Data Warehouse Evolution: Trade-offs between Quality and Cost of Query Rewritings*

Amy J. Lee[§], Andreas Koeller[‡], Anisoara Nica[†], and Elke A. Rundensteiner[‡]

(§)Department of I & S Mgmt.
HK Univ. of Science and Technology
Kowloon, Hong Kong
imalee@ust.hk

(‡)Department of CS
Worcester Polytechnic Institute
Worcester, MA 01609-2280
{koeller|rundenst}@cs.wpi.edu

(†) Department of EECS
University of Michigan
Ann Arbor, MI 48109-2122
anica@eecs.umich.edu

Query rewriting has been used as a query optimization technique for several decades to reduce the computational cost of a query. It has generally been assumed that any rewritten query will generate the identical query result as the original query, in terms of both the query interface and the query extent. Hence, this is called "equivalent query rewriting".

Recently, query rewriting with *relaxed semantics* has been proposed as a means of retaining the validity of a data warehouse (i.e., materialized queries) in a changing environment [2, 3, 4]. Attributes in the query interface can be classified as essential or dispensable (if it cannot be retained) according to the query definer's preferences. Similarly, preferences for query extent can be specified, for example, to indicate whether a subset of the original result is acceptable or not. A query rewriting is said to be acceptable if it preserves the essential information of the original query and satisfies the constraint on the view extent. Since each rewriting may preserve the original query to a different degree, a potentially large number of acceptable yet non-equivalent query rewritings may be found. Therefore, we need to semantically select the most promising rewriting out of all possible ones. Research issues that must be answered for solving this problem are outlined below.

We have found that the two most important factors influencing the desirability of a query rewriting are: the information preserved by the rewriting w.r.t. the original query result (*quality*) and the cost of acquiring the query results (*cost*). A rewriting is more desired than others if it is "semantically close" to the original one and could be acquired economically. We have designed the first analytic model, Quality-Cost Model (QC Model), to assess rewritings on both factors [1].

A rewriting has a better quality in terms of the query interface if it retains more dispensable attributes than others. A rewriting is superior in extent to others if it preserves more of the original extent without introducing surplus tuples. The percentage of the tuples preserved by a rewriting is computed by the overlapping query extents between the rewriting and the original query. Estimation of the overlapping extents before rewritings are actually computed is a research issue [1].

Since data content changes are more frequent than schema changes at the ISs, we propose to use the long-term (incremental) view maintenance cost as our indicator for costs. The view maintenance cost is composed of three factors, namely the number of exchanged messages between the information space and the data warehouse, the number of bytes of data transferred between these two sites, and the number of I/Os performed by the external ISs in order to process incremental view maintenance. We have run experimental studies to assess trade-offs between these three factors in a real environment [1].

The *quality* factor is then combined with the *cost* factor to decide the overall ranking of a rewriting. To our best knowledge, this is the first work that deals with this novel issue, the *non-equivalent query rewriting problem*.

*This work was supported in part by the NSF NYI grant #IRI 94-57609. Dr. Rundensteiner would also like to thank IBM for the IBM Partnership Award as well as for IBM Fellowships for her students.

References

[1] A. J. Lee, A. Koeller, A. Nica, and E. A. Rundensteiner. Data Warehouse Evolution: Trade-offs between Quality and Cost. Technical Report WPI-CS-TR-98-2, WPI, 1998.

[2] A. J. Lee, A. Nica, and E. A. Rundensteiner. Keeping Virtual Information Resources Up and Running. In *Proceedings of IBM Centre for Advanced Studies Conference CASCON'97, Best Paper Award*, pp 1–14, November 1997.

[3] A. Nica, A. J. Lee, and E. A. Rundensteiner. The CVS Algorithm for View Synchronization in Evolvable Large-Scale Information Systems. *EDBT'98*, pp 359–373.

[4] E. A. Rundensteiner, A. J. Lee, and A. Nica. On Preserving Views in Evolving Environments. *KRDB'97*, pp 13.1–13.11.