UNIVERSIDAD
DE LA REPUBLICA
URUGUAY

UNIVERSITE DE VERSAILLES
SAINT-QUENTIN-EN-YVELINES
FRANCE

Data Quality Evaluation in Data Integration Systems

Verónika Peralta

AMW'2007, Punta del Este, October 26th 2007

Joint work with **Raúl Ruggia** & **Mokrane Bouzeghoub**

Agenda

- **Motivations**
- **Quality evaluation problem**
- **Quality evaluation framework**
  - Illustrated for *data freshness*
- **Conclusions**

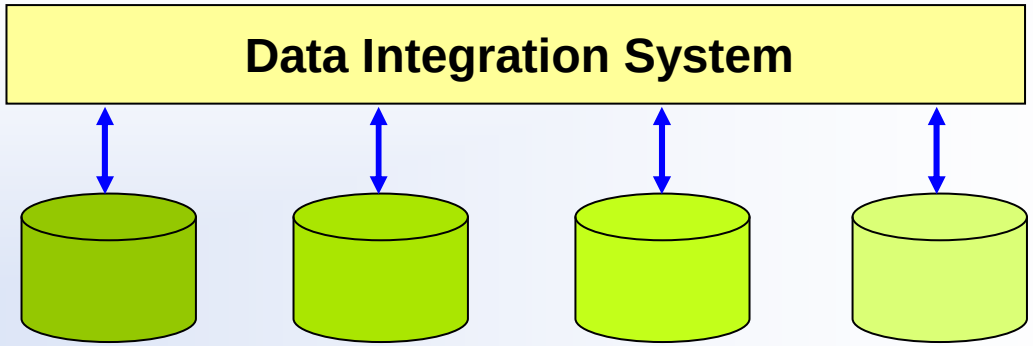
Data Quality

Context: Querying multiple, distributed, autonomous and heterogeneous data sources

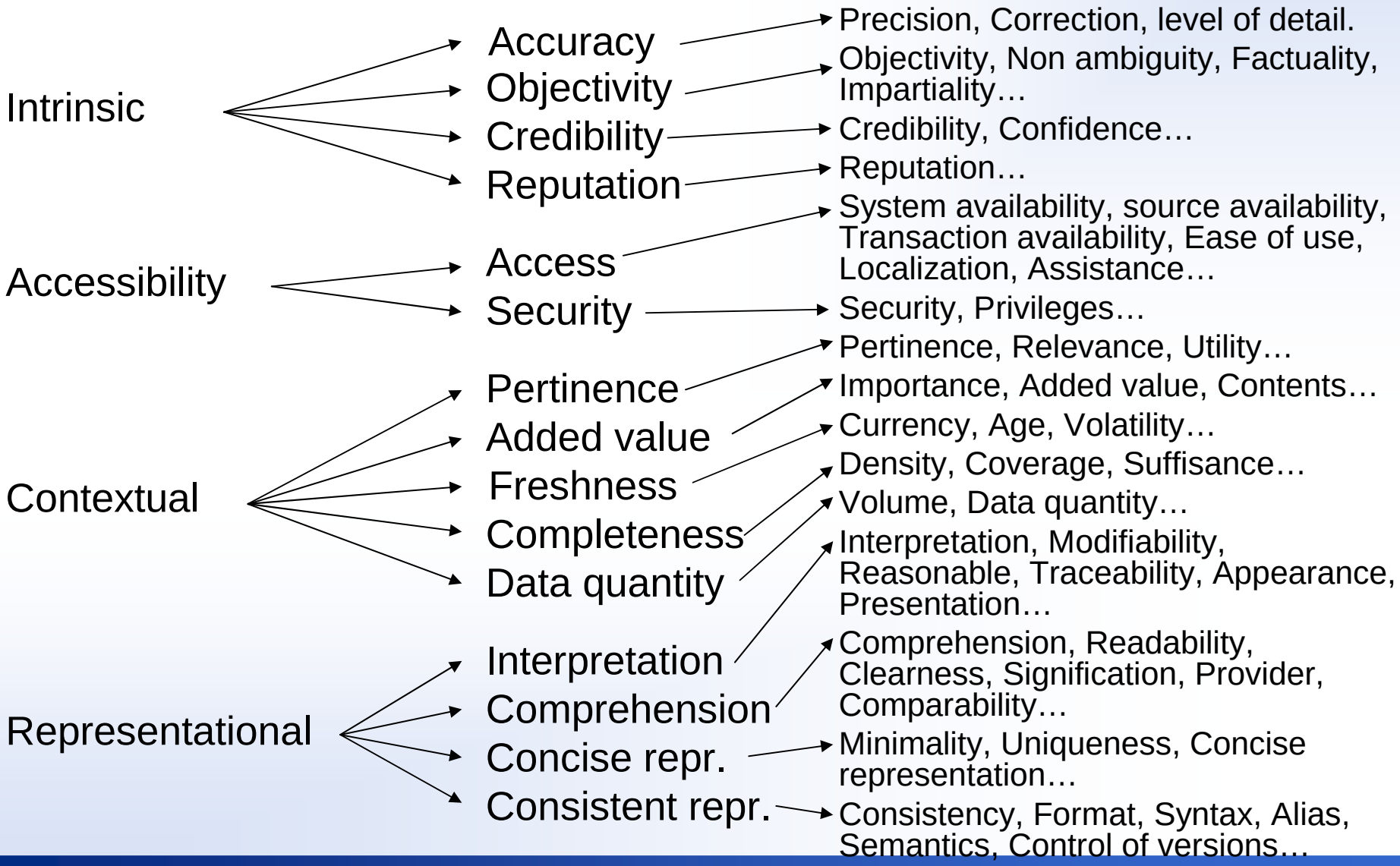
Where data comes from ?
Which is its precision degree ?
Which confidence can I have in data ?
Have I all pertinent answers ?
When was data produced ?
…

