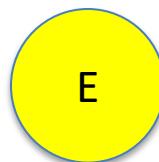
PROV

- PROV was developed by the W3C provenance incubator group
- PROV is a data model for **provenance interchange on the Web**
- PROV is a specification to express provenance records, which contain **descriptions of the entities and activities involved in producing and delivering** or otherwise **influencing** a given object

PROV Basic Concepts

- Provenance is represented as a graph
- Entities

– *Entity*



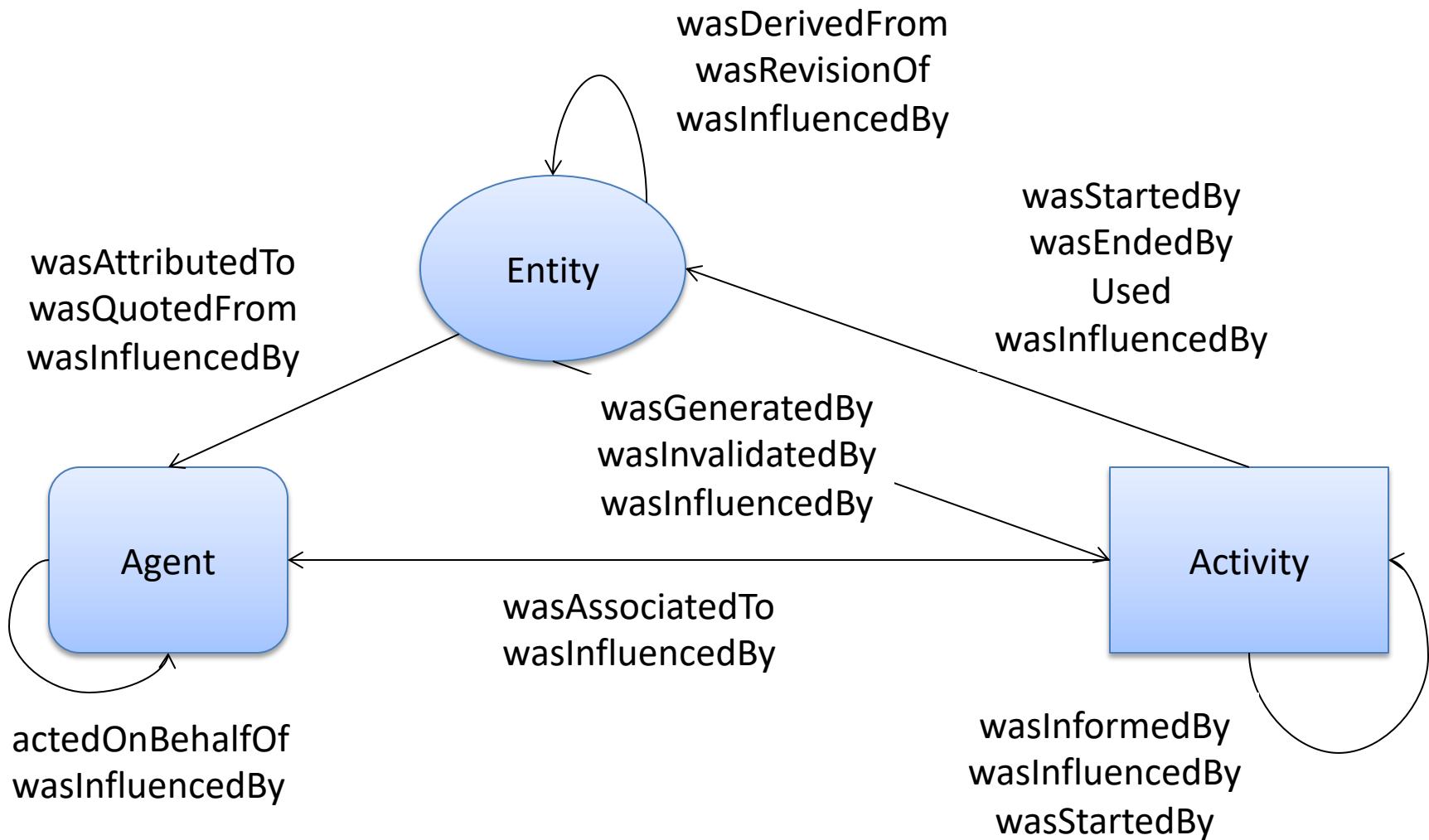
– *Activity*



– *Agent*

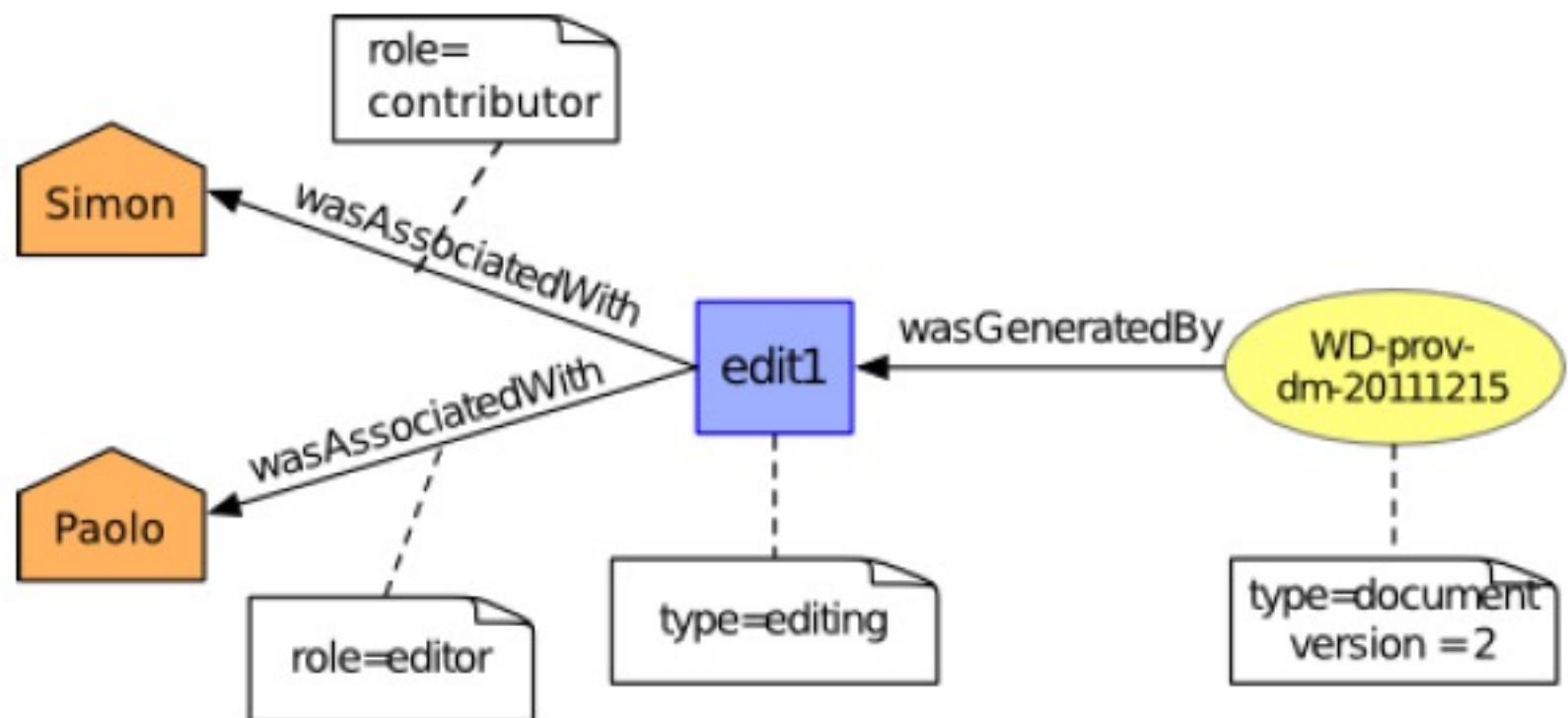


PROV Causality Relations



Example 1

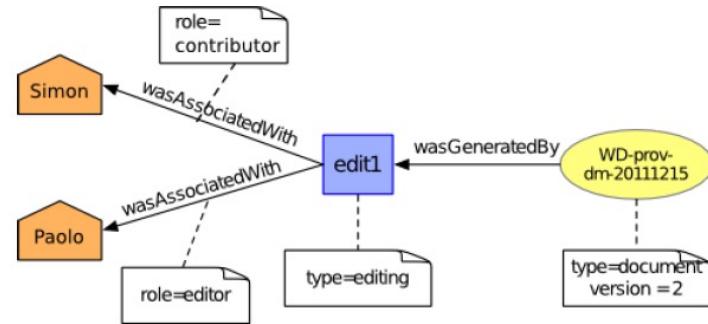
A document is edited by some editor, using contributions from various contributors.



