IPAW'12

International Provenance and Annotation Workshop

Towards Integrating Workflow and Database Provenance

Fernando Chirigati and Juliana Freire

Polytechnic Institute of NYU

NYU·poly

Database Provenance

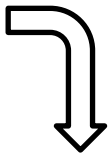
- Fine-grained provenance
- Focus: derivation of a piece of data in a dataset
  - Provenance for tuples in a relational database
  - Propagation of provenance through queries
  - "Which parts of the database D contributed to the piece of data t according to query Q?" [Cheney et al., FTDB 2009]

Employee		
Name	**DeptName**	**Salary**
And	CBE	$50,000
Chris	CSE	$60,000
Robert	CSE	$55,000
Ryan	ECE	$40,000

⋈

Department	
DeptName	**Address**
CBE	6 MetroTech
CSE	2 MetroTech

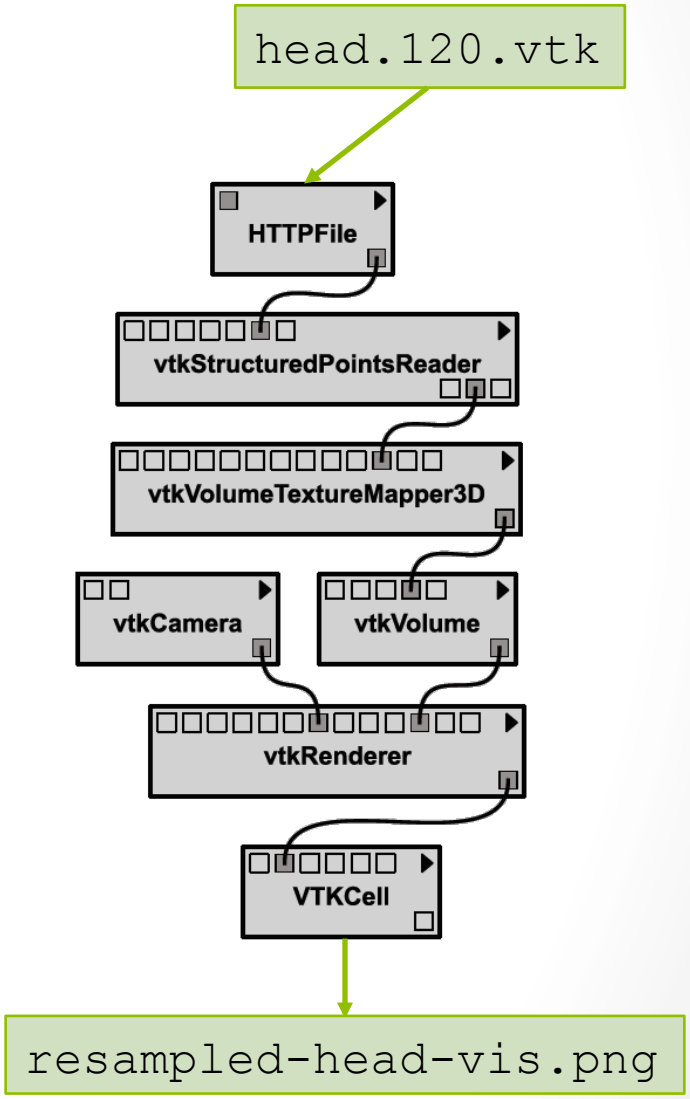
=



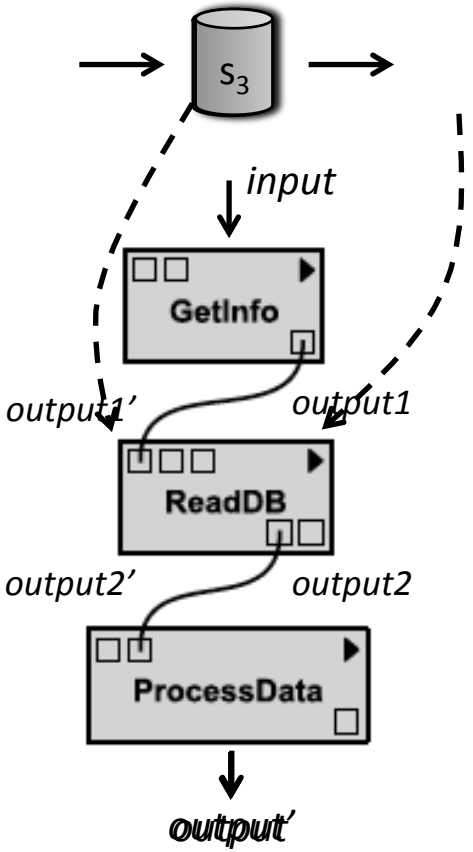
Name	DeptName	Salary	Address
And	CBE	$50,000	6 MetroTech
Chris	CSE	$60,000	2 MetroTech
Robert	CSE	$55,000	2 MetroTech

Workflow Provenance

- Coarse-grained provenance
- A directed graph describing a computational task