– Chic apartment, sea terrace …
– Cannasvieiras center, $5000…
– English beach, T4, garage…

Query :
Apartments to rent at Florianopolis



Multitude of quality factors



Difficulty of quality evaluation

- **Open domain:** multitude of criteria, multitude of perceptions
- **Disparate state of the art, ranging from empiric estimation methods to formal and complex evaluation models**
- **Data quality may rarely be evaluated *'de visu',***
  - Either we characterize the processes that produce data
  - Or we analyze the correlations among data
- **Quality evaluation tools are:**
  - Either embedded into IS (compilation, execution, correction)
    - Automatic decisions / actions
  - Or external to IS (observation, inspection, diagnosis)
    - Aid to the designer

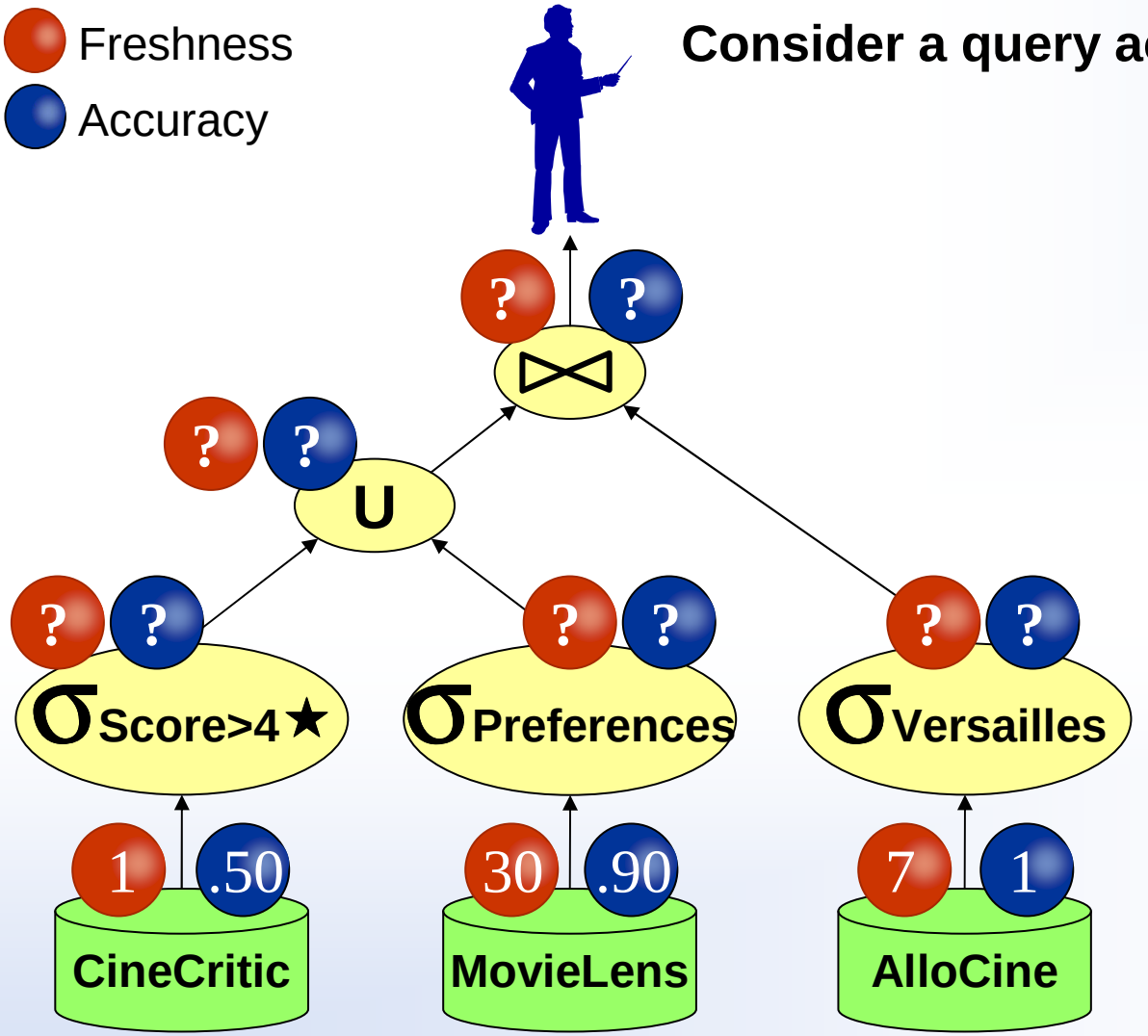
Two main approaches

- **Data-oriented approach:**
  - Data inspection (detection of anomalies)
  - Data cleaning (corrective actions)
- **Process-oriented approach:**
  - Analysis of activities and detection of critical paths
  - System improvement (evolution, maintenance)
- **Both approaches may be combined but in practice they are rarely considered together:**
  - Complexity of systems
  - Cost vs. immediate needs

Problems (query evaluation time)

Freshness

Accuracy

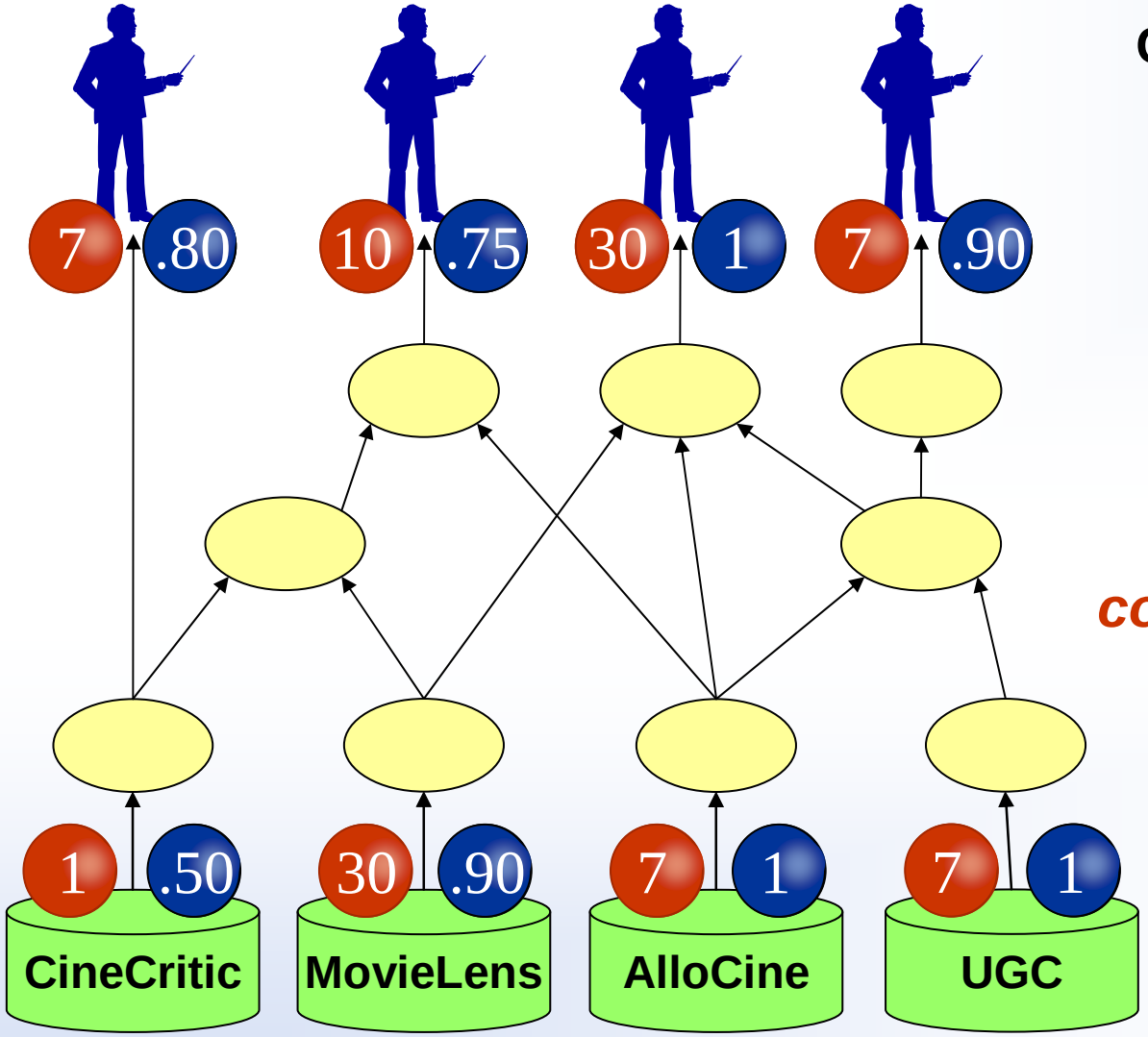


Consider a query accessing multiple sources (with their own quality)

How to calculate the quality of results?

