INTRODUCTION

A hypertext view is a hypertext containing data from an underlying database. The materialization of such hypertexts, that is, the actual storage of their pages in the site server, is often a valid option. Suitable auxiliary data structures and algorithms must be designed to guarantee consistency between the structures and contents of each heterogeneous component where base data is stored and those of the derived hypertext view.

This topic covers the maintenance features required by the derived hypertext to enforce consistency between page content and database status (Sindoni, 1998). Specifically, the general problem of maintaining hypertexts after changes in the base data and how to incrementally and automatically maintain the hypertext view are discussed and a solution using a Definition Language for Web page generation and an algorithm and auxiliary data structure for automatic and incremental hypertext view maintenance is presented.

BACKGROUND

Some additional maintenance features are required by a materialized hypertext to enforce consistency between page contents and the current database status. In fact, every time a transaction is issued on the database, its updates must be efficiently and effectively extended to the derived hypertext. In particular, (i) updates must be incremental, that is, only the hypertext pages dependent on database changes must be updated and (ii) all database updates must propagate to the hypertext.

The principle of incremental maintenance has been previously explored by several authors in the context of materialized database views (Blakeley et al., 1986; Gupta et al., 2001; Paraboschi et al., 2003; Vista, 1998; Zhuge et al., 1995). Paraboschi et al. (2003) give a useful overview of the materialized view maintenance problem in the context of multidimensional databases. Blakeley et al. (1986) propose a method in which all database updates are first filtered to remove those that cannot possibly affect the view. For the remaining updates, they apply a differential algorithm to re-evaluate the view expression. This exploits the knowledge provided by both the view definition expression and the database update operations. Gupta et al. (2001) consider a variant of the view maintenance problem: to keep a materialized view up-to-date when the view definition itself changes. They try to “adapt” the view in response to changes in the view definition. Vista (1998) reports on the integration of view maintenance policies into a database query optimizer. She presents the design, implementation and use of a query optimizer responsible for the generation of both maintenance expressions to be used for view maintenance and execution plans. Zhuge et al. (1995) show that decoupling of the base data (at the sources) from the view definition and view maintenance machinery (at the warehouse) can lead the warehouse to compute incorrect views. They introduce an algorithm that eliminates the anomalies.

Fernandez et al. (2000), Sindoni (1998) and Labrinidis & Roussopoulos (2000) have brought these principles to the Web hypertext field. Fernandez et al. (2000) provide a declarative query language for hypertext view specification and a template language for specification of its HTML representation. Sindoni (1998) deals with the maintenance issues required by a derived hypertext to enforce consistency between page content and database state. Hypertext views are defined as nested oid-based views over the set of base relations. A specific logical model is used to describe the structure of the hypertext and a nested relational algebra extended with an oid invention operator is proposed, which allows views and view updates to be defined. Labrinidis & Roussopoulos (2000) analytically and quantitatively compare three materialization policies (inside the DBMS, at the web server and virtual). Their results indicate that materialization at the Web server is a more scalable solution and can facilitate an order of magnitude more users than the other two policies, even under heavy update workloads.

The orthogonal problem of deferring maintenance operations, thus allowing the definition of different policies, has been studied by Bunker et al. (2001), who provide an overview of the view maintenance subsystem of a commercial data warehouse system. They describe optimizations and discuss how the system’s focus on star schemas and data warehousing influences the maintenance subsystem.
Related Content

Closed-Itemset Incremental-Mining Problem
www.igi-global.com/chapter/closed-itemset-incremental-mining-problem/10583?camid=4v1a

Data Quality-Based Requirements Elicitation for Decision Support
www.igi-global.com/chapter/data-quality-based-requirements-elicitation/7616?camid=4v1a

Secure Building Blocks for Data Privacy
www.igi-global.com/chapter/secure-building-blocks-data-privacy/11053?camid=4v1a

Comparative Genome Annotation Systems
www.igi-global.com/chapter/comparative-genome-annotation-systems/7731?camid=4v1a