Database Research at Arizona State University

Susan D. Urban Suzanne W. Dietrich Forouzan Golshani

Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287-5406
{s.urban | dietrich | golshani }@asu.edu

1. INTRODUCTION

This paper provides an overview of the current database research activities within the Intelligent Information Systems Group in the Department of Computer Science and Engineering at Arizona State University. The focus of our research is on the integration of data and knowledge management issues, with specific emphasis on multimedia systems, object-oriented databases, active databases, deductive databases, and heterogeneous, distributed database environments. For further details, see our Department Web page at **http://www.eas.asu.edu/~csedept**.

2. ANOMALOUS RULE BEHAVIOR IN ACTIVE, OBJECT-ORIENTED DATABASE SYSTEMS

Susan D. Urban

This research (Grant No. IRI-9109195) has addressed the problem of termination and confluence in active database rules, building on previous research involving the generation of active database rules from constraints (NSF Grant No. IRI-8996260). The research has been conducted using a declarative, object-oriented model, known as CDOL (Comprehensive, Declarative Object Language), which has been enhanced with a general and flexible, active database architecture. Active rules in CDOL are modeled as conditional term rewrite rules that rewrite the current database state. Rules communicate using messages that represent update requests and occurrences of events.

Since the termination problem is undecidable in general, the problem has been approached in this work by developing sufficient conditions that ensure termination. Our results consist of three termination-proving methods, respectively called the *triggering graph* method, the *termination orderings* method, and the *refined triggering graph* method. These methods rely upon graph-theoretic, algebraic, and logic (satisfiability) approaches, respectively. The triggering graph method is a conservative approach that uses the notion of acyclic graphs to ensure terminating rules. The termination orderings method applies results from reducing conditional term rewrite systems to address a syntactic approach to the termination problem. Finally, the refined triggering graph method makes use of events and rule conditions to establish decidable sufficient conditions for removing arcs from a triggering graph that provably cannot be involved in infinite rule triggering sequences. The process of establishing sufficient conditions makes use of satisfiability algorithms associated with semantic query processing in CDOL. Each termination method is complementary to the others, so that all three techniques can be applied for a more complete analysis of termination results.

Work on confluence analysis has applied results from terminating conditional term rewrite systems with convergent, contextual critical pairs to address the confluence of active database rules. By mapping termination and confluence problems for active rules to the known domain of termination and confluence problems in condition term rewrite systems, it is possible to draw upon established results for addressing the confluence problem, thus establishing a sound theoretical basis for confluence analysis of active rules. In particular, we make use of results from conditional term rewrite systems that require a careful analysis of all critical overlaps between rule conditions.

[Kara91] Karadimce, A. and Urban S. D., "Diagnosing Anomalous Rule Behavior in Deductive Databases with Integrity Maintenance Production Rules," *Proc. of the 3rd Int. Workshop on Foundation of Models and Languages for Data and Objects*, Austria, Sept. 1991, pp. 77-102.

[Kara93] Karadimce, A. and Urban S. D., "A Framework for Declarative Update and Constraint Maintenance in Object-Oriented Databases," *Proc. of the 9th Int. Conf. on Data Eng.*, Austria, Apr. 1993, pp. 391-398.

[Kara94] Karadimce, A. and Urban, S. D., "Conditional Term Rewriting Systems as a Formal Basis for Analysis of Active Rules," *IEEE Computer Society Int. Workshop on Research Issues in Data Eng.: Active Database Systems*, Houston, Feb., 1994.

[Kara96] Karadimce, A. and Urban, S. D., "Refined Triggering Graphs: A Logic-Based Approach to Termination Analysis in an Active, Object-Oriented Database," to appear in *Proc. of the 12th Int. Conf. on Data Eng.*, Feb. 1996.