How to calculate the quality of intermediate results?

Problems (design time)



Consider several queries (with different quality expectations)

How can we bound the quality of results?

How can we obtain constraints for sources?

Approach

- **Develop a framework for:**
  - Providing a formal base for quality evaluation
  - Analyzing quality factors and metrics
  - Identifying DIS properties that influence quality factors
  - Developing quality evaluation algorithms

- **For each quality factor:**
  - Analyze definitions, metrics, properties
  - Model DIS properties
  - Evaluate data quality
  - Analyze critical paths
  - Identify improvement actions

illustration for data freshness

Freshness

- **Several freshness definitions:**
  - **Currency**: distance extraction – delivery
    - Example: banc balance
    - Example of metric: time passed since data extraction

Extraction frequency

  - **Timeliness**: distance creation/update – delivery
    - Example: Top 10 CDs
    - Example of metric: time passed since data creation/update

Update frequency

Dimensions that influence freshness

- Several parameters influence freshness evaluation
- We classify them in 3 dimensions:
  - Nature of data
  - Architecture types
  - Synchronization policies

Dimension 1: Nature of data

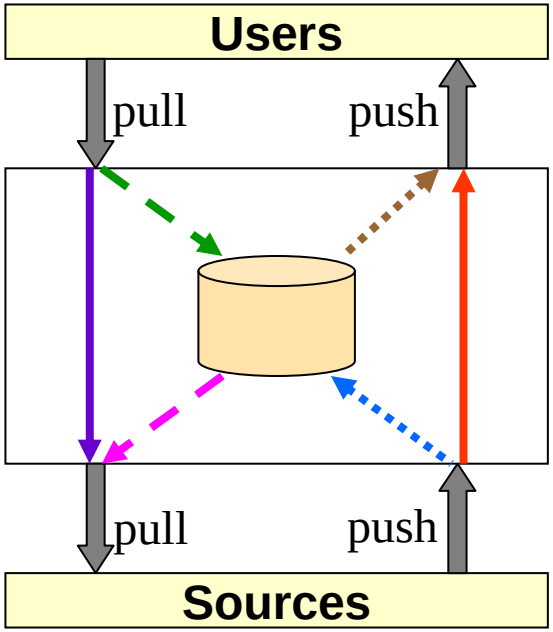
- **Data does not change with the same rhythm. The update frequency is a determinant element for freshness evaluation**
  - Stable data: ex. city names, postal codes
  - Long-term-changing data: ex. customer addresses
  - Frequently-changing data: ex. stock, real-time info
- **Sometimes, we can anticipate data freshness by correlating DB current state and data update cycle:**
  - Change events
    - Ex. Marital status (married, divorced, …)
  - Change frequency
    - Ex. Cinema billboards (every Wednesday)

Dimension 2: Architecture Types

- **DIS extraction, integration and delivery processes may introduce delays**
  - Important delays: influencing data freshness
  - Negligible delays: compared to data lifecycle
- **Data freshness also depends on replication mode:**
  - Virtual systems (execution and communication costs)
  - Caching systems (time to live)
  - Materialized systems (refreshment period)

Dimension 3: Synchronization policies

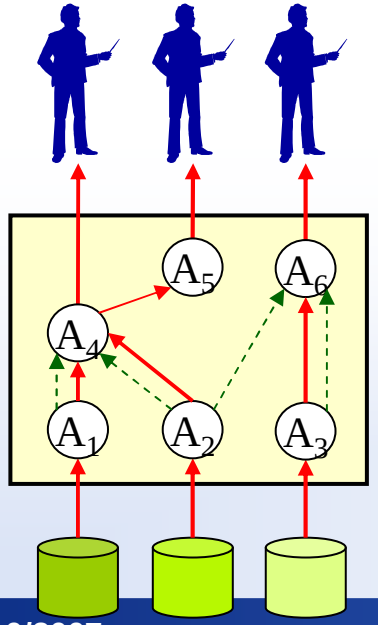
- **2 levels of synchronization:**
  - Sources ↔ DIS ↔ users (pull / push)
- **Categories :**
  - Synchronous policies:
    - Pull-pull, push-push
  - Asynchronous policies:
    - Pull/pull, pull/push, push/pull, push/push