  - Vertices = modules = processing steps + parameters
  - Edges = connections between output and input ports
  - Execution order determined by flow of data from output to input ports
- Record of the entire history of the derivation of the output [Davidson and Freire, SIGMOD 2008]

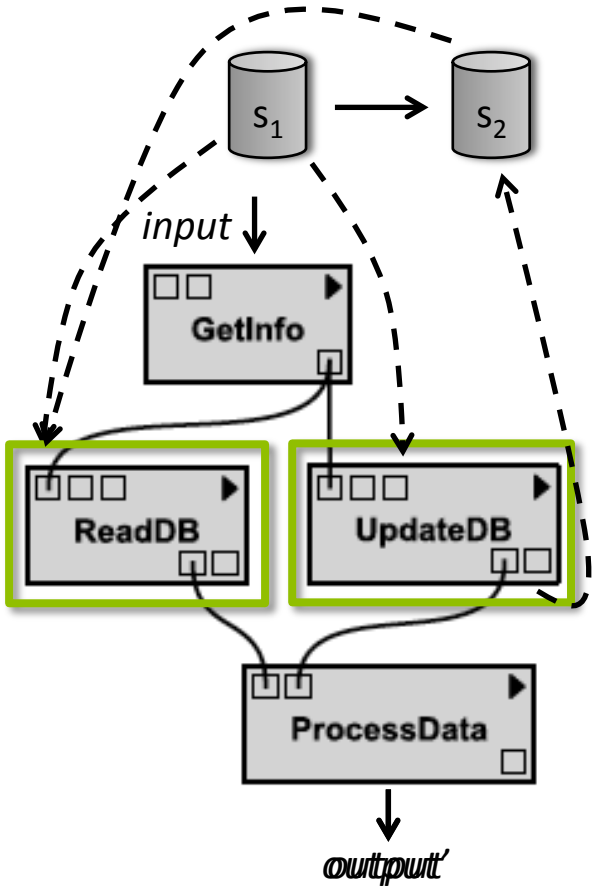


Workflows + Databases: Challenges



output (Run 1) ≠ output' (Run 2)

Workflows + Databases: Challenges



Run 2

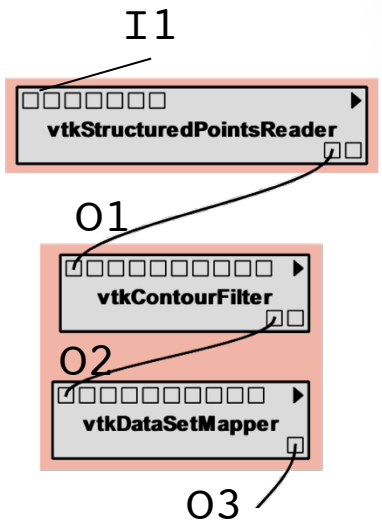
output (Run 1) ≠ output' (Run 2)

Workflows + Databases: Challenges

- *Workflows* are functional: there are no states or side effects
  - Outputs are a *function* of the inputs

```
O3 = vtkDataSetMapper(input=O2)
O2 = vtkContourFilter(value=57,input=O1)
O1 = vtkStructuredReader(input=I1)
```

- *Databases* follow a stateful model
- The models seem to be incompatible: accesses to databases ***break*** the stateless scientific workflow model
- How to support reproducibility?
- How to properly describe the provenance of workflows that access (or modify) databases?



