Foundations for Information Integration

27.06 - 02.07.99

organized by

Serge Abiteboul, Dana Florescu, Alon Levy, Guido Moerkotte

We are currently witnessing an explosion in the amount of information that is available on-line (e.g., sources on the Internet, company-wide intranets, etc). Providing easy and efficient access to this information—known as the problem of *data integration* raises an important challenge to several fields of Computer Science including Database Systems, Artificial Intelligence, Operating Systems, Networking and Human Computer Interaction.
The challenge is to develop techniques for providing uniform access to the wealth of available information. Usually, data integration is achieved by providing the user a mediated schema that hides the details of each of the data sources, and lets the user focus on specifying what he wants, rather than specifying how or where to find the information. The data integration problem is complicated by the fact that the data sources are autonomous, employ different data models and are heterogeneous both semantically and syntactically. Furthermore, the data sources are often only semistructured (e.g., they do not have explicit schemas, the schemas are unknown, or the sources contain extraneous information such as advertisements or other information meant for human consumption). The techniques that need to be developed include modeling of the contents of information sources, high-level query facilities, flexible approaches to selecting relevant sources, novel methods for query optimization and flexible query execution models.

Contents

1 Active Views

Serge Abiteboul (INRIA, Rocquencourt)

Active Views is a system for allowing Web clients to interact with a database and among themselves. The main ideas persued are

- the use of XML all over from the interface that uses a standard browser to the repository that is an XML repository (Axielle of Ardent Software)
- the emphasis on activity and reactivity. For instance a client may request to be notified when some data changes or some event occurs.
- the emphasis on declarativity. The specification is in a declarative language AVL that includes an XML query language.

The main test application for Active Views is a Web Catalog.

2 Flexible and Scalable Cost-based Query Planning for Mediators: A transformational Approach

Jose Ambite (University of Southern California, Marina del Rey)

We presented an approach to cost-based query planning in mediators based on a general planning paradigm called Planning by Rewrite (PbR). Our work yields several contributions. First, the PbR-based query planner combines both source integration and cost optimization in a single search space. Second, by using local search techniques our planner explores the search space efficiently and produces higher quality lans. Third, because our planner is an instantiation of a domain-independent framework, it is very flexible and can be extended ina principled way.

3 Optimizing Techniques for Semistructured Integration Systems: An Algebraic Perspective

Vassilis Christophides (University of Crete, Heraklion)

Integration Systems based on a semistructed middleware representation considerably simplify the integration process, as they allow a completely declarative specification. However, query evaluation in these systems is still very expensive, as much less optimization techniques are available compared to, e.g., federated database systems. This is due mainly to two reasons: first, there is no well-defined and understood optimization framework for semistructured data, then in a heterogeneous context it is difficult to understand source query capabilities in order to take advantage for query evaluation.
To resolve the first difficulty, we propose a new algebra for semistructured data. This algebra contains a usual object algebra whose rewriting properties are well-known plus two new semistructured operators. These new operators have good rewriting properties allowing them to be incorporated in the optimization process. To resolve the second difficulty we propose to describe source query capabilities in terms of the algebra. We show that this approach allows an accurate description of both "structured" query languages (SQL, OQL) and "semistructured" access languages (like full-text indexes). Finally we show how this complete framework is used for optimization in various examples. The algebraic approach is currently implemented in the YAT system.

4 Using XQL for Information Integration

Peter Fankhauser (GMD, Darmstadt)

XQL (XML Query Language) is a query language for XML documents. Its main construct are path expressions to select subtrees from a document tree. To use XQL for information integration mainly two additional features are

required - joins and multi-document support. Joins can be achieved by path expressions which navigate across multiple subtrees rather than only into subtrees. Multi-documents can be supported by operators which dereference links. The formal semantics of XQL with these extensions can be described by mapping XQL to an algebra. The domain of the algebra are XML documents represented as locators. Locators are generated by associating to each node the entire path from the document root, ordered by left to right, depth first traversal of the document tree. To preserve document-order under algebraic manipulation every node on every locator gets associated a unique id in document order. On this basis a variant of relational algebra - the locator algebra - can be used for describing XQL. Select selects sublists of locators, project discards subpaths from locators, union and difference combine lists of locators in document order, and cartesian product multiplies two lists of locators by concatenating each locator from the first list with each locator from the second list in document order. The locator algebra preserves document order and the possibly irregular document context of manipulated subtrees. With small exceptions the usual equivalence axioms for relational algebra also hold for the locator algebra; cartesian product is non-commutative. More details on the current state of XQL can be found at http://metalab.unc.edu/xql. Example applications using XQL for information integration can be found via http://xml.darmstadt.gmd.de.