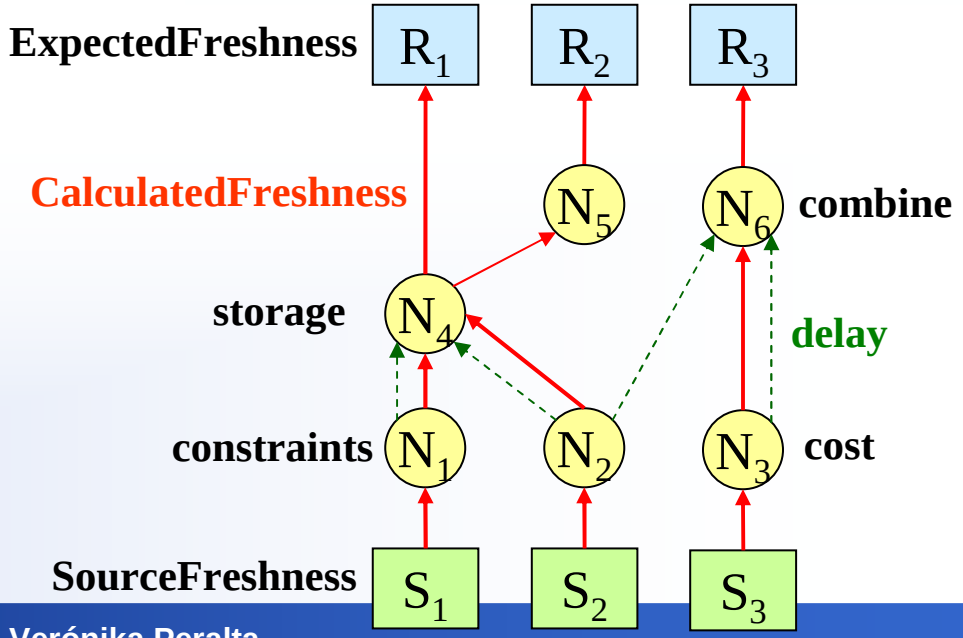


- **Asynchronous policies introduce additional delays (refreshment frequencies)**

Reasoning support

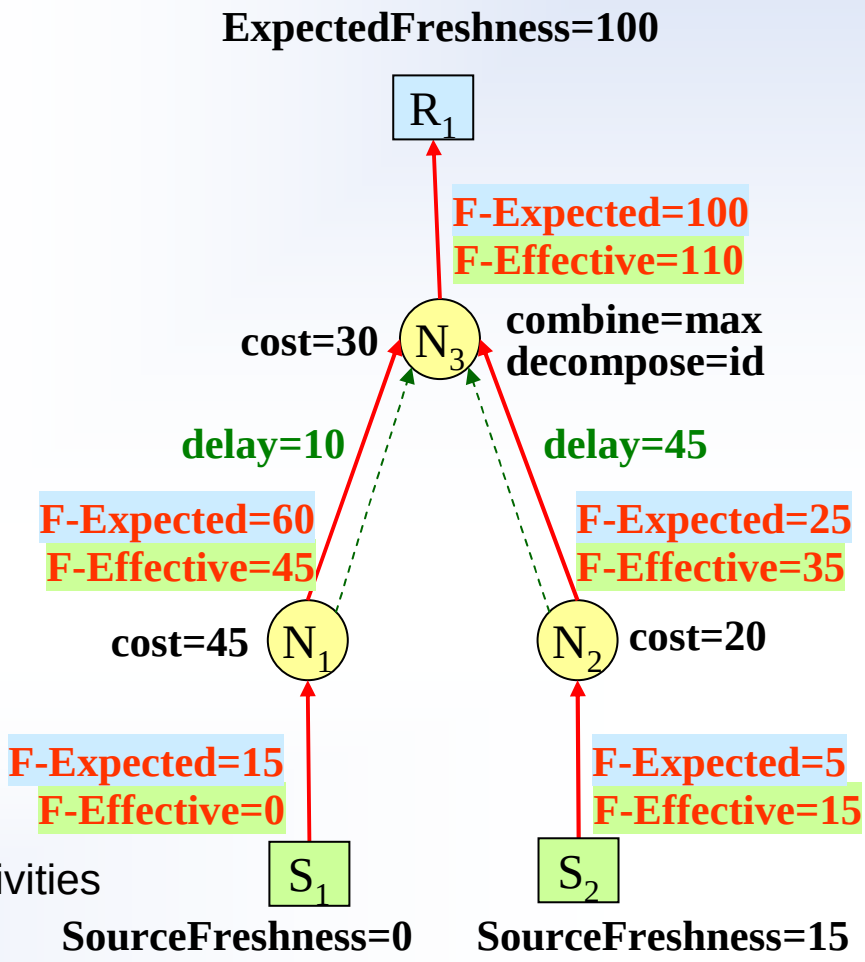
- **Freshness evaluation bases on an abstract representation of DIS processes**
  - Process graph
    - Nodes: sources, activities, queries
    - Edges: synchronization, data flow
  - Quality graph (same topology than process graph)
    - Nodes: parameters of sources, activities and queries
    - Edges: synchronization delays, freshness of data flows





Labels associated to freshness

- **Derived from the 3 dimensions or calculated**
  - Source nodes:
    - **Freshness** of **source** data
  - Query nodes:
    - **Freshness expected** by users
  - Activity nodes:
    - Execution **cost** of an activity
    - **Combine** function for several freshness values
    - **Decompose** function for freshness constraints
  - Control edges :
    - **Delay** between the execution of 2 activities
  - Data edges:
    - **Effective freshness** produced by an activity
    - **Expected freshness** for an activity