[Davidson and Freire, SIGMOD 2008]

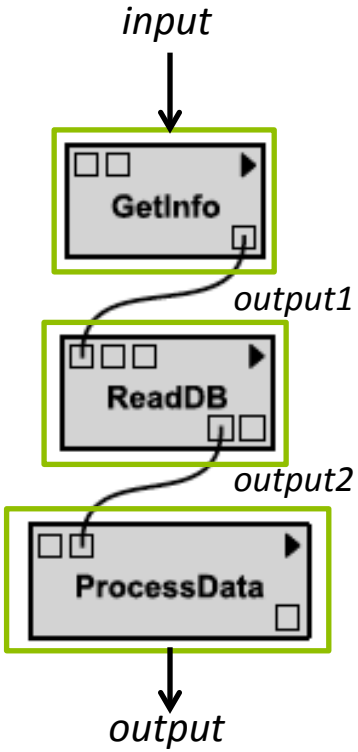
Contributions

- Model for integrating database and workflow provenance
  - Keeps track of each state of a database --- leverage transaction temporal databases [Jensen, 2000]
  - Uniquely identifies a database state using *transaction time*
  - Supports **reproducibility** and **provenance querying**
  - Reflects functionality available in commercial relational databases
    - Oracle RDBMS: *http://www.oracle.com/us/products/database/options/total-recall/overview/index.html*
    - DB2: *http://www.ibm.com/developerworks/data/bestpractices/temporal/index.html?ca=drs-*
- Proof-of-concept implementation
  - VisTrails workflow system
  - Oracle RDBMS

Background and Definitions

- ***Stateless workflows***
  - A module $m \in M$ is represented by the function

$$f_m(I_m) = < O_m >$$



$$f_{GetInfo}(input) = < output1 >$$

$$f_{ReadDB}(output1) = < output2 >$$

$$f_{ProcessData}(output2) = < output >$$

Background and Definitions

- ***Stateful databases***
  - Use the model of transaction temporal databases
  - Adapt the *backlog* schema [Jensen, JIS 1993]
    - A backlog keeps track of changes in a relation
    - It is append-only
  - Maintain a sequence of states $S(R)$ of a relation R

$$S(R) = \{(S_1(R), T_1(R)), \ldots, (S_n(R), T_n(R))\}$$

  - The difference (delta) between two states is computed as:

$$\Delta_{j,i}(R) = S_j(R) - S_i(R),\ i < j$$

Example of Backlog Relation

Emp BR_{Emp}

S_1 S_2 S_3

K	*Name*	*Job*	*T*	*Op*	*U*
1	Robert	Researcher	10	I	fchirigati
2	Claire	Assistant Director	10	I	fchirigati
1	Robert	Researcher	15	D	jfreire
1	Robert	Research Assistant	15	I	jfreire
3	Eric	Administrative Director	20	I	fchirigati