PROV-N

- Provenance in PROV can be represented textually as well
- The representation resembles Datalog facts

PROV-N



```
entity(tr:WD-prov-dm-20111215, [ prov:type="document" ,  
ex:version="2" ])  
  
activity(ex:edit1, [ prov:type="editing" ])  
  
wasGeneratedBy(tr:WD-prov-dm-20111215, ex:edit1, -)  
  
agent(ex:Paolo, [ prov:type='prov:Person' ])  
agent(ex:Simon, [ prov:type='prov:Person' ])  
  
wasAssociatedWith(ex:edit1, ex:Paolo, -, [  
prov:role="editor" ])  
wasAssociatedWith(ex:edit1, ex:Simon, -, [  
prov:role="contributor" ])
```

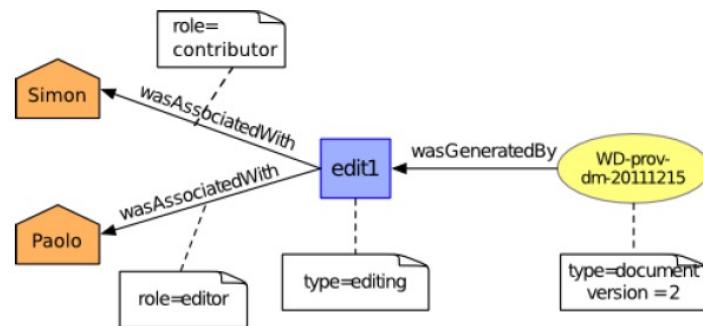
Entity

- [3] entityExpression ::= "entity" "(" identifier optionalAttributePairs ")"
- [4] optionalAttributePairs ::= ("," "[" attributePairs "]")?
- [5] attributePairs ::= (| attributeValuePair ("," attributeValuePair)*)
- [6] attributeValuePair ::= attribute "=" literal

Entity	Non-Terminal
id	Identifier
attributes	optionalAttributePairs

Entity

- [3] entityExpression ::= "entity" "(" identifier optionalAttributeValuePairs ")"
- [4] optionalAttributeValuePairs ::= ("," "[" attributeValuePair "]")?
- [5] attributeValuePair ::= (| attributeValuePair ("," attributeValuePair)*)
- [6] attributeValuePair ::= attribute "=" literal



```
entity(tr:WD-prov-dm-20111215, [ prov:type="document" ,  
ex:version="2" ])
```

Activity

[7] activityExpression ::= "activity" "(" identifier ("," timeOrMarker "," timeOrMarker)? optionalAttributeValuePairs ")"

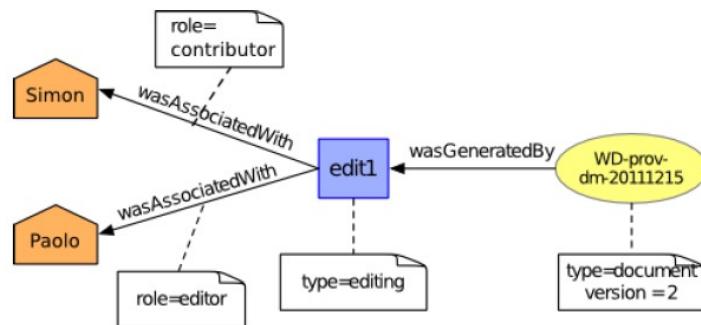
[8] timeOrMarker ::= (time | "-")

Activity	Non-Terminal
id	identifier
startTime	timeOrMarker
endTime	timeOrMarker
attributes	optionalAttributeValuePairs

Activity

[7] activityExpression ::= "activity" "(" identifier ("," timeOrMarker "," timeOrMarker)? optionalAttributeValuePairs ")"

[8] timeOrMarker ::= (time | "-")



```
activity(ex:edit1, [ prov:type="editing" ])
```

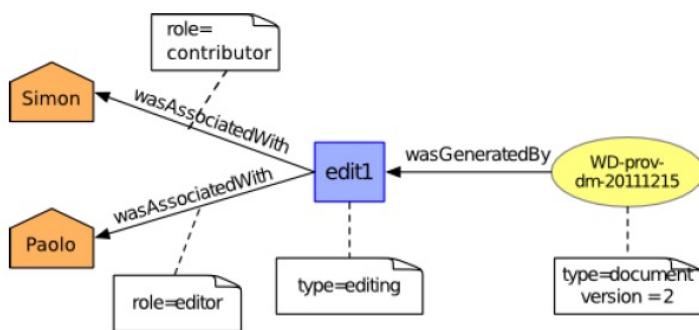
Agent

[18] agentExpression ::= "agent" "(" identifier optionalAttributeValuePairs ")"