5 Experiences with Strudel: Memoirs from the Information Trenches

Mary Fernandez (AT&T Labs Research, Florham Park)

High-speed intranet and web browsers have increased dramatically the demand for integrated information systems in large corporations. High-speed intranets reduce virtual and real distances between sources, and browsers provide a common platform for deployment. Nonetheless, implementing data intensive Web sitex requires tedious programming. We describe a case-study of a production Web site in AT&T, called Hightoll Notifier (HTN). HTN was reengineered using Daytona, a RDBMS, and Strudel, a system for Web site creation and management. In Strudel, a Web site is specified by a declarative

query over underlying data sources. Declarative specifications unlike implementations in imperative scripting languages, have several benefits: they are analyzable, optimizable, and more concise. In our case study, the Strudel site definition was over 4 times smaller than the equivalent implementation in a scripting language. Moreover, the Strudel definition can be evaluated under different evaluation policies, which generalize the current practice of either generating sites only statically or dynamically.

6 Integrating and Querying Dynamic Web Data

Juliana Freire (Bell Labs, Murray Hill)

There is growing interest in developing Web-based applications that allow end users to search for information about products and services in ways that cannot be directly accomplished by search engines. The design of database systems (Web Bases) to support such applications is currently an active area of research. A particularly challenging problem is designing Web Bases for querying.

We propose a 3-layer architecture for designing and implementing Web Bases for querying dynamic Web content (i.e. data that can only be extracted by filling multiple forms). The lowest layer, the virtual physical layer, provides navigation independence by shielding the user from the complexity associated with retrieving data from raw Web sources. Next, the traditional logical layer supports site independence. The top layer is analogous to the external schema layer in traditional databases.

We use this architectural framework to address two problems unique to Web Bases–retrieving dynamic Web contents in the virtual physical layer and querying of the external schema by the end user. The layered architecture makes it possible to automate data retrieval to a much greater deal from existing proposals. Wrappers for the virtual physical layers can be created semi-automatically, without requiring the Web Base designer to write programs. For the external layer, we propose a semantic extension of the universal relation interface which provides powerful yet simple ad hoc querying capabilities.

7 A General Framework for Data Cleaning

Helena Galhardas (INRIA, Rocquencourt)

Data integration solutions dealing with large amounts of data have been strongly required in the last few years. Besides the traditional data integration problems (e.g. schema integration, local to global schema mappings), three additional data problems have to be dealt with: (1) the absence of universal keys across different databases that is known as the object identity problem, (2) the existence of keyborad errors in the data, and (3) the presence of inconsistencies in data coming from multiple sources. Dealing with these problems is globally called the data cleaning process. In this work, we propose a framework which offers the fundamental services required by this process: data transformation, duplicate elimination and multi-table matching. These services are implemented using a set of purposely designed macro-operators. Moreover, we propose an SQL extension for specifying each of the macro-operators. One important feature of the framework is the ability of explicitly including the human interaction in the process. The main novelty of the work is that the framework permits the following performance optimizations which are tailored for data cleaning applications: mixed evaluation, neighborhood hash join, decision push-down and short-circuited computation. We measure the benefits of each.

8 Adaptive Generation for Data Integration

Zack Ives (University of Washington, Seattle)

Query processing in data integration occurs over network-bound, autonomous data sources. This requires extensions to traditional optimization and execution techniques for three reasons: there is an absence of quality statistics about the data, data transfer rates are unpredictable and bursty, and slow or unavailable data sources can often be replaced by overlapping or mirrored sources. This paper presents the Tukwila data integration system, designed to support adaptivity at its core using a two-pronged approach. Interleaved

planning and execution with partial optimization allows Tukwila to quickly recover from decisions based on inaccurate estimates. During execution, Tukwila uses adaptive query operators such as the double pipelined hash join, which produces answers quickly, and the dynamic collector, which robustly and efficiently computes unions across overlapping data sources. In this talk I will describe the Tukwila architecture, which extends previous innovations in adaptive execution (such as query scrambling, mid-execution re-optimization, and choose nodes), and present experimental evidence that our techniques result in behavior desirable for a data integration system.