$$S(Emp) = \{(S_1(Emp),10),(S_2(Emp),15),(S_3(Emp),20)\}$$

$\Delta_{3,1}(Emp) =$

1	Robert	Researcher	15	D	jfreire
1	Robert	Research Assistant	15	I	jfreire
3	Eric	Administrative Director	20	I	fchirigati

Integrating Workflow and Database Provenance

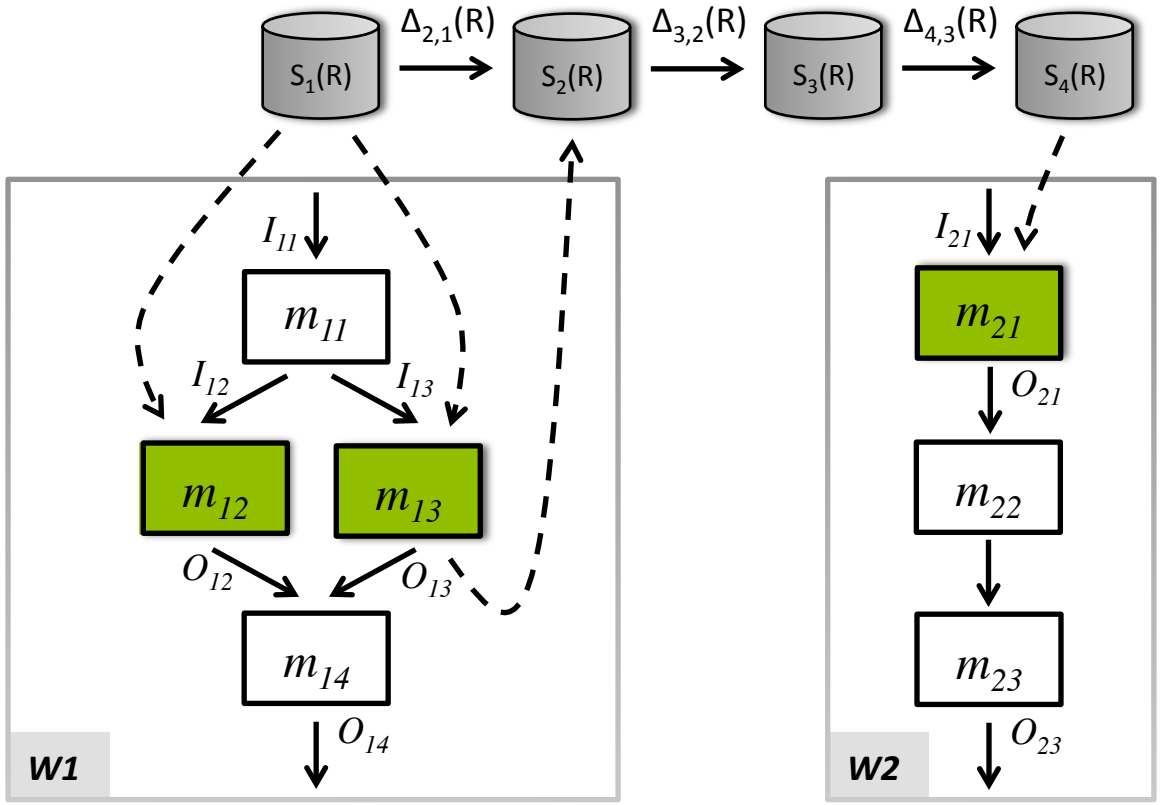
- Key idea: capture information about the database states
- Previously, we had:

$$f_m(I_m) = < O_m >$$

- Now, we have:

$$f_m(I_m, \underbrace{[R, T_b(R)]}_{\text{state before the execution}}) = < O_m, \underbrace{[R, T_a(R)]}_{\text{state after the execution}} >$$