3. CONDITION MONITORING IN ACTIVE, DEDUCTIVE DATABASE SYSTEMS

Suzanne W. Dietrich

This work on monitoring conditions in active, deductive database systems has been performed jointly with John Harrison, who is now in the Department of Computer Science in the University of Queensland in Brisbane, Australia. The focus of the research was on the efficient monitoring of conditions expressed in the context of an active database system that supports declarative conditions expressed as safe, recursive Datalog rules with stratified negation. One difficulty in monitoring is posed by the expressiveness of the conditions in a deductive database environment, which may be defined in terms of SPJ (selection, projection, join), union, (stratified) negation and recursion. Another difficulty in monitoring is that, unlike query evaluation, which propagates bindings extracted from the queries *down* to filter the access to base relations, condition monitoring must propagate bindings from the changes to the base relations *up* through the rules that define the conditions. Yet another difficulty in monitoring is that the condition monitor must have access to both database states to monitor and reason about the conditions. We have developed an incremental condition monitoring algorithm known as the Propagation/Filtration (PF) algorithm, which refers to its two-phased approach to evaluation: a propagation phase and a filter phase. The propagation phase propagates the base relation changes up through the rules and identifies potential changes to the intensional relations. The filtration phase filters the potential changes to identify actual changes, since a potential change may represent the addition or removal of a proof, which may or may not represent an actual change to the database state.

The PF algorithm also provides for the incremental maintenance of materialized views. Updates to stored relations that participate, directly or indirectly, in the definition of a materialized view can cause the materialized view to be inconsistent with its definition. To avoid the costs of complete recomputation of the view in the updated database state, the materialized view can be updated incrementally using the update propagation approach provided by the PF algorithm.

[Harr92a] Harrison, J., *Condition Monitoring in an Active Deductive Database*, Ph.D. Diss., Dept. of Computer Science and Eng., Arizona State University, Aug. 1992.

[Harr92b] Harrison, J. and Dietrich, S. W., "Maintenance of Materialized Views in Deductive Databases: An Update Propagation Approach," *Proc. of the Deductive Database Workshop* in conjunction with the *Joint Int. Conf. and Symposium on Logic Programming*, Washington, D. C., Nov. 1992, pp. 56-65

[Harr93] Harrison, J. and Dietrich, S. W., "Integrating Active and Deductive Rules", *Proc. of the First Int. Workshop on Rules In Database Sys.*, Edinburgh, Scotland, 30th August-1st Sept. 1993, Springer Verlag, pp. 288-305

[Harr94] Harrison, J. and Dietrich, S. W., "Incremental View Maintenance", *Proc. of the 5th Australasian Database Conf.*, 17-18th Jan. 1994, New Zealand, pp. 45-63.

4. A DOOD RANCH PROJECT

Susan D. Urban and Suzanne W. Dietrich

A DOOD RANCH Project (Active, Deductive, Object-Oriented Databases: Relating Action, Negation, Constraints, and Horn rules) is investigating two complementary aspects of active database development environments: the analysis of rules for termination and confluence properties and the run-time testing and debugging of active rules. The research (NSF Grant No. IRI-9410993) represents an integration of the projects described in Sections 2 and 3 and is being performed in the context of an active, deductive, and object-oriented database. The run-time testing and debugging environment will allow users to step through the rule triggering process, observing rule behavior and testing different rule triggering sequences. The execution model includes a propagation algorithm for monitoring changes to derived data, thus integrating deductive and active rule processing. We intend to investigate the integration of rule generation (from constraints) and transaction development to produce a practical approach to the specification and analysis of rules prior to rule execution.

An initial prototype of a run-time debugging environment for active rules has already been developed as part of this work. The Prototype Environment for Active Rule Debugging (PEARD) simulates an active rule environment using Smalltalk with a graphical interface created using VisualWorks. The rule simulation environment includes various tools for the specification and analysis of active rules. Rule definition is flexible, where changes to rules can be applied immediately during a debugging session without recompiling the system. A breakpoint debugging tool allows breakpoints to be set so that the state of variables may be inspected and changed any time a breakpoint is reached during rule execution. A rule visualization tool displays the rule triggering process in graph form, supporting different visualization granularities to help the user understand rule execution. Other debugging features include a means for detecting potential cycles in rule execution and a utility to examine different rule execution paths from the same point in the rule triggering process. The development of PEARD has helped to identify design and development considerations for active database debugging tools.

[Diet92] Dietrich, S. W., Urban, S. D., Harrison, J., and Karadimce, A., "A DOOD Ranch at ASU: Integrating Active, Deductive, and Object-Oriented Databases," *IEEE Computer Society Data Eng. Bulletin*, vol. 15, no. 1-4, Dec. 1992, pp. 40-43.

[Urba93] Urban, S. D. and Dietrich, S. W., "A Deductive, Object-Oriented Model as a Formal Framework for Active Database Environments," *Proc. of the Workshop on Combining Declarative and Object-Oriented Databases*, Washington, D.C., May 1993.

[Jaeh95] Jaehne, Alexander, *A Prototype of a Run-Time Debugging Environment for Active Database Rules*, M.S. Thesis, Dept. of Computer Science and Eng., Arizona State University, Summer 1995, pp. 116.

[Urba95] Jaehne, A., Urban, S. D., and Dietrich, S. W., "PEARD: Prototype Environment for Active Rule Debugging," submitted for journal publication.

5. INFRASTRUCTURE FOR INTEGRATED, HETEROGENEOUS ENGINEERING DATABASES AND DESIGN HISTORIES

Susan D. Urban

This research has investigated: 1) integrated databases to support engineering design, and 2) extensions of database functionality for capturing design histories (NSF Grant No. IRI-9117143). The work has been performed jointly with Prof. Jami J. Shah in the Department of Mechanical and Aerospace Engineering at Arizona State University.

The focus of the integration portion of the research has been on the use of object-oriented database technology for developing an integrated, multidatabase environment for supporting mechanical engineering design. The objective of the environment has been to provide designers with a high level view of how all of the different files and databases used in the design process fit together to produce a global design of a product. The integrated view makes use of the STEP standard and the EXPRESS semantic data model to provide a framework for the representation of metadata that describes object-oriented views of the different tools of the environment. Communication is achieved using a process that is similar to the dynamic interface language of the Object Management Group. The integrated environment has established a framework that supports future investigation of query processing and constraint propagation in a heterogeneous database environment.

The second component of the research has involved the development of a database environment that is capable of capturing design histories. A design history is a database for associating design objects at various phases of the design to the design processes that produced or modified the objects and recording the reasons and rationale for performing various design steps. As part of this research, we have developed the Design History Data Model (DHDM), an extended object-oriented data model that supports process modeling capabilities. The DHDM incorporates a meta data and schema evolution component to support the capture of both pre-specified design activities and dynamic design activities. Prespecifiable design specifications are design tasks that are well-known and can be planned in advance. Dynamic design involves the specification of design tasks as they actually occur using schema evolution. Modeling design activities is achieved through the use of process classes that are composed of subprocesses through the use of temporal aggregation concepts. Specific occurrences of design activities are modeled as instances of process classes. We are currently concentrating on the formalization of the DHDM, focusing on the temporal meaning of the process modeling concept.

[Urba93] Urban, S., Shah, J., Rogers, M., "An Overview of the ASU Engineering Database Project: Interoperability in Engineering Design," *Proc. of the 3rd Int. Workshop on Research Issues in Data Eng.: Interoperability in Multidatabase Sys.*, Austria, Apr. 1993, pp. 73-76.

[Urba94] Urban, S., Shah, J., Rogers, M., Jeon, D., Ravi, P., Bliznakov, P., "A Heterogeneous, Active Database Architecture for Engineering Data Management," *J. of Computer Integrated Manufacturing*, vol. 7, no. 5, 1994, pp. 276-293.

[Jeon95] Jeon, D., Urban, S., Shah, J., Liu, H., Bliznakov, P., Rogers, M., "Metadata Extensions to an Object-Oriented Data Model for the Dynamic Capture of Engineering Design Histories," *Proc. of the IFIP-TC-2 Working Conf. on Database Semantics (DS-6)*, Stone Mountain, May 1995, to be published by Chapman and Hall, 1996.

[Urba95] Urban, S., Shah, J., Liu, H., Ravi, P., and Rogers, M., "The Shared Design Manager: A Flexible Approach to Interoperability in Engineering Design, accepted for publication in *J. of Integrated Computer-Aided Eng.*, to appear in 1995.

6. QUERY PROCESSING AND ACTIVE RULES IN MULTIDATABASE ENVIRONMENTS

Susan D. Urban

The focus of this research project is on the investigation of 1) query processing issues in multidatabase environments and 2) the use of active rules in multidatabase environments for the maintenance of integrity constraints between different databases. Our initial work in this area involved the use of an object algebra as a global query language for access to relational databases. Our approach uses a self-describing data model to build a meta database that, in addition to a global schema, contains information about the local schemas and how the global schema maps to the local schemas. Structural mappings were implemented by developing translation rules that reason about the semantic differences to translate queries over the global schema, expressed using an object algebra, into SQL queries.

Our current directions for query processing are to extend our architecture to integrate not only structured databases, but also unstructured components such as file systems, CAE and CASE tools, and legacy systems. In particular, we are investigating the use of an object algebra as the query language for expressing queries over an integrated schema that combines structural database mappings with operational interfaces. The use of an object-oriented database as a data warehouse is also being investigated to support storage of semantic information about the heterogeneous environment and to support the temporary caching of objects for constraint checking and query processing.

Another important area of this work addresses the issue of integrity constraint maintenance in a multidatabase environment and the use of active database concepts to automate this task as much as possible. We are investigating techniques for the expression of non-trivial constraints which can cross local database boundaries. We are also investigating the extension of current active rule concepts to define a distributed, active rule language and environment, where active rules for maintaining interdatabase constraints are automatically generated from constraint specifications. The goal of this research is to provide a high level, declarative constraint specification language for a multidatabase environment with a compiler that automatically generates optimized integrity maintenance rules for global integrity constraints.

[Urba94] Urban. S. and Abdellatif, T., "An Object-Oriented Interface to Relational Databases in a Heterogeneous Database Environment," *Proc. of the 1994 Dist. Computing Sys. Conf.*, Poland, June 1994, pp. 387-395.

[Urba95] Urban, S. and Abdellatif, T., "Query Translation from Object-Oriented to Relational Queries in a Heterogeneous Database Environment," *J. of Sys. Integration*, vol. 5, 1995, pp. 123-156.

[Gome96] Gomez, L. "An Active Approach for Constraint Maintenance in a Multidatabase Environment," Arizona State University, Dept. of Computer Science and Eng., Ph.D. proposal, Spring 1996.

[Lava96] Lavariega, J. "Object Query Processing Using Structural Mappings and Operational Interfaces in a Multidatabase Environment," Arizona State University, Dept. of Computer Science and Eng., Ph.D. proposal, Spring 1996.

7. NEXT GENERATION DATABASE APPLICATIONS

Suzanne W. Dietrich

With the maturity of research on next generation database systems including the integration of active, declarative and object-oriented features, the database community realizes the importance of investigating applications that can take advantage of these advanced features. One such application domain is that of software maintenance. Together with Frank W. Calliss, a colleague in the area of software engineering within the Department of Computer Science and Engineering at ASU, we have focused on a particular technique for program understanding known as inter-module code analysis (IMCA). IMCA is an analysis technique aimed specifically at helping a programmer analyze a program that consists of a collection of interconnected modules, where a module is a construct that provides encapsulation and abstraction capabilities to a programmer. Our work has focused on using deductive database systems and DOODs (declarative, object-oriented database systems) for implementing an inter-module code analysis system of object-based languages, such as Ada. We are currently extending the approach to object-oriented languages such as C++. IMCA is an application that can take advantage of the additional capabilities that next generation database systems have to offer, such as inherent support for expressive rules, complex objects, class hierarchies and inheritance.

Our focus on applications has led to an additional emphasis of technology transfer with local industry. Bull Worldwide Information Systems has not only provided funding for the above work on IMCA but has also funded projects on database performance and distributed databases.

[Diet92a] Dietrich, S. W. and Calliss, F. W., "A Conceptual Design for a Code Analysis Knowledge Base," *J. of Software Maintenance - Research and Practice*, vol. 4, Mar. 1992, pp. 19-36.