Agent	Non-Terminal
id	identifier
attributes	optionalAttributeValuePairs

Agent

[18] agentExpression ::= "agent" "(" identifier optionalAttributeValuePairs ")"



```
agent(ex:Paolo, [ prov:type='prov:Person' ])  
agent(ex:Simon, [ prov:type='prov:Person' ])
```

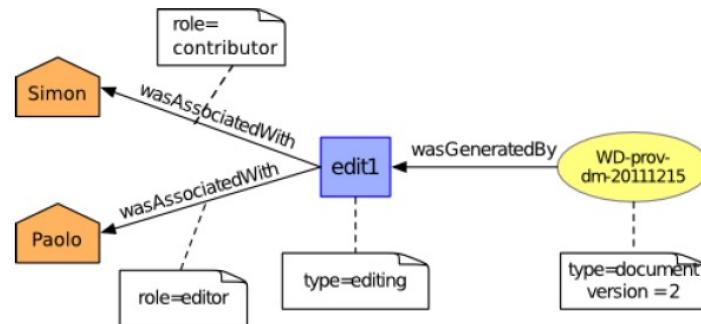
Generation

- [9] generationExpression ::= "wasGeneratedBy" "(" optionalIdentifier eldentifier ("," alidentifierOrMarker "," timeOrMarker)? optionalAttributeValuePairs ")"
- [10] optionalIdentifier ::= (identifierOrMarker ";")?
- [11] identifierOrMarker ::= (identifier | "-")

Generation	Non-Terminal
id	optionalIdentifier
entity	elIdentifier
activity	alIdentifierOrMarker
time	timeOrMarker
attributes	optionalAttributeValuePairs

Generation

- [9] generationExpression ::= "wasGeneratedBy" "(" optionalIdentifier elIdentifier ("," alIdentifierOrMarker "," timeOrMarker)? optionalAttributeValuePairs ")"
- [10] optionalIdentifier ::= (identifierOrMarker ";")?
- [11] identifierOrMarker ::= (identifier | "-")



wasGeneratedBy (tr:WD-prov-dm-20111215, ex:edit1, -)

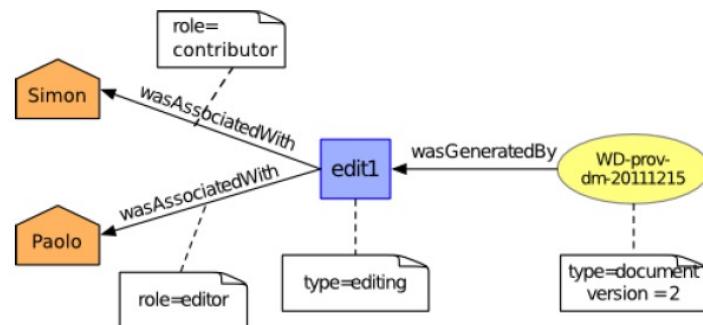
Association

```
[20] associationExpression ::= "wasAssociatedWith" "(" optionalIdentifier  
alidentifier ( "," agIdentifierOrMarker "," elIdentifierOrMarker )?  
optionalAttributeValuePairs ")"
```

Association	Non-Terminal
id	optionalIdentifier
activity	alidentifier
agent	agIdentifierOrMarker
plan	elIdentifierOrMarker
attributes	optionalAttributeValuePairs

Association

```
[20] associationExpression ::= "wasAssociatedWith" "(" optionalIdentifier
alidentifier ( "," agIdentifierOrMarker "," elIdentifierOrMarker )?
optionalAttributeValuePairs ")"
```