Example of Integrated Model

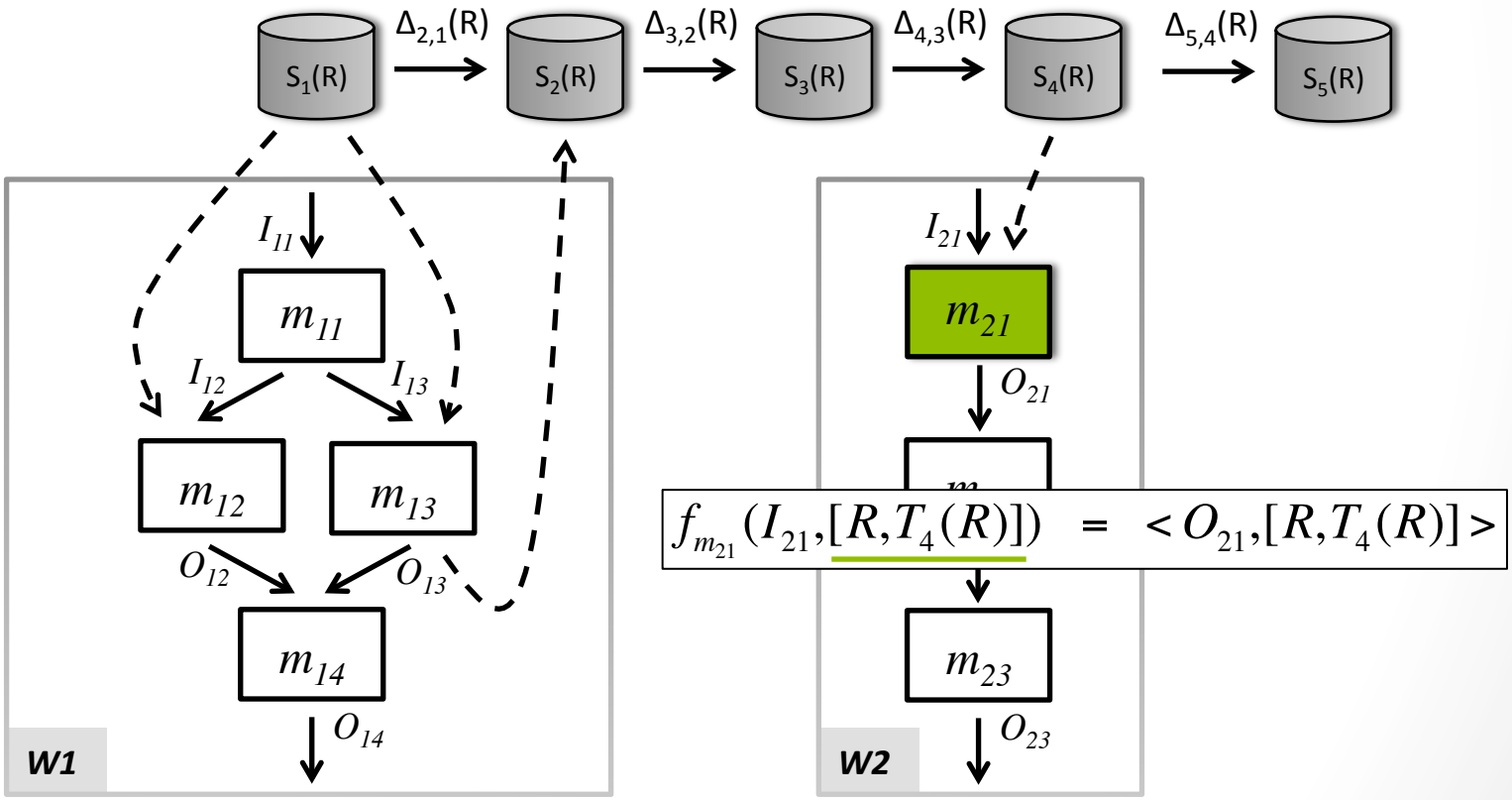


$$f_{m_{12}}(I_{12},[R,T_1(R)]) = <O_{12},[R,T_2(R)]>, f_{m_{21}}(I_{21},[R,T_4(R)]) = <O_{21},[R,T_4(R)]>$$