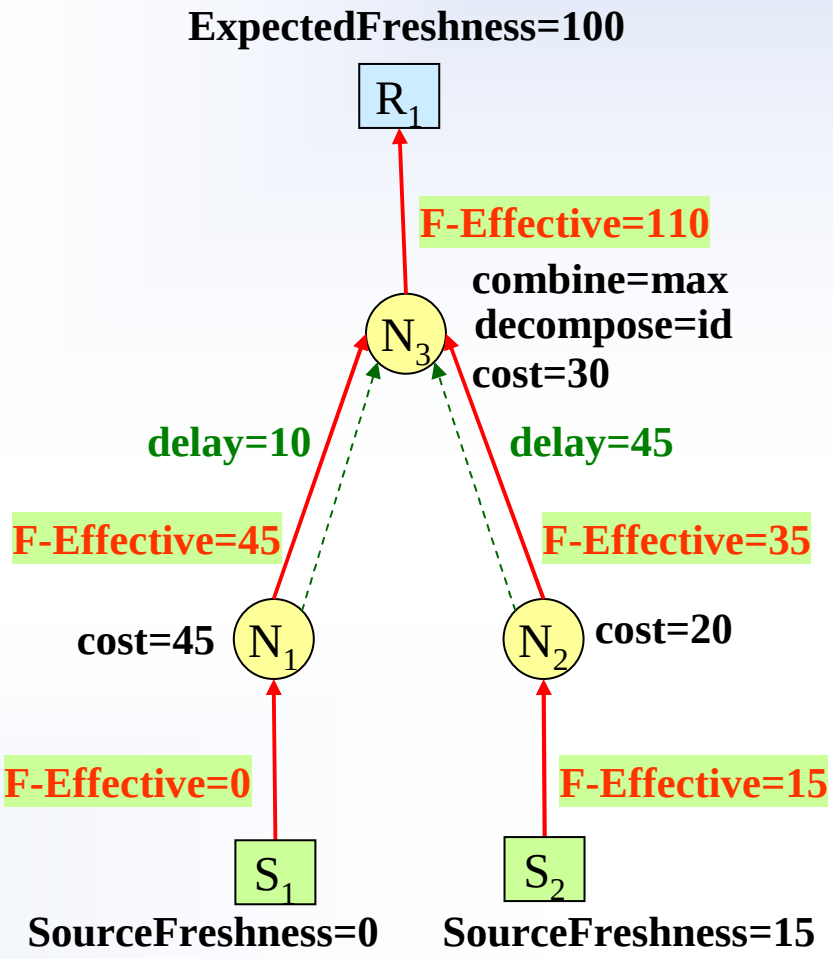


Construction of the quality graph

- **Input: process graph**
  - Identification of activities (processes)
  - Identification of sources
- **Definition of the quality graph**
  - Definition of user expectations (expected freshness)
  - Instantiation of graph properties (bounds / statistics / actual values)
  - Calculation of data freshness
    - **Calculation by forward propagation**
    - **Calculation by backward propagation**