9 ObjectGlobe: Database Patchwork on the Internet

Alfons Kemper (Universität Passau)

The vision of the ObjectGlobe project is to ship functionality to the data providers instead of shipping (potentially huge violumes of) data to the client systems.
The core of our ObjectGlobe system is a distributed query engine, which is capable of incorporating services of three different service providers:

- Data provider,
- Functionality provider and
- Cycle Provider.

The integration of data providers means that wrappers for arbitray data sources can be loaded dynamically into a running query processor. Through the use of these wrappers, the data sources are also included dynamically.
Functionality can be included in several different forms:

- Iterator.
- Aggregate function.
- Transformer function.

- Predicate function.

As with the wrappers the integration is performed dynamically by loading the class files from another server.
A cycle provider has to run a Java VM and on it our server program. Since it is possible to distribute the operators of a query evaluation plan among arbitrary (known) server processes, queries can exploit the new server at once after the registration.
The Java security model (e.g., executing operations in a protected "sandbox"environment, authenticating dynamically loaded code via digital signatures) is utilized to guarantee the safety of data and function providers when new functionalty is loaded via the Internet into the ObjectGlobe query engine.
More information can be found on our ObjectGlobe web site
http://www.db.fmi.uni-passau.de/projects/ObjectGlobe.phtml

10 What's in a name? – First Steps Towards a Structural Approach to Integrating Large Content-based Knowledg-Bases

Ralf Küsters (RWTH Aachen, Aachen)

We are concerned with the problem of integrating two content-based knowledge bases. It is well known that even the same slice of reality can be modeled in various ways. This ranges from single morphological variants of identifiers to reification of relationships. Hence human intervention is required in order to set up some kind of correspondance meaning between the schemas. However, modern real-life ontologies, such as the Galen medical ontology, are so large that it seems not feasible for humans to perform this task without computer aid. We therefore aim at exploiting the structural information to help find candiate equivalent concepts from one ontology to the other.
To this end, we have set up a formal framework for correspondance mappings and investigated limitations of structural information to find such mappings. Finally, first algorithmic and empirical results have been presented.

11 Regression Testing for Wrapper Maintenance

Nicholas Kushmerick (University College Dublin, Dublin)

Recent work on Internet information integration assumes a library of *wrappers*, specialized information extraction procedures. Maintaining wrappers is difficult, because the formatting regularities on which they rely often change. The *wrapper verification* problem is to determine whether a wrapper is correct. Standard regression testing approaches are inappropriate, because both the formatting regularities and a site's underlying content may change. We introduce Rapture, a fully-implemented, domain-independent verification algorithm. Rapture uses well-motivated heuristics to compute the similarity between a wrapper's expected and observed output. Experiments with 27 actual Internet sites show a substantial performance improvement over standard regression testing.

12 A database perspective on Lotus Domino/Notes

C. Mohan (IBM Almaden Research Center, San Jose)

Lotus Notes was released in 1989 as a groupware product. both the server (Domino) and the client (Notes) versions of the product do their own persistent storage management by directly using the file system, without relying on a DBMS. While it was designed initially as a workgroup product for use by a small number of users, it has been enhanced extensively over the years, allowing it to be successfully deployed in many large enterprises with a year-end 1998 installed base of 34 million seats. Unlike in RDBMSs, support for semistructured data management has been one of the unique features of Notes from the very beginning. From the first release, support for replication and disconnected operation has also been one of the most significant and innovative features. More recently, a major feature implemented in the

latest relase (R5) is a traditional log-based recovery scheme. This was done by extending the ARIES recovery method to take into account the unique characteristics of Notes' storage engine functionality and enhancements over the last few years have transformed Domino into a web application server.

13 Queries with Incomplete Answers over Semistructured Data

Werner Nutt (DFKI, Saarbrücken)

Semistructured data occur in situations where information lacks a homogeneous structure and is incomplete. Yet, up to now, the incompleteness of information has not been reflected by special features for query languages for semistructured data. Thus, we have investigated principles of queries that allow for incomplete answers.
Queries over classical structured data models contain a number of variables and conditions on these variables. An answer is a binding of the variables by elements of the database such that the conditions are satisfied. In the present case, we loosen this concept in so far as we allow also answers that are partial, that is, not all variables are bound. Partial answers make it necessary to refine the model of query evaluation. The first modification relates to the satisfaction of conditions: under some circumstances we consider conditions involving unbound variables as satisfied. Second, the order to prevent a proliferation of answers, we only accept answers that are maximal in the sense that there are no assignments that bind more variables and satisfy the conditions of the query.
Our model of query evaluation consists of two phases, a search and a filter phase. In the search phase, we use a graph pattern to match a maximal portion of the database graph. In the filter phase, the maximal matchings are subjected to constraints which may be weak or strong. We describe polynomial algorithms for evaluating special types of queries and assess the complexity of evaluating other queries for several kinds of constraints.

14 Query Reformulation for Data Integration

Rachel Pottinger (University of Washington, Seattle)

Query reformulation is the problem of translating the user query from a mediated schema to a query over the data sources. Query reformulation is an instance of the problem of answering queries using views. This problem has been studied in the literature in depth; algorithms for answering queries using views have been studied for conjunctive queries, recursive queries, and cases in which there exist limitations on the access patterns to the data. Previously developed algorithms did not scale up to large number of views and larger queries. We present a new algorithm for answering queries using views, and present experiments that show that it scales up and performs significantly better than previous algorithms.

15 Wide area query processing

Louiqa Raschid (University of Maryland, College Park)

We present our research on technology to scale wrapper mediator architectures for query processing with heterogeneous information sources across a dynamic wide area network. We call these sources WebSources. We develop a cost model for WebSources. It uses WebPT - a tool that learns using query feedback from accessing WebSources, and predicts a response time for accessing the WebSource and a confidence for the prediction. We develop a Web query optimizer (WQO) that extends a traditional relational optimizer for this environment. The WQO uses a capability based rewriting CBR tool. The CBR tool respects the limited capabilities of WebSources in finding some accepted queries that can be evaluated on the WebSources. The CBR tool generates a pre-plan that reflects the limitations of queries for WebSources. WQO then uses the pre-plan to drive a relational optimizer, which generates plans which respect the pre-plan knowledge. The WQO uses the cost model to choose a good plan.

16 A Flexible and Semantic Approach for Information Integration: the PICSEL Approach

Marie-Christine Rousset (Université Paris Sud, Paris)

PICSEL is an information integration system over data sources that are distributed and possibly heterogeneous. The main characteristics of PICSEL are:

1. The integration of the sources is driven by the semantics of the domain of the application, through the logical description of an ontology.
2. The description of the content of the sources is done by a set of logical views over the domain onlology.
3. PICSEL exploits the expressive power and the algorithms of CARIN. CARIN combines in a logical and uniform setting the expressive power of Datalog and Description Logic.

The main algorithmic contribution of PICSEL is a new algorithm for reformulating queries expressed in terms of domain relations into a set of specalized queries expressed in terms of source relations. This provides a new class of queries and views for which the problem of rewriting queries using views is decidable.

17 Type Checking for XML Transformations

Dan Suciu (AT&T Labs Research, Florham Park)

This work is motivated by the need to type check declarative XML transformations w.r.t. the input and the output DTD. An XML document is viewed as an unranked, ordered tree.
First, we show that type inference (a stronger problem than type checking) does not have a solution for the simplest XML transformation languages.

Second, we give a solution to type checking for a very general class of tree transformations: those expressed with k-pebble transformers (a machine defined here). Deriving such a general class is necessary since no generally accepted XML query/transformation languages exist. We sow that all transformations expressed in current XML languages (XSL, XML-QL, UnQL, StruQL) are expressed by k-pebble transformers.
Third, we advocate k-pebble transformers as a simple, robust, and elegant yardstick for measuring XML query language expressive power.