$$f_{m_{13}}(I_{13},[R,T_1(R)]) = <O_{13},[R,T_2(R)]>$$

Reproducibility

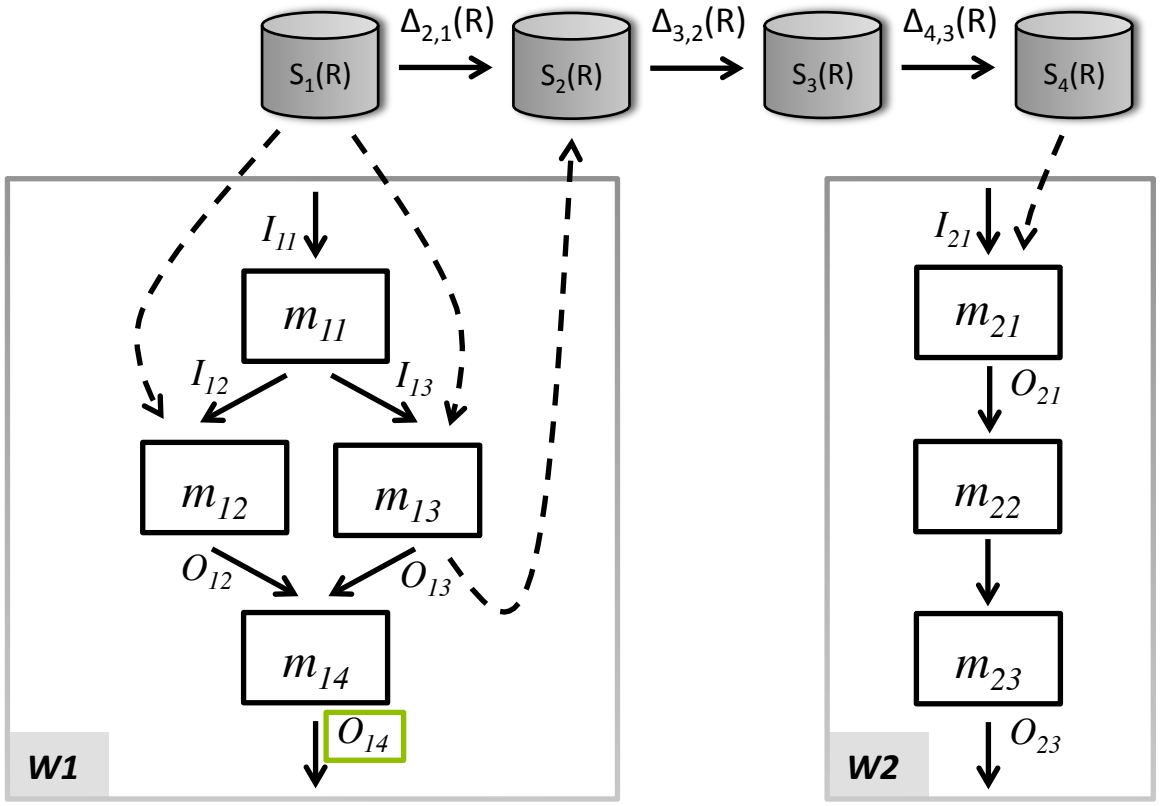
- Database states are tracked during workflow execution
- To reproduce, retrieve *original* state using recorded transaction time, if necessary



Querying Provenance

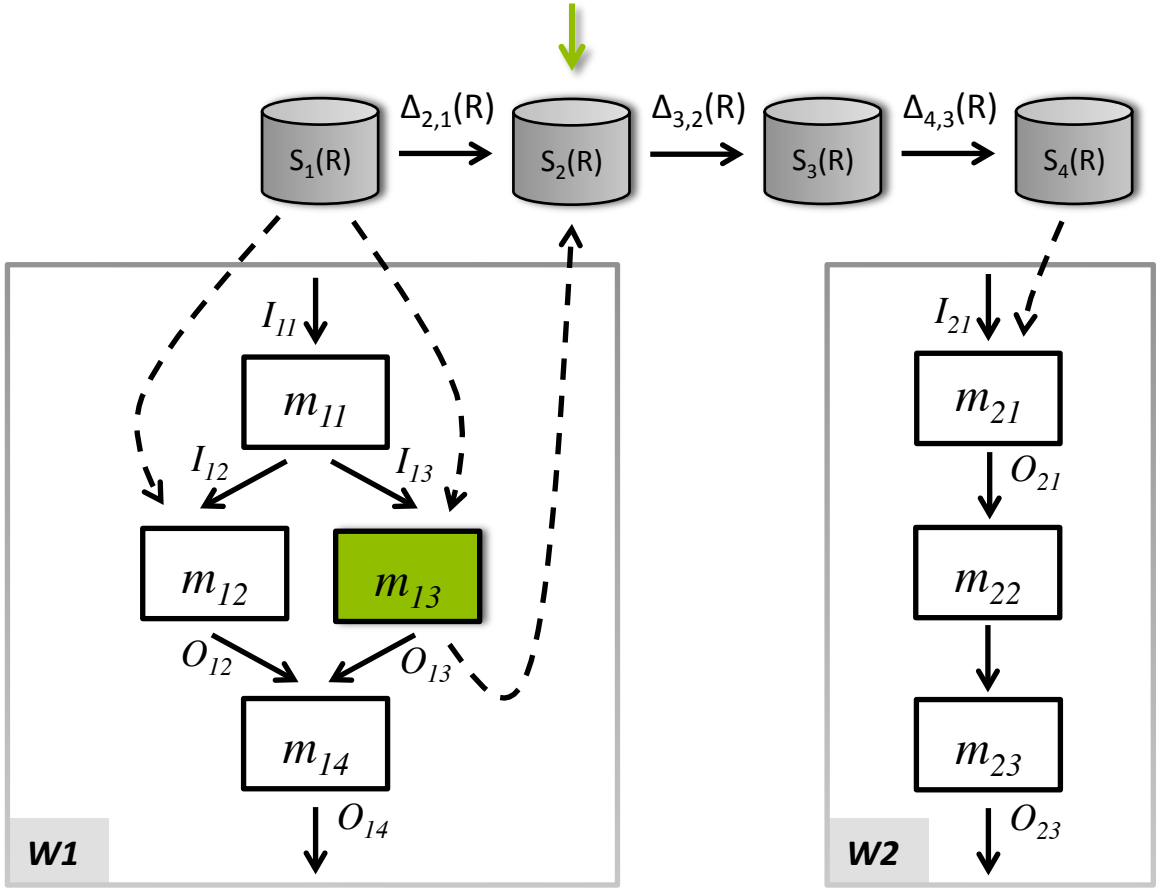
- The model supports different queries
  - Lineage of an output
  - Lineage of a database state
  - How-provenance --- *see paper for details*
  - Lineage of a tuple --- *see paper for details*