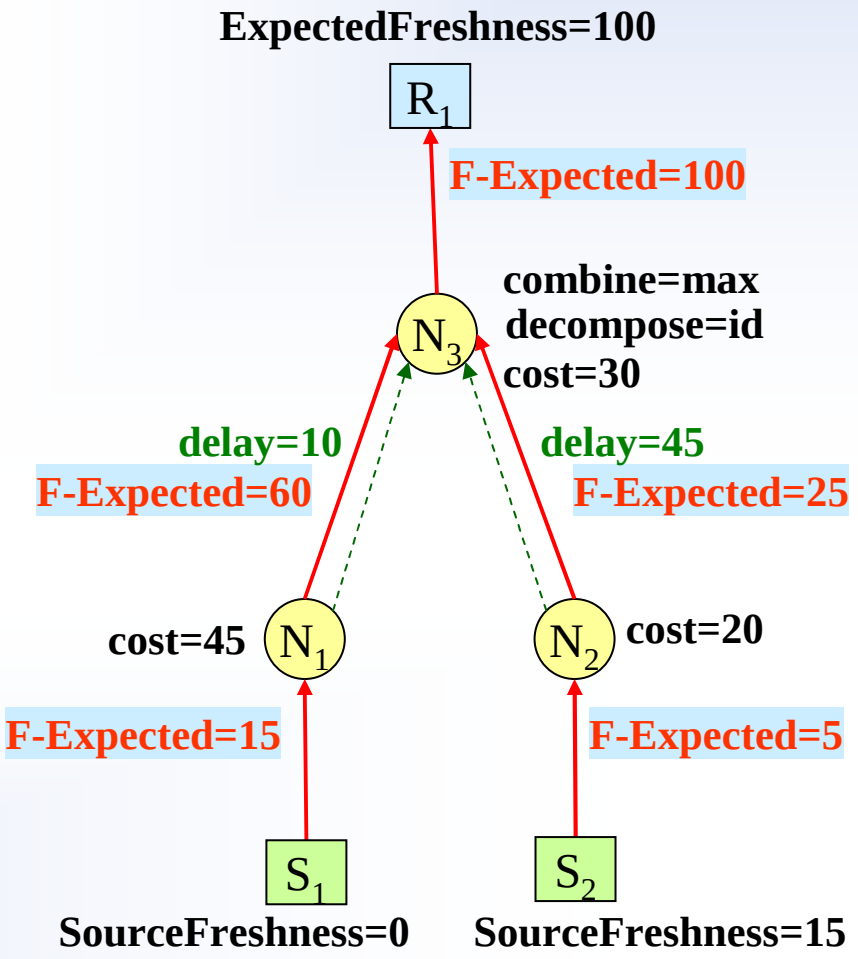
Forward propagation

- **Allows:**
  - Fixing freshness bounds
  - Verifying graph conformity for user expectations
- **Mechanism:**
  - Propagate freshness values along the graph (topological order)
  - Calculate the freshness produced by each node
    - combining properties of the node and its predecessors



Backward propagation

- **Allows:**
  - Fixing freshness constraints for sources
  - Verifying graph conformity for source freshness
- **Mechanism :**
  - Propagate freshness values along the graph (inverse topological order)
  - Calculate freshness constraints for each node
    - Decomposing freshness constraints of successor nodes