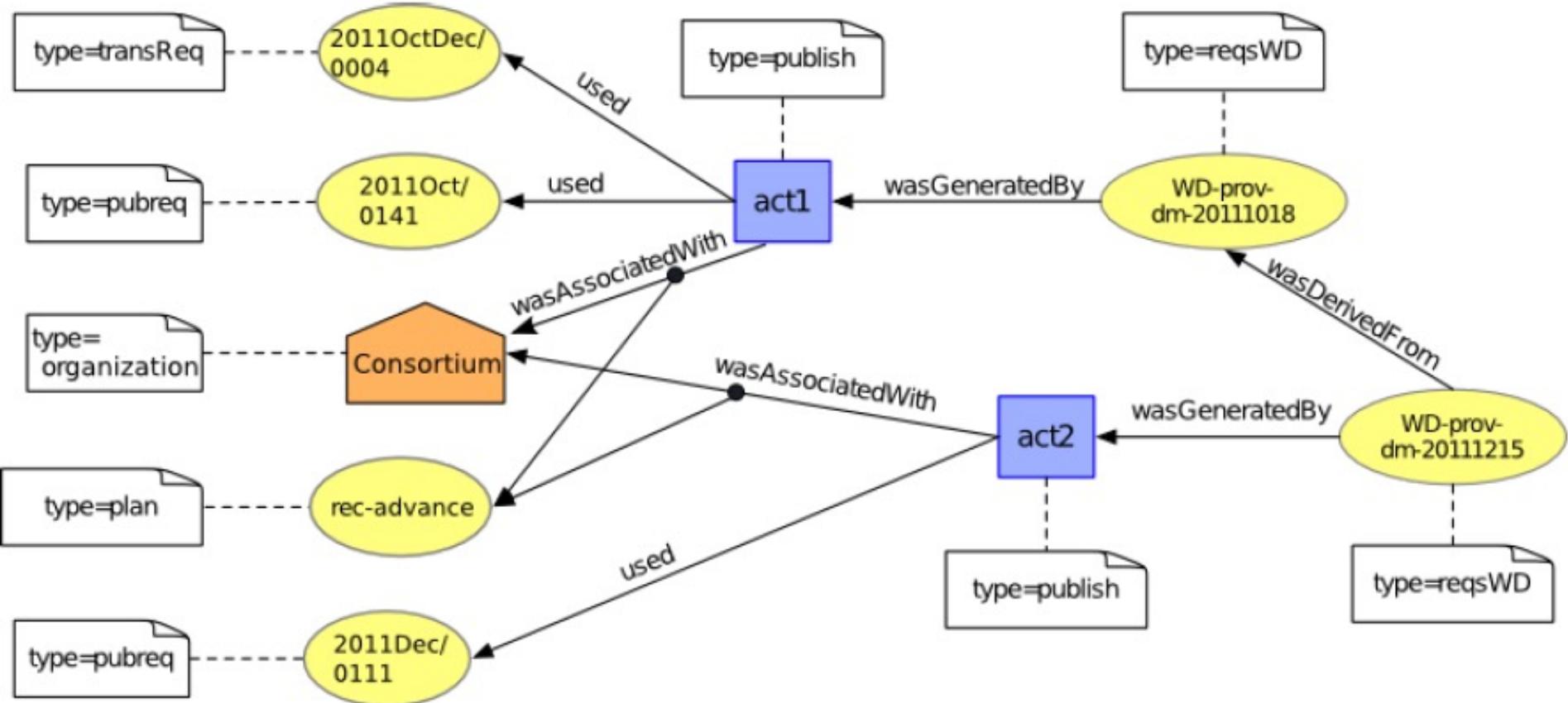


```
wasAssociatedWith(ex:edit1, ex:Paolo, - , [
prov:role="editor" ])
wasAssociatedWith(ex:edit1, ex:Simon, - , [
prov:role="contributor" ])
```

Example 2

The World Wide Web Consortium publishes documents according to its publication policy. Working drafts are published regularly to reflect the work accomplished by working groups. Every publication of a working draft must be preceded by a "publication request" to the Webmaster. The very first version of a document must also be preceded by a "transition request" to be approved by the W3C director. All working drafts are made available at a unique IRI. In this scenario, we consider two successive versions of a given document, the policy according to which they were published, and the associated requests.

Example 2 (cont.)



And much more...

