Curriculum vitae

David Toman

March 2002

Department of Computer Science
University of Waterloo
200 University Avenue West
Waterloo, Ontario, Canada N2L 3G1

tel: (519) 888-4567 ext 4447
fax: (519) 885-1208
david@uwaterloo.ca
http://db.uwaterloo.ca/~david

Research Interests:

- Temporal databases and query languages (time-related issues in Database Management systems, in particular compact representation of histories, integrity constraints, transaction processing, applications to spatial databases, GIS, and CAD systems);
- Query processing in Embedded Control Systems (query compilation, data layout, schema information),
- Representation and query languages for semi-structured data (query processing for XML);
- Finite model theory, model checking, and their applications to database systems, and formal methods in software development;
- Efficient implementation techniques for (non-imperative) programming languages.

Education:

- PhD. in Computer Science, Department of Computing and Information Sciences, Kansas State University, Manhattan, KS (1996).
  Advisor: Jan Chomicki,
  Dissertation: Foundations of Temporal Query Languages.
- MS (MGr., includes BS) in Computer Science (summa cum laude), Department of Computer Science, Masaryk University Brno, Czechoslovakia, (1992), GPA 4.0.
  Advisor: Jiri Zlatuska,
  Thesis: Operational Semantics of Prolog, Semantics of WAM.

Employment:

Assistant Professor
Department of Computer Science, University of Waterloo, Ontario, Canada (1998–).

Visiting Professor
BRICS, University of Aarhus, Denmark (fall 1997, fall 1999, spring 2001).

NSERC/NATO Postdoctoral Fellow
Department of Computer Science, University of Toronto, Canada (1996-1998).

Teaching/Research Assistant
Department Computing and Information Sciences, Kansas State University (1992-1995).

Research Associate
Institute of Computer Science, Masaryk University, Czech Republic (1987-1992).

Publications:

Articles in Refereed Journals

1. Querying ATSQL Databases with Temporal Logic (with Jan Chomicki and Michael H. Bohlen). *ACM Transactions on Database Systems (TODS)*, Vol. 26, No. 2, 145-178, 2001.
2. Datalog with Integer Periodicity Constraints (with Jan Chomicki). *Journal of Logic Programming*, Vol. 35, No. 3, 263-290, 1998.
3. Memoing Evaluation for Constraint Extensions of Datalog. *Constraints: An International Journal, Spec. Issue on Constraints and Databases*, Vol. 2 No. 3/4, 1997.
4. Implementing Temporal Integrity Constraints Using an Active DBMS (with Jan Chomicki). *IEEE Transactions on Knowledge and Data Engineering, Special section on Temporal and Real-time Databases*, Vol. 7, No. 4, 1995.

Refereed Book Chapters

1. SQL/TP: a Temporal Extension of SQL. In *Constraint Databases*, Gabriel Kuper, Leonid Libkin, and Jan Paredaens, editors, chapter 19, 391–399, 2000.
2. Point-based Temporal Extensions of SQL and their Efficient Implementation. *Dagstuhl seminar on Temporal Databases*, Germany, 1997. In *Temporal Databases: Research and Practice*, Etzion, Jajodia, Sripada, eds., LNCS 1399, chapter 3.4, 211–237, 1998.
3. Temporal Logic in Information Systems (with Jan Chomicki). In *Logics for Databases and Information Systems* (Chomicki and Saake eds.), Kluwer Academic publishers, 31–70, 1998. Preliminary version in *BRICS Lecture Series LS-97-1*, 1997.

Articles in Refereed Conferences

1. Optimizing Temporal Queries: Efficient Handling of Duplicates (with Ivan T. Bowman). Proc. *TIME 2001, International Symposium on Temporal Representation and Reasoning*, Cividale del Friuli, Italy, 93-100, 2001.
2. Expiration of Historical Databases. Proc. *TIME 2001, International Symposium on Temporal Representation and Reasoning*, Cividale del Friuli, Italy, 128-135, 2001.
3. On Decidability and Complexity of Description Logics with Uniqueness Constraints (with Vitaliy L. Khizder and Grant E. Weddell). Proc. *ICDT'01, International Conference on Database Theory*, London, UK, LNCS 1973, 54-67, 2001.
4. Reasoning about Duplicate Elimination with Description Logic (with Vitaly L. Khizder Grant Weddel). Proc. *Computational Logic CL 2000 (DOOD)* London, UK, LNAI 1861, 1017-1032, 2000.