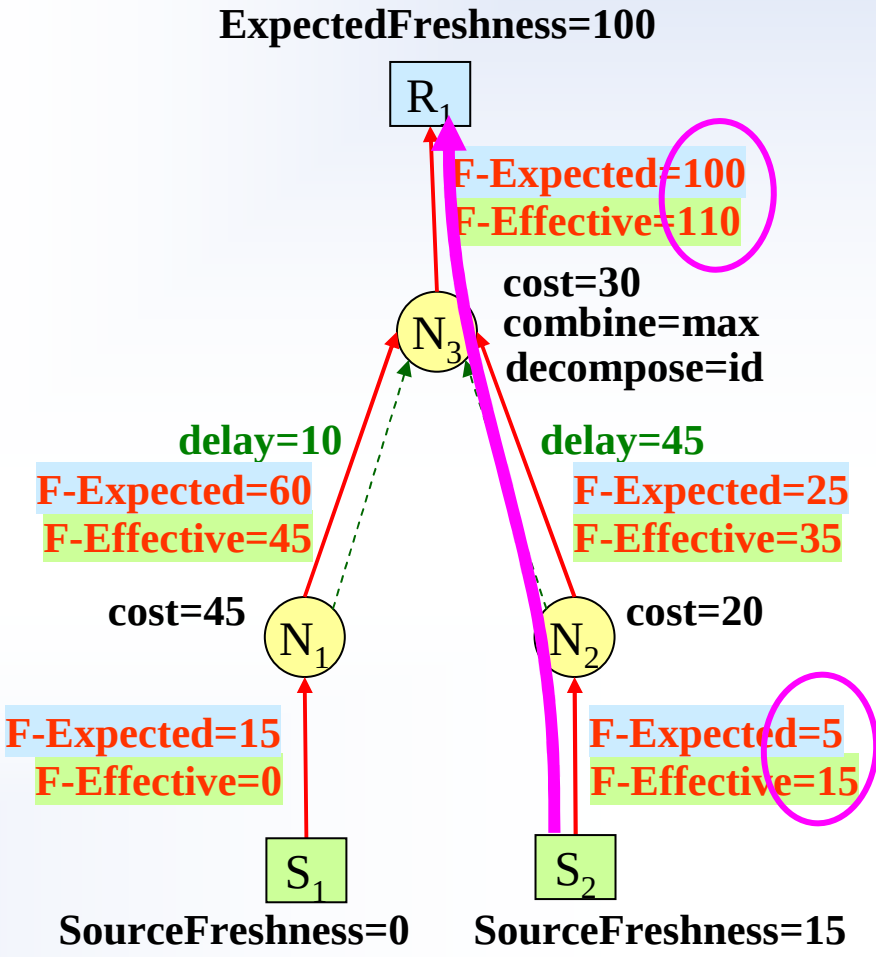
Analysis of the quality graph

- **A graph is satisfactory if**

  for every user, it satisfies his freshness expectations

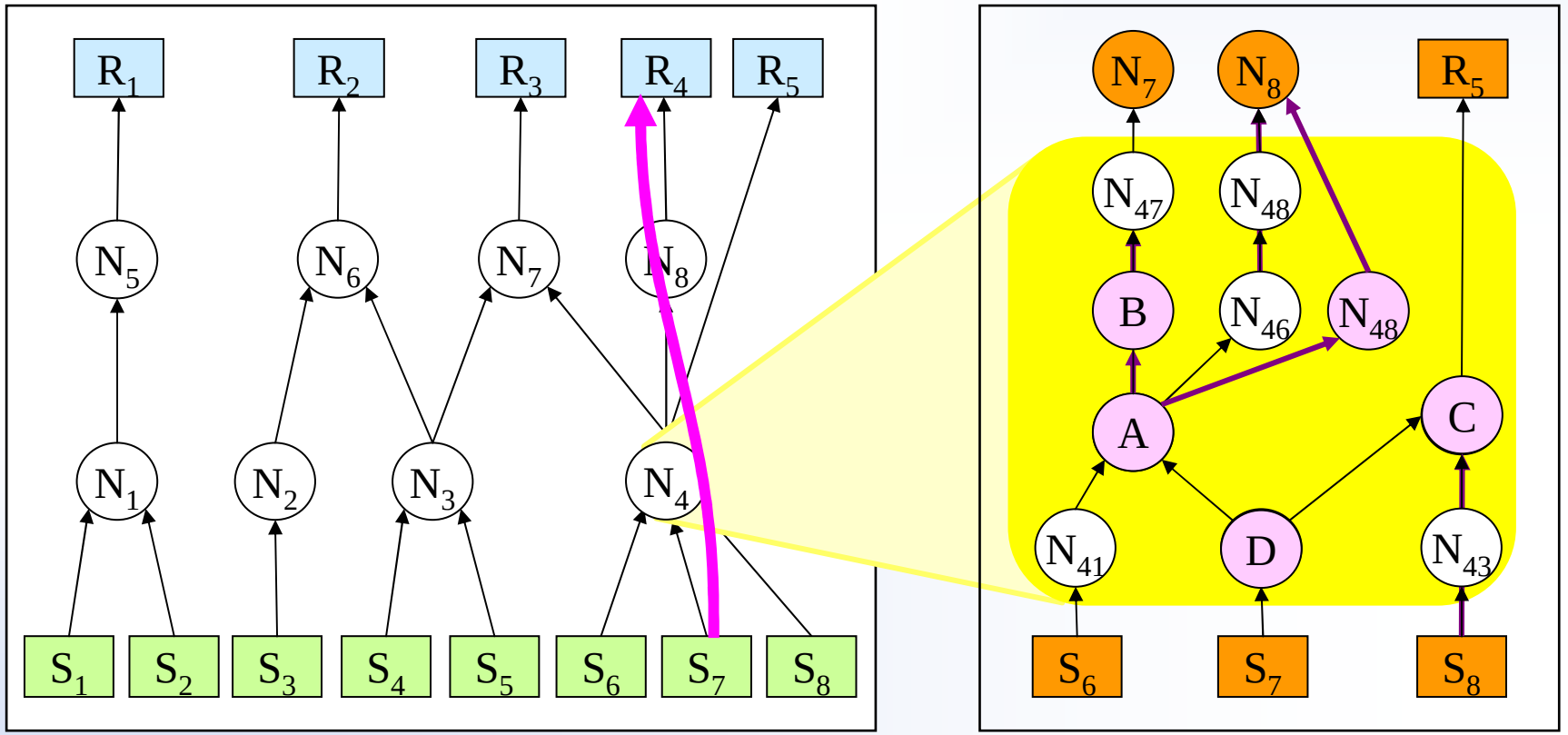
- **If an expectation is not satisfied**

  we should find the **critical paths** determining the sub-graphs to restructure



Refining and restructuring approach

Hierarchy of activities + browsing operations + restructuring operations



Browsing and restructuring operations

- **Browsing primitives:**
  - Focus+, focus–
  - Zoom+, zoom–
- **Restructuring primitives:**
  - Add node / edge / labels
  - Delete node / edge / labels
- **Restructuring macro-operations:**
  - Decompose node
  - Parallelize node
  - Fusion nodes
  - Replace node / sub-graph
  - …

  Possibility of defining new macro-operations

Summary of the proposal

- **Build a quality graph:**
  - Identify the topology (activities, sources)
  - Fix labels for each node and edge
  - Define combine functions for each node
- **Execute propagation algorithms**
- **Verify the conformity of results to user expectations / source constraints**
- **For each non-conforming result:**
  - Determine critical paths
  - Analyze critical paths
  - Propose restructuring actions

Conclusions

- **A framework for quality evaluation, analysis and improvement**
  - Analysis of quality factors
  - Quality graphs
  - Parametric evaluation algorithms
  - Analysis of critical paths
  - Improvement actions
- **The framework may be adapted to other quality factors**
  - Proposed for data freshness. Extended for data accuracy.
  - Currently studying consistency, completeness and uniqueness (Quadris project).
- **The approach was prototyped**
  - Currently using it in several application domains (Quadris project).

Thanks

PhD manuscript:

http://www.prism.uvsq.fr/~vepe/pubs/2006/phd-vp.zip

UNIVERSITE DE VERSAILLES
SAINT-QUENTIN-EN-YVELINES
FRANCE