[Diet92b] Dietrich, S. W., Brown, M., Cortes-Rello, E., and Wunderlin, S., "A Practitioner's Introduction to Database Performance Benchmarks and Measurement," *The Computer Journal*, vol. 35 (4), Aug. 1992, pp. 322-331.

[Diet95] Dietrich, S. W. and Calliss, F. W., "Using Deductive Databases to Facilitate the Maintenance Process," *Proc. of the 1995 Phoenix Conf. on Computers and Communication*, Mar. 1995, pp. 310-318.

[Diet96] Dietrich, S. W., Fan, C., and Cortes-Rello, E., "An Application of Fragmentation Transparency in a Distributed Database System: A Case Study," To appear in *J. of Sys. and Software*, Dec. 1996.

8. DATABASE EDUCATION

Suzanne W. Dietrich and Susan D. Urban

This research (funded by NSF Grant No. DUE-9451489) focuses on the use of cooperative group learning concepts in support of an undergraduate database management course that emphasizes the theoretical and practical aspects of database application development. Specifically, this work introduced cooperative group projects that focus on the practical aspects of databases using relational database technology. The course project is divided into three main phases, involving requirements analysis and conceptual design, relational database mapping and prototyping, and database system implementation using Microsoft Access. The project deliverables are designed so that students not only develop a database implementation, but also evaluate their design in terms of functional dependencies, normal forms, the lossless join property, and the dependency preservation property. These deliverables and their evaluation illustrate the need for sound database design principles. Students are required to actively participate in each phase, with students assuming different roles in each phase to allow them to experience different leadership responsibilities. As part of the grading process, students evaluate their own performance as well as the performance of others in the group.

Previous work by the principal investigator involving database education includes the development of a database interpreter for relational query languages. RDBI is an educational tool, implemented in Prolog, that takes advantage of the connection between logic programming and databases to provide an evaluator for relational query languages. The tool utilizes a common data definition facility and recognizes the following query languages: relational algebra, domain relational calculus, tuple relational calculus, and SQL. The students use RDBI for a sequence of homework assignments that both provide invaluable feedback on each language and illustrate the differences between the languages.

(See `http://www.eas.asu.edu/~cse412` for more information on the cooperative group projects and RDBI, which is available by ftp.)

[Diet93] Dietrich, S. W., "An Educational Tool for Formal Relational Database Query Languages," *Computer Science Education*, vol. 4, 1993, pp. 157-184.

[Diet96] Dietrich, S. W. and Urban, S. D., "Database Theory in Practice: Learning from Cooperative Group Projects," to appear in *Proc. of the 27th SIGCSE Tech. Symp.*, Feb. 1996.

9. MULTIMEDIA DATABASE SYSTEM EVA

Forouzan Golshani

EVA is a multimedia database system capable of storage, retrieval, management, analysis and delivery of objects of various media types. The interface language deals with the temporal and spatial aspects of multimedia information retrieval and delivery, in addition to the usual capabilities provided by the ordinary database languages. EVA has five classes of operations, namely: operations for querying and updating multimedia information, operations for screen management, temporal operators, operators for specifying rules and constraints, and aggregation (computational) operators.

One of the unique features of this system is its ability to manage semantic information on multimedia objects, particularly images and motion pictures. Like other types of digital information, images must be classified based on the semantics of their contents. The most discernible difference between still images and moving pictures stems from movements and variations. Thus to go from the realm of still image repositories to video databases, we must be able to deal with motion. Particularly, we need the ability to classify objects appearing in a video sequence based on their characteristics and features such as shape or color, as well as their movements. By describing the movements that we derive from the process of motion analysis, we introduce a dual hierarchy consisting of spatial and temporal parts for video sequence representation. This gives us the flexibility to examine arbitrary frames at various levels of abstraction, and to retrieve the associated temporal information (say, object trajectories) in addition to the spatial representation. Our algorithm for motion detection uses the motion compensation component of the MPEG video encoding scheme. The algorithm then computes trajectories for objects of interest. All this content information is stored in predefined structures as part of the multimedia information repository.

EVA, an extension of the query language Varqa, is a functional language whose notation is based on that of conventional set theory. It is formally defined in an algebraic framework. EVA is object oriented and supports objects, object classes, attributes and methods of objects, and relationships between objects.