5. A Point-based Temporal Extension of SQL. Proc. *5th International Conference on Deductive and Object-Oriented Databases*, Montreux, Switzerland, LNCS 1340, 103-121, 1997. Full version TR-CS-362, 1997, University of Toronto.
6. Point vs. Interval Query Languages for Temporal Databases. Proc. *15th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, Montreal, PQ, 58-67, 1996.
7. First-Order Temporal Queries Inexpressible in Temporal Logic (with Damian Niwinski). Proc. *Advances in Database Technology EDBT'96*, Avignon, France, LNCS 1057, 307-324, 1996. Preliminary version in Workshop on Theory of Constraint Databases, PODS 1995 (TR UNL-CSE-95-08).
8. Querying TSQL2 Databases with Temporal Logic (with Michael H. Bohlen, Jan Chomicki, and Richard T. Snodgrass). Proc. *Advances in Database Technology EDBT'96*, Avignon, France, LNCS 1057, 325-341, 1996.
9. Top-Down beats Bottom-Up for Constraint Based Extensions of Datalog. Proc. *International Symposium on Logic Programming ILPS'95*, Portland, OR, 98-112, 1995.
10. Datalog with Integer Periodicity Constraints (with Jan Chomicki and David S. Rogers). Proc. *International Symposium on Logic Programming ILPS'94*, Ithaca, NY, 189-203, 1994.
11. The Register Allocation in WAM (with Ludek Matyska and Adriana Jergova). Proc. *8th International Conference on Logic Programming ICLP'91*, Paris, France, 142-156, 1991.

Other Refereed Articles

1. Fine Grained Information Integration with Description Logics. (with Huizhu Liu and Grant Weddel) Proc. *4nd Int. Workshop on Description Logics DL2002*, Toulouse, France, (to appear) 2002.
2. Query Processing in Embedded Control Programs. (with Grant Weddel) Proc *Databases in Telecommunications.* LNCS 2209, 68-87, 2001.
3. On Attributes, Roles, and Dependencies in Description Logics and the Ackermann Case of the Decision Problem. (with Grant Weddel) Proc. *3nd Int. Workshop on Description Logics DL2001*, Stanford, CEUR-WS, vol.49, 76-85, 2001.
4. On Decidability and Complexity of Description Logics with Uniqueness Constraints. (with Vitaly L. Khizder Grant Weddel) Proc. *2nd Int. Workshop on Description Logics DL2000*, Aachen, Germany, CEUR-WS, vol.33, 193-202, 2000.
5. Constraint Databases and Program Analysis using Abstract Interpretation. Proc. *2nd Int. Workshop on Constraint Databases Systems CDB'97*, Delphi, Greece, LNCS 1191 (Constraint Databases and Applications), 246-262, 1997.
6. Computing the Well-founded Semantics for Constraint Extensions of Datalog. Proc. *CP'96 workshop on Constraint Databases*, Cambridge, MA, LNCS 1191 (Constraint Databases and Applications), 64-79, 1996.
7. Implementing Temporal Integrity Constraints Using an Active DBMS (with Jan Chomicki). Proc. *RIDE'94 Active Database Systems*, Houston, TX, 87-95, 1994.

Other Publications

1. Foundations of Temporal Query Languages. PhD. Dissertation. Kansas State University, 1996.

2. WAM Based Efficient Implementation of Logic Programming Languages (with Ludek Matyska, Adriana Jergova, Petr Mejzlik, Vaclav Racansky, and Jiri Zlatuska). Final Report of IBM Academic Initiative Research Project, 1992.
3. Operational Semantics of Prolog, Semantics of WAM. MS Thesis, Faculty of Science, Masaryk University Brno, 1992.

In Preparation/Submitted

1. Optimizing Temporal Queries: Efficient Handling of Duplicates (with Ivan T. Bowman). Invited to special issue of DKE (Elsevier Science).
2. Variable Independence and Aggregation Closure (with Jan Chomicki, Dina Goldin, and Gabriel Kuper). revised manuscript under review in TKDE.
3. Querying XML: On the Utility of Interval Encoding (with Grant E. Weddell). TR-CS-2002-09, University of Waterloo.
4. Efficient Handling of Nested XML Queries using Relational Technology. (with Mariano P. Consens and David DeHaan). TR-CS-2002-12, University of Waterloo.
5. Logical Data Expiration. Invited chapter for *Logics for Emerging Applications of Databases.* (Chomicki, van der Meyden, and Saake, eds.)
6. Note on Online Decompilation of PROLOG Programs. BRICS 1999.

Grants, Fellowships, and Awards:

