Data Freshness Evaluation in Different Application Scenarios

Verónica Peralta – Mokrane Bouzeghoub

Laboratoire PRiSM – Université de Versailles

DKQ’2005 – Paris, France
Data freshness evaluation in a Data Integration System (DIS)

- A set of distributed autonomous data sources, possibly providing same data
- Each data source may have its own freshness
- Data sources may have access constraints
- DIS activities may range from simple query evaluation to data cleaning, transformation and aggregation
- User queries may concern one or several sources
- Users accept stale data within fixed boundaries
Motivation

♦ Freshness of results depends on:
  – Source data freshness
  – Production processes

Several Problems:
- Acquire source freshness values
- Acquire DIS property values
- Propagate (and combine) freshness values to query results
- Improve the result freshness
Agenda

♦ Data Freshness
♦ Quality Evaluation Framework
♦ Data Freshness Evaluation
  – General Approach
  – Instantiation Process
♦ Data Freshness Enforcement
♦ Conclusions
Data Freshness

Data freshness quality factors:

- **Currency**: Gap extraction – delivery
  - How stale is data with respect to sources?
- **Timeliness**: Gap creation/update – delivery
  - How old is data? Is its age appropriate?

![Diagram showing data freshness quality factors: creation time, extraction time, delivery time, currency, timeliness.](image-url)
Quality Evaluation Framework

♦ Composed of:
  – Data sources
  – Query classes
  – DIS calculation processes
  – Properties (DIS features and quality measures)
  – Algorithms (for propagating quality values)

represented as a graph
DIS representation

- DIS represented as a workflow of calculation activities (steps).

- A labeled calculation dag (LCDag) is a dag, with the same dataflow structure and properties associated to nodes and edges.
Labeled Calculation Dag (LCDag)
Properties

Two types of properties:

- **Descriptions**: Indicate DIS features
  - E.g.: costs, delays, policies, strategies, constraints
- **Measures**: Indicate freshness values
  - A source actual value acquired from a source
  - A calculated value obtained executing an evaluation algorithm
  - An expected value expressed by users
Properties Associated to Freshness

♦ Freshness of delivered data depends on:
  – Source data freshness.
  – Execution delay of the DIS.

♦ Properties associated to data freshness:
  – Processing cost: Time needed for executing an activity.
  – Synchronization delays: Delay between the execution of consecutive activities.
  – Source actual freshness: Freshness of data in a source.
Freshness Evaluation Algorithm

Algorithm principle:

* Source nodes S:
  - **Freshness (S) = ActualFreshness (S)**

* Other nodes N:
  - **Freshness (N) = Freshness (P) + S-delay (P,N) + P-Cost (N)**

* Freshness of several input nodes are combined
FUNCTION DataFreshnessEvaluation (G: LCDag) RETURNS LCDag
BEGIN
  INTEGER value;
  FOR EACH source node A DO
    value = getActualFreshness(G, A);
    G.addProperty(A, “freshness”, value);
  ENDFOR;
  FOR EACH activity and target node A in topological order DO
    HASHTABLE valList;
    FOR EACH node B in G.getPredecessors(A) DO
      value = G.getPropertyValue(B, “freshness”) + getSyncDelay(G, B, A);
      valList.add(B, value);
    ENDFOR;
    value = combine(valList) + getProcCost(G, A);
    G.addProperty(A, “freshness”, value);
  ENDFOR;
RETURN G;
END
Different algorithms for different scenarios:

- Different metrics and units
  - E.g. timeliness, currency. → different quality actual values
- Different DIS features
  - E.g. In virtual DIS there is no delay between activities execution → different cost models
- Different user quality requirements
  - E.g. When users tolerate freshness values of “weeks” activity costs of “seconds” can be omitted. → different cost models
Examples:

- A mediation system that answers queries about films, cinemas and billboard.
  
  *timeliness, no cost, no delay, priorities*

- A web portal that caches information about availability of places.
  
  *currency, cost model, refreshment delay, maximum*

- A data warehousing system that stores statistics about films and opinions.
  
  *timeliness, cost model, materialization delay, maximum*
FUNCTION DataFreshnessEvaluation (G: LCDag) RETURNS LCDag
BEGIN
  INTEGER value;
  FOR EACH source node A DO
    value = getActualFreshness(G,A);
    G.addProperty(A,"freshness",value);
  ENDFOR;
  FOR EACH activity and target node A in topological order DO
    HASHTABLE valList;
    FOR EACH node B in G.getPredecessors(A) DO
      value = G.getPropertyValue(B,"freshness") + getSyncDelay(G,B,A);
      valList.add (B, value);
    ENDFOR;
    value = combine(valList) + getProcCost(G,A);
    G.addProperty (A,"freshness",value);
  ENDFOR;
  RETURN G;
END
Data Freshness Enforcement

- Improving DIS design:
  - Reducing costs.
  - Synchronizing activities.

- Negotiate with users to relax freshness requirements

- Negotiate with source providers to relax source constraints
Improving DIS design

- **Strategies:**
  - Reduce activity costs
  - Synchronize activities to reduce delays.

- **Sometimes we can concentrate in critical paths**

- **The tool allows:**
  - Identifying critical paths
  - Changing property values and re-executing
Relaxing freshness requirements

♦ **Bottom-up strategy:**
  – Shows users the guarantee freshness

♦ **Direct application:**
  – Evaluating data freshness for several alternative implementations of the DIS
  – Comparing evaluated freshness between them.

♦ The tool was used in conjunction with a generator of mediation queries
  – Evaluating the freshness of the generated queries and selecting the best one. [BDA’2004]
Relaxing source constraints

♦ **Top-down strategy:**
  – Shows the freshness needs for each source

♦ **Direct application:**
  – Comparing alternative data sources.

♦ **The tool allows:**
  – Bottom-up and top-down propagation
Conclusion

♦ A framework for data freshness evaluation
  – General evaluation approach.
  – Instantiation mechanism.

♦ Prototype
  – Implements the framework components.
  – Supports instantiation.
  – Supports bottom-up and top-down propagation.
  – Visualization facilities (e.g. critical path).

♦ Future works:
  – Automating the instantiation process.
  – Confront evaluated values with user expectations (profiles).
  – Improve the tool: scalability, interfaces.