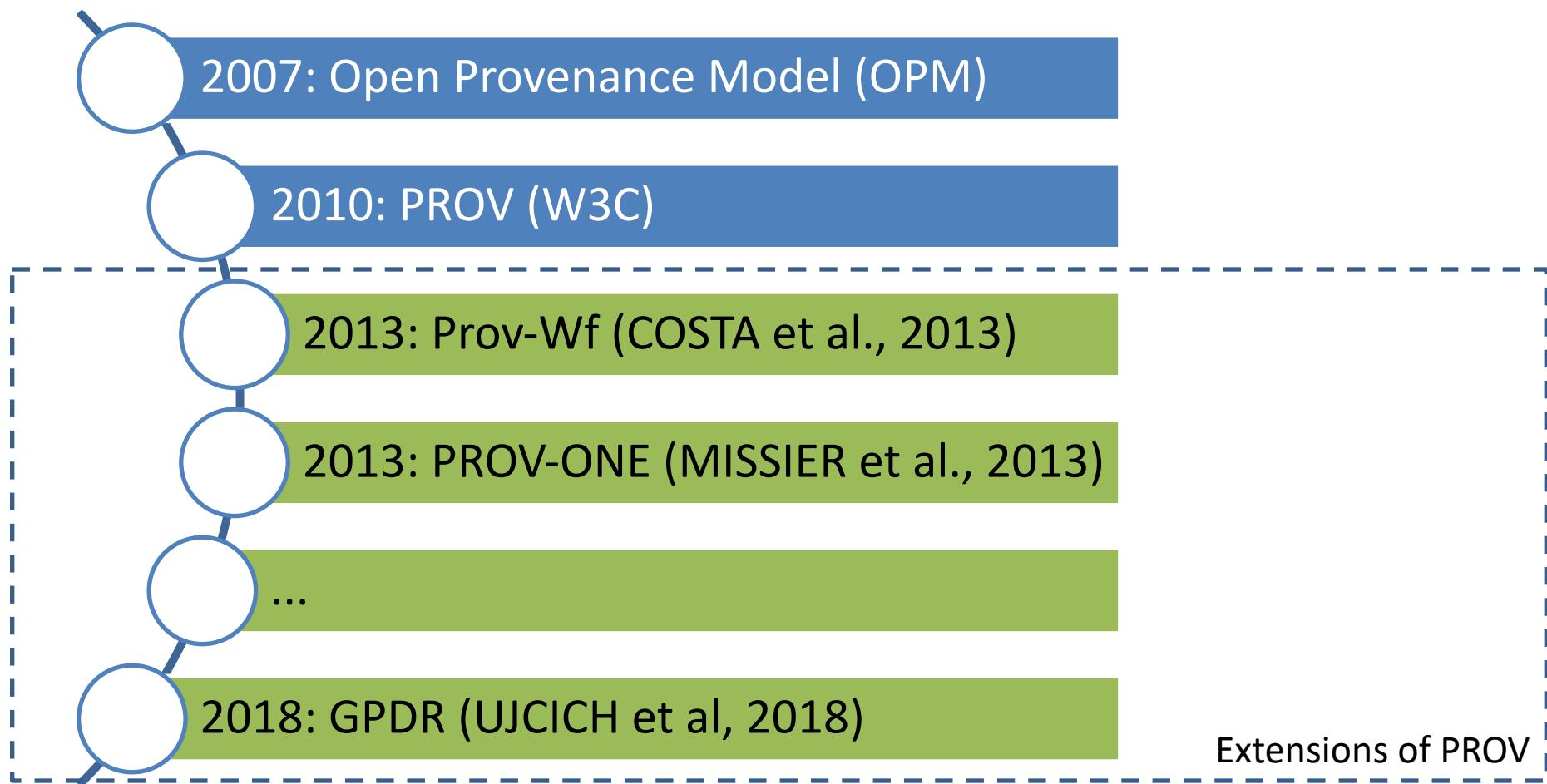
- For a complete list of PROV relationships, and how to represent them, please check the W3C website



PROV is extensible

- New properties can be added to agents, processes, entities and relationships as needed

Timeline of Provenance Models



Managing Provenance

- A Provenance Management System consists of three main components
 - A capture mechanism
 - A representation model
 - An **infrastructure** (for storage, access and queries)

Repositories

- Dataverse (<http://dataverse.org/>)
- Zenodo (<https://zenodo.org>)
- MyExperiment (<https://www.myexperiment.org/>)
- GenBank
(<https://www.ncbi.nlm.nih.gov/genbank/>)
- ...

Infrastructure

- During this course we will discuss several tools that capture, manage and/or store provenance somehow
 - VisTrails
 - Taverna
 - noWorkflow
 - YESWorkflow
 - Reprozip
 - CDE

Infrastructure

- During this course we will discuss several tools that capture, manage and/or store provenance somehow
 - VisTrails
 - Taverna
 - noWorkflow
 - YESWorkflow
 - Reprozip
 - CDE

This is **only a subset** of the available tools, but serve to give an overview of the types of tools available to capture and manage provenance

Workflow Provenance x Data Provenance

- Literature distinguishes **Workflow Provenance** (what we saw today until this point) from **Data Provenance**
- **Data Provenance** allows to track the processing of individual data items (e.g., tuples) at the “highest resolution,” i.e., the provenance itself is at the level of individual data items (and the operations they undergo).

Source: Melanie Herschel, Ralf Diestelkämper, Houssem Ben Lahmar: **A survey on provenance: What for? What form? What from?** VLDB J. 26(6): 881-906 (2017)

Data Provenance

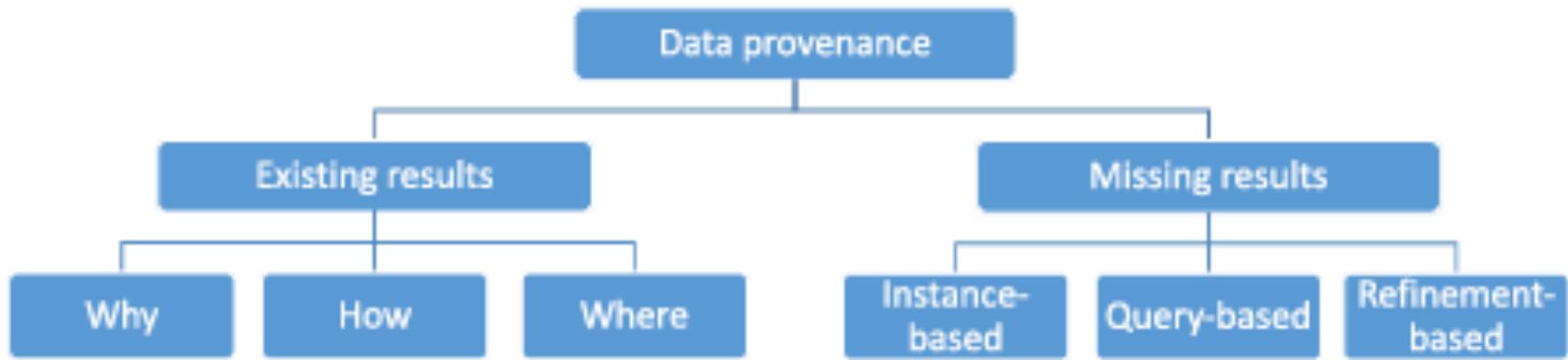


Fig. 4 Classification of data provenance research

Source: Melanie Herschel, Ralf Diestelkämper, Houssem Ben Lahmar: **A survey on provenance: What for? What form? What from?** VLDB J. 26(6): 881-906 (2017)

Types of Data Provenance

- **Why Provenance:** this tuple was in the result because some other tuple was in the input
- **How Provenance:** this tuple was formed by joining tuple t1 from R1 to tuple t2 from R2
- **Where Provenance:** this value was derived from the value x in tuple t1 from R1

Source: Peter Buneman, Wang-Chiew Tan: **Data Provenance: What next?** SIGMOD Record 47(3): 5-16 (2018)

