Formal Specification and Optimization of ETL Scenarios

Timos Sellis
National Technical University of Athens
(joint work with Alkis Simitsis and Panos Vassiliadis)
Motivation

- ETL and Data Cleaning tools cost – the standard figures
  - 30% of effort and expenses in the budget of the DW
  - 55% of the total costs of DW runtime
  - 80% of the development time in a DW project
- ETL market: a multi-million market
- ETL tools will not be replaced by other tools in near future
- ETL tools in the market
  - software packages
  - in-house development
- No standard, no common model
Problems

- The key factors underlying the main problems of ETL processes are:
  - **vastness** of the data volumes
  - **quality problems**, since data is not always clean and has to be cleansed
  - **performance**, since the whole process has to take place within a specific time window
  - **evolution** of the sources and the data warehouse can eventually lead, even to daily maintenance operations
But is it a “hot” problem?

- Some practitioners find that it may be obsolete

- Very few papers from academia
  - Even DOLAP has 2-3 papers in its history
  - Some interesting statistics!

- Do you agree?
Extract-Transform-Load (ETL)
Modeling Work – Why?

☐ Conceptual
  ■ we need a simple model, sufficient for the early stages of the data warehouse design; we need to be able to model what our sources “talk” about

☐ Logical
  ■ we need to model a workflow that offers formal and semantically founded concepts to capture the characteristics of an ETL process

☐ Execution
  ■ we need to find a good execution strategy for ETL processes, not in an ad-hoc way
Logical Model - Architecture Graph [CAiSE 03]

Example

PS1(PKEY, SUPPKEY, DEPT, QTY)
PS2(PKEY, SUPPKEY, DEPT, COST)
S1.PARTSUPP(PKEY, SUPPKEY, DEPT, QTY, COST)
S2.PARTSUPP(PKEY, SUPPKEY, DATE, QTY, COST)
DW.PARTSUPP(PKEY, SUPPKEY, DATE, QTY, COST)
Architecture Graph

Example

PS1 (PKEY, SUPPKEY, DEPT, QTY)
PS2 (PKEY, SUPPKEY, DEPT, COST)
S1.PARTSUPP (PKEY, SUPPKEY, DEPT, QTY, COST)
S2.PARTSUPP (PKEY, SUPPKEY, DATE, QTY, COST)
DW.PARTSUPP (PKEY, SUPPKEY, DATE, QTY, COST)
Architecture Graph

Example
Architecture Graph

Example
Architecture Graph

Example
Architecture Graph

Example
Conceptual to Logical [DOLAP 05]

Example: a conceptual scenario
Conceptual to Logical

- Concepts and attributes → recordsets and attributes
Conceptual to Logical

- Transformations, ETL constraints → Activities

which is the proper execution order?

Timos Sellis – DOLAP 2006
Example: a logical scenario

order equivalence?

$\text{SK}, f_1, f_2$ or $\text{SK}, f_2, f_1$ or ... ?
Optimization of ETL Processes [ICDE 05]

- ETL workflows
  - are complex
  - involve a lot of recordsets and activities
  - comprises of activities that perform the same process to the same set of data

- Common settlement:
  - ad-hoc optimization based on the experience of the designer
  - execute ETL workflow as it is; hopefully, the optimizer of the DBMS would improve the performance
Optimization of ETL Processes

- An ETL workflow is **NOT** a big query
- Techniques adapted from traditional optimization are not enough
  - existence of functions
    - where it is allowed to push an activity before/after a function?
  - existence of black-box activities
    - unknown semantics
    - can not interfere in their interior
  - naming conflicts
Optimization of ETL Processes

- How can we improve an ETL workflow in terms of execution time?

- We model the ETL processes optimization problem as a state search problem
  - we consider each ETL workflow as a state
  - we construct the search space
  - the optimal state is chosen according to our cost model’s criteria, in order to minimize the execution time of an ETL workflow

Timos Sellis – DOLAP 2006
Optimization of ETL Processes

- Transition from one state to the other
  - interchange two activities of the workflow

\[
\text{SWA}(a_1, a_2) \quad \text{SWA}(a_2, a_1)
\]
Optimization of ETL Processes

- Transition from one state to the other
  - replace homologous tasks in parallel flows with an equivalent task over a flow to which these parallel flows converge
  - divide tasks of a joint flow to clones applied to parallel flows that converge towards the joint flow

\[ a, a_1, a_2: \text{homologous activities} \]
“Transition” from one state to the other:

- merge / split group of activities
Optimization of ETL Processes

Example:
- cost model = \( f(\text{card(processed\_rows)}) \)
- input of 8 rows in each flow
- selectivities: \( \sigma = 50\% \), \( SK_1 = SK_2 = 100\% \)
- cost formulae: \( cost(SK) = n\log_2 n \), \( cost(\sigma) = n \) (ignore the cost of \( U \))

\[
c_1 = 2n\log_2 n + n = 56
\]
\[
c_2 = 2(n + (n/2)\log_2 (n/2)) = 32
\]
\[
c_3 = 2n + (n/2)\log_2 (n/2) = 24
\]
Optimization of ETL Processes

☐ Transition Applicability
  - when the *swap* is allowed?
  - when the *factorize* is allowed?
  - when the *distribute* is allowed?

☐ Correctness of the transitions
  - *post-conditions* of ETL workflows
  - *equivalence* of ETL workflows
Optimization of ETL Processes

- **Algorithms**
  - exhaustive (1)
  - heuristic (2)
  - greedy (3)

- Briefly, algorithms (2) and (3) improve the performance of ETL workflows over 70% (avg) during a satisfactory for DW’s period of time (in a time range of sec..10min)
Research Challenges

- Extension of the ETL mechanisms for non-traditional data, like XML/HTML, spatial and biomedical data
- Apply the techniques described to different environments (like Active DWs)
- Use richer semantics to describe sources, reason about them, etc.
Research Challenges – use of ontologies

☐ Key idea
  - an ontology-based approach to facilitate the conceptual design of an ETL scenario

☐ An ontology
  - is a “formal, explicit specification of a shared conceptualization”
  - describes the knowledge in a domain in terms of classes, properties, and relationships between them
  - machine processable
  - formal semantics
  - reasoning mechanisms

☐ The Web Ontology Language (OWL) is used as the language for the ontology
Think of new models for the case of large distributed environments with many sources e.g. P2P

- Can the techniques scale?
- Can they adapt to the different semantics, like approximate and incomplete answers?
- Can we make the techniques “goal”-driven rather than strict: e.g. I want to have 100% over this week’s data, 80% over last week’s, etc?
- How to integrate static and dynamic cases (peers come and leave, others stay there for a long period)?