Lineage of an Output



$$lineage(O_{14}) = \{W_1, I_{11}, [R, T_1(R)]\}$$

Lineage of a State



$$f_{m_{13}}(I_{13},[R,T_1(R)]) \;=\; <O_{13},[R,T_2(R)]>$$

$$lineage(S_2(R)) = \{W_1, f_{m_{13}}, I_{13}, [R,T_1(R)]\}$$

How-Provenance

- The model has information about states S_b and S_a
- The model can retrieve the difference $\Delta_{a,b}$
- Through this difference, we know the operations performed by the module
  - We know exactly ***how*** the module modified the relation

Lineage of a Tuple

$$
\begin{aligned}
&ComputeLineage(t):\\
&\quad lineage(t) = \{\}\\
&\quad b\_times = select\ T,\ Op\ from\ R_B\ where\ K = k_t\\
&\quad for\ each\ workflow\ instance\ W:\\
&\qquad for\ each\ module\ m \in M:\\
&\qquad\quad for\ each\ T\ in\ b\_times:\\
&\qquad\qquad if\ (T\ in\ f_m\text{'}s\ output)\ and\ (T\ not\ in\ f_m\text{'}s\ input):\\
&\qquad\qquad\quad lineage(t).\ add([f_m, Op])
\end{aligned}
$$

Implementation

- VisTrails
  - Workflow-based data exploration system
  - Provides support for provenance
- Oracle RDBMS
  - Total Recall feature
  - Tracks the changes by using a *history table*
    - Similar to a backlog relation