Types of Data Provenance

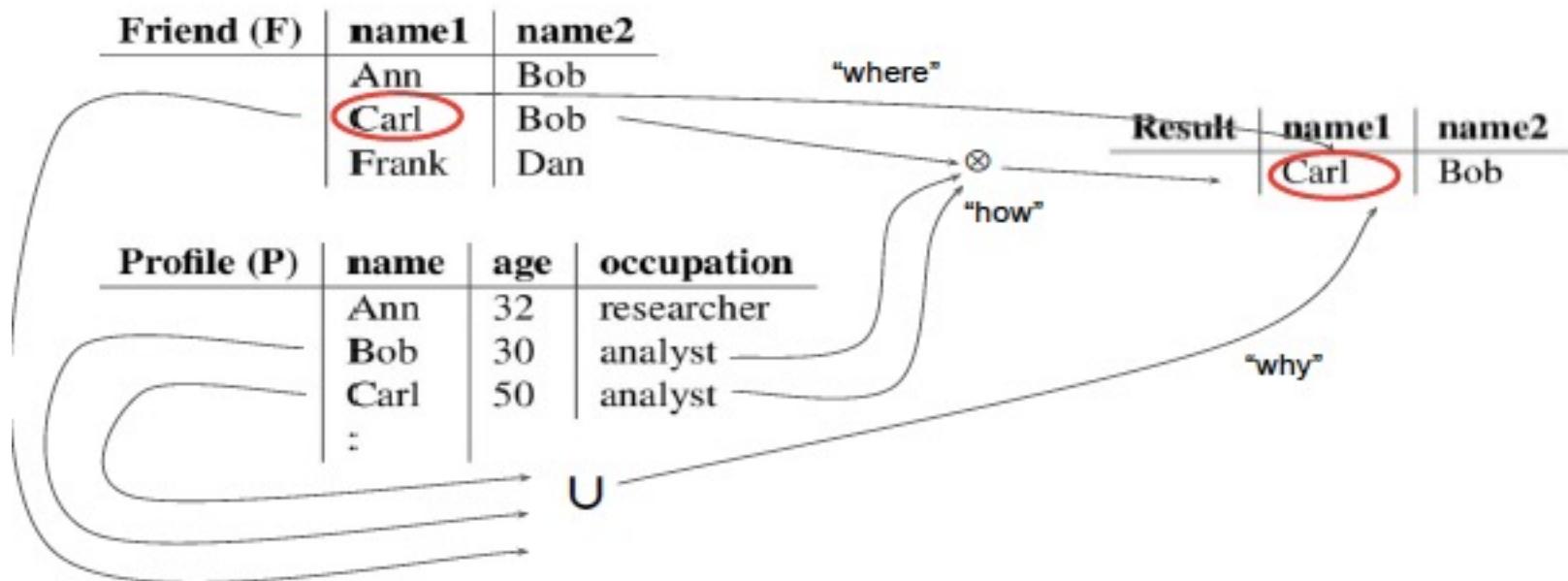


Figure 1: An illustration of where, how, and why-provenance.

Source: Peter Buneman, Wang-Chiew Tan: **Data Provenance: What next?** SIGMOD Record 47(3): 5-16 (2018)

Our focus

- In the remaining of this course, we will focus on Workflow Provenance

Exercise

- Assume that you want to re-execute an experiment that one of your students run 10 years ago. You have the source code (a Python script), the input data, and also all the results described in a paper that was published years ago. What would you need to do in order to re-execute that experiment? What could go wrong? How to prevent this from happening again in the future?

Provenance of these slides...

- Some of these slides were authored by Fernando Chirigati (NYU), João Felipe Nicolaci Pimentel (UFF), Leonardo Murta (UFF) and Troy Kohwalter (UFF)
- Benjamin E. Ujcich, Adam Bates, William H. Sanders: **A Provenance Model for the European Union General Data Protection Regulation.** IPAW 2018: 45-57
- Costa, F., Silva, V., de Oliveira, D., Ocaña, K., Ogasawara, E., Dias, J., Mattoso, M.: **Capturing and querying workflow runtime provenance with PROV: a practical approach.** In: EDBT/ICDT Workshops. (2013).
- Freire, J., Koop, D., Santos, E., Silva, C. **Provenance for Computational Tasks: A Survey.** Computing in Science & Engineering, 2008.
- Frew, J., Metzger, D., Slaughter, P. Automatic capture and reconstruction of computational provenance. Concurrency and Computation: Practice and Experience, 2007.
- Gil, Y., Miles, S., 2010. **PROV Model Primer.** W3C. URL
<http://www.w3.org/TR/prov-primer/>
- Marinho, A. ; Murta, L. ; Werner, C. ; Braganholo, V. ; Cruz, S. ; Ogasawara, E. ; Mattoso, M. **ProvManager: a provenance management system for scientific workflows.** Concurrency and Computation, v. 24, p. 1513-1530, 2012.

Provenance of these slides...

- Melanie Herschel, Ralf Diestelkämper, Houssem Ben Lahmar:
A survey on provenance: What for? What form? What from? VLDB J. 26(6): 881-906 (2017)
- Missier, P., Dey, S., Belhajjame, K., Cuevas-Vicentín, V., Ludäscher, B.: **D-PROV: Extending the PROV Provenance Model with Workflow Structure.** In: TaPP (2013).
- Moreau, L., Freire, J., Futrelle, J., McGrath, R.E., Myers, J., Paulson, P.: **The Open Provenance Model: An Overview.** In: IPA(W. (2008).
- Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E., Den Bussche, J.V., 2011. **The Open Provenance Model core specification (v1.1).** In: Future Generation Computer Systems 27, 743–756.
- Murta, L., Braganholo, V., Chirigati, F., Koop, D., Freire, J. **noWorkflow: Capturing and Analyzing Provenance of Scripts.** In: IPA(W, 2014.
- Peter Buneman, Wang-Chiew Tan: **Data Provenance: What next?** SIGMOD Record 47(3): 5-16 (2018)
- Pimentel, J. ; Dey, S. ; McPhillips, T. ; Belhajjme, K. ; Koop, D. ; Murta, L. ; Braganholo, V. ; Ludascher, B.. **Yin & Yang: Demonstrating Complementary Provenance from noWorkflow & YesWorkflow.** In: IPA(W 2016, Washington D.C. , v. 9672. p. 161-165.
- Susan B. Davidson, Juliana Freire: **Provenance and scientific workflows: challenges and opportunities.** SIGMOD Conference 2008: 1345-1350