[Dimi94] Dimitrova, N. and Golshani, F., "Rx for Semantic Video Retrieval," *Proc. of the ACM Multimedia Conf.*, San Francisco, Oct. 1994, ACM Press, pp. 219-226.
[Gols94a] Golshani, F., "Images Glore", *IEEE MultiMedia*, vol.1, no. 4, Winter 1994, pp. 71-73.
[Gol94b] Golshani, F. and Dimitrova, N., "Retrieval and Delivery of Information in Multimedia Database Systems", *Information and Software Technology*, vol. 36, no.4, June 1994, pp. 235-242.
[Dimi95] Dimitrova, N. and Golshani, F., "Motion Recovery for Video Content Classification," in *ACM Trans. on Office Information Sys.*, vol. 13, no. 4, Oct. 1995.

10. DYNAMIC ROUTE PLANNING

Forouzan Golshani

Like many other AI problems, this problem works with and depends on databases, in this case a database of maps and other geographic information systems. The Route Planner (sponsored by McDonnell Douglas) finds an optimal path through a varying danger zone by taking into account uncertain and incomplete information received from multiple sources. It uses the Dempster-Shafer theory of belief for management of uncertainty, in conjunction with a variation of the A* algorithm for path optimization. Based on the information on the area stored in the database, and by pooling the incomplete information about the weather, current danger, and other relevant parameters, the route planner generates the best route. It then continually re-examines the optimality of the route based on the new information, and re-calculates the optimized route in real-time until the destination is reached.

[Gols92] Golshani, F., Cortes-Rello, E., and Ahluwalia, S., "RUTA-100: A Dynamic Route Planner," *Proc. IEEE Int. Conf. on Tools with Artificial Intelligence*, Arlington, Nov. 1992, pp 418-423.
[Gols95] Golshani, F., Cortes-Rello, E., Howell, T., "Dynamic Route Planning with Uncertain Information," to appear in *Knowledge Based Systems*, Elsevier, 1995.

11. SCHEMA INTEGRATION IN HETEROGENEOUS DATABASE ENVIRONMENTS

Forouzan Golshani

This work has involved the development of a graphical tool for the automatic generation of a global schema from the schemas of a group of federated databases (sponsored by Bull Worldwide Information Systems). The tool converts each of the schemas into an object oriented form by deriving the dependency information and other relevant semantics. Each of the participating schemas is then represented in a graphical form. The next step is the gathering of the assertions that would lead to the identification of the corresponding syntactic and semantic values. Such relationships as object class correspondence and attribute correspondence are captured trivially in this process. The tool checks the collection of assertions for consistency and produces a global schema in graphical form. If necessary, the user can then make the necessary refinements. If desired, the global schema is transformed into a relational form. Since the tool is embedded in a distributed database management system, all the appropriate SQL commands are generated in order to create the integrated relational schema along with the necessary access paths. Contact F. Golshani for public domain literature on this project.

12. AUTOMATIC DISCOVERY OF DATA DEPENDENCIES

Forouzan Golshani

The key to a good relational database design is the thorough identification of the underlying data dependencies. However, defining the semantics of the data elements is not a trivial task. In addition, with the exception of some newer versions, commercial relational database management systems do not make any attempt to uphold the prescribed dependencies. This work has resulted in the development of an automated method for inferring functional dependencies from existing relational databases. Concomitantly, algorithms for discovering multivalued and join dependencies are also developed. Identification of the candidate keys of the relation is considered as a special case of the functional dependencies. The technique is based on the "learning from examples" paradigm of machine learning, and uses Armstrong's axioms as heuristics to guide the discovery process. The algorithms have been tested on an Oracle database.

[Gava95] Gavaskar, P. and Golshani, F., "An Intelligent Tool for Discovering Data Dependencies in Relational Databases", Chapter 8 in *Artificial Intelligence and Automation*, (N. Bourbakis, ed.) World Scientific, 1995, pp. 149-195.
[Gols95] Golshani, F. and Gavaskar, P., "Automatic Discovery of Data Dependencies," submitted to *IEEE Trans. on Knowledge and Data Eng.*, 1995.