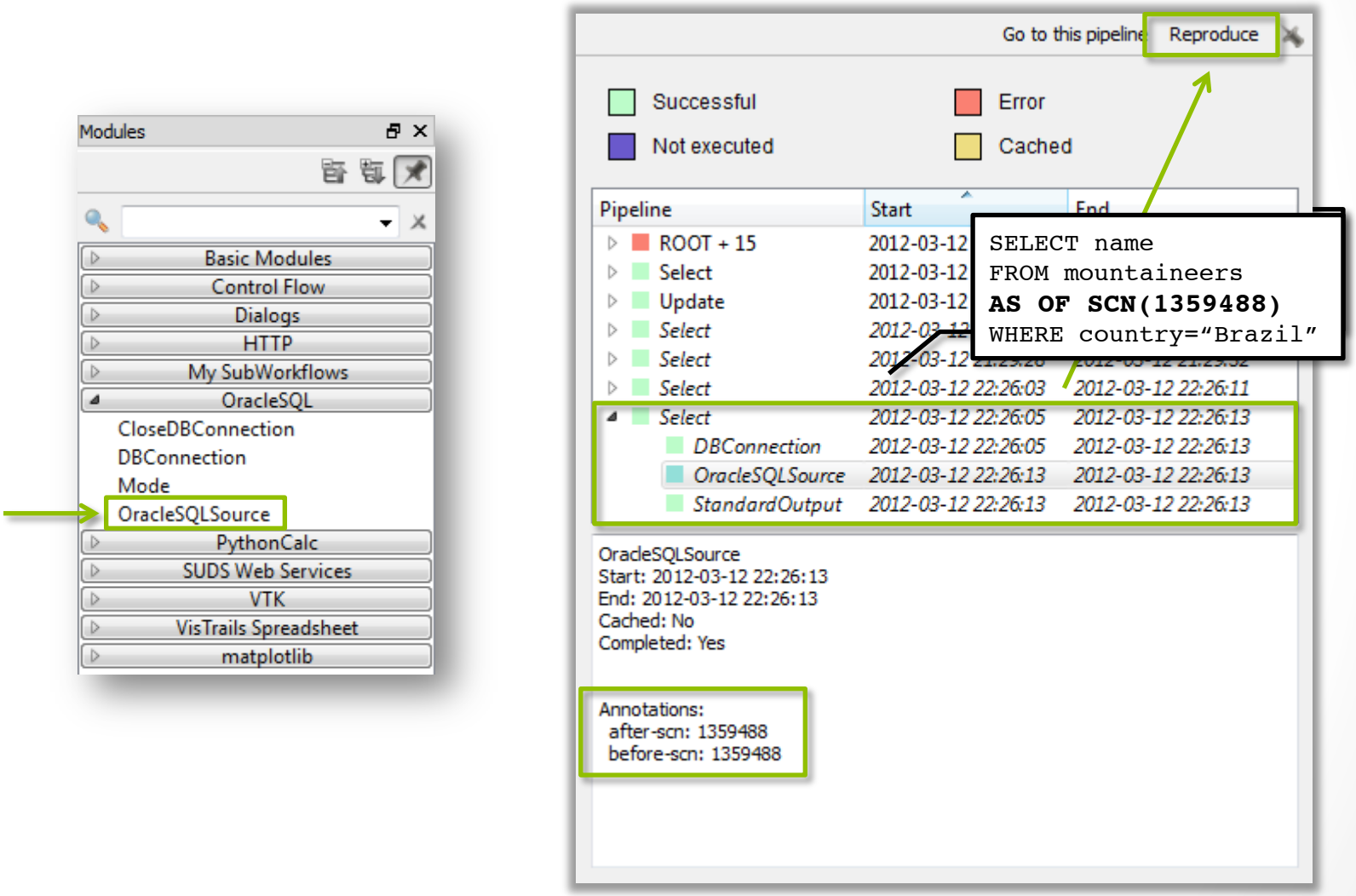

```
SELECT column_name FROM table_name
       AS OF time
```

→ S

```
SELECT column_name FROM table_name
       VERSIONS BETWEEN time_1 AND time_2
```

→ Δ

VisTrails Total Recall Package



Related Work

- [Acar et al., TaPP 2010]
  - Propose a *common* provenance graph model
  - Use DFL to support database queries and workflow steps

- [Amsterdamer et al., VLDB 2011]
  - Propose a framework to integrate fine-grained database-style provenance into workflows
  - Modules are Pig Latin programs
    - Translation to nested relational calculus expressions

Conclusion

- Proposed model to integrate workflow and database provenance
  - Leverages functionality of transaction temporal databases to capture database states and uniquely identify them
  - Supports reproducibility
  - Supports a rich set of provenance queries that straddle workflows and databases
- Future work
  - Querying -- efficiency and interfaces
  - Support DDL operations

IPAW'12

International Provenance and Annotation Workshop

Acknowledgements

Thanks to: **VisTrails Team**, **Dieter Gawlick** and **Venkatesh Radhakrishnan** (Oracle) and **Jan Van den Bussche**

This work is partially supported by the **National Science Foundation**

Any questions?

THANK YOU!

NYU·poly