- CITO Project *Text Indexing for Efficient Query Processing in Data Warehouses.* (CDN$ 143,000), University of Waterloo 2001-2002 (with Ian J. Munro and Gaston Gonnet.)
- CFI/OIT *Software Technology for Embedded Control Programs: The Effective Evolution of Very Large Legacy Communication Software* (CDN$ 705,000 awarded August 2000). In cooperation with Nortel Networks. (with Grant Weddell, Ken Salem, Ric Holt, and Brendan Frey.)
- CFI/OIT New Opportunities *A Collaborative Environment for Algorithm Development, Simulation and Historical Data Warehousing* (CDN$ 750,000), University of Waterloo 2000 (with Brendan Frey, Dale Schuurmanns, Michael Godfrey, Michele Mosca, Ken Seng Tan, and Changbao Wu.)
- CITO Project *Database Management for Embedded Control Programs* (CDN$ 180,000), University of Waterloo 2000-2002 (with Grant Weddell, Ken Salem, and Charlie Clarke.)
- NSERC Individual Research Grant, Department of Computer Science, University of Waterloo (CDN$ 84,000) 1999-2003.
- NSERC/NATO Postdoctoral Fellowship, Department of Computer Science, University of Toronto (CDN$ 58,000) 1996-1998.
- DIMACS Summer school on Finite Model Theory, Proof Complexity, and Computer-Aided Verification, DIMACS, Rutgers University, NJ, 1995 (NSF fellowship).
- EDBT Summer School on Advances in Database Technology, Alghero, Italy, 1991 (EDBT fellowship).
- NATO International Summer School on Logic and Algebra of Specification, Marktoberdorf, Germany, 1991 (NATO fellowship).
- IBM Academic Initiative CSFR Project grant: Semantics of Logic Programming Languages (with Ludek Matyska and Jiri Zlatuska), 1990-1992.

Prototype Research Systems:

- Compiler for a temporal extension of SQL to serve as a pre-processor for a standard relational DBMS (in progress).
- An empirical comparison of performance of query evaluation algorithms for constraint deductive queries (Prolog Meta-interpreter, 1996).
- Compiler of temporal constraints to SQL (Starburst and INGRES DBMS, 1994-1995).
- Efficient implementation of logic programming languages (funded in part by the Ministry of Education of CSFR, 1989-1992; part of the IBM Academic Initiative in CSFR 1990-1992).
- WAM-based Prolog compiler for MS-DOS and UNIX (funded in part by the Ministry of Education of CSFR, 1987-1989).
- Multiuser Editor for RSX-11M with small footprint in memory (Macro-11, RSX-11M, PDP-11/34, 1987).

Invited Talks:

- Expiration of Data (keynote talk)
  – Int. Symp. on Temporal Representation and Reasoning, Manchester, UK, July 2002.
- Query Processing in XML
  – University at Buffalo, Buffalo, NY, October 2001.
- On Expiration in Historical Databases
  – McGill University, Montreal, PQ, August 2001,
  – Dagstuhl Seminar 00291 *Logics for Emerging Applications of Databases*, July 2000.
- Query Processing in Embedded Control Systems
  – BRICS, University of Aarhus, Denmark, December 1999,
  – BRICS, University of Aalborg, Denmark, November 1999.
- Temporal Databases (a tutorial; with Jan Chomicki)
  – *Int. Conf. on Advances in DB Technology, EDBT'98*, Valencia, Spain, March 1998,
  – *Int. Conf. on Temporal Logic, ICTL'97*, Manchester, UK, July 1997.
- Temporal Query Languages (interview talk)
  – Dalhousie University, Halifax, NS, Canada, April 1998,
  – Rensselaer Polytechnic Institute, Troy, NY, April 1998,
  – University of Waterloo, Waterloo, ON, Canada, March 1998,
  – Northeastern University, Boston, MA, March 1998,
  – University of Nebraska, Lincoln, NE, March 1998,
  – Lucent Bell Labs, Murray Hill, NJ, March 1998,
  – AT&T Research, Florham Park, NJ, March 1998,
  – University of Alberta, Edmonton, AB, Canada, February 1998,
  – York University, North York, ON, Canada, February 1998.
- Temporal Extensions of SQL
  – City University, London UK, July 1997,
  – Dagstuhl Seminar 9726 *Temporal Databases*, June 1997,
  – SUNY at Stony Brook, NY, March 1997, and
  – Lucent Bell Labs, NJ, March 1997.

- First-Order Temporal Queries Inexpressible in Temporal Logic
  – University of Birmingham, Birmingham UK, July 1997,
  – BRICS, University of Aarhus, November 1996, and
  – Memorial University of Newfoundland, NF, Canada, July 1996.
- Top-Down beats Bottom-Up for Constraint Based Extensions of Datalog
  – University of Toronto (interview talk), ON, Canada, June 1995.
- Datalog with Integer Periodicity Constraints.
  – University of Nebraska-Lincoln, NE, 1994.

Professional Activities:

- **Refereeing for journals:**
  ACM Transactions on Programming Languages and Systems (TOPLAS), ACM Transactions on Database Systems (TODS), Journal of Logic Programming (JLP), IEEE Transactions of Knowledge and Data Engineering (TKDE), Information and Computation, Information Processing Letters, Computational Intelligence, International Journal of Foundations of Computer Science, Journal of the Interest Group in Pure and Applied Logics (IGPL).
- **Refereeing for conferences:**
  IEEE Symposium on Foundations of Computer Science (FOCS), ACM Conference on Management of Data (SIGMOD), ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation (PEPM), International Conference on Logic for Programming and Automated Reasoning (LPAR), IEEE International Conference on Data Engineering (ICDE, PC member), International Conference on Very Large Databases (VLDB), International Conference on Temporal Logic (ICTL), and International Conference on Principles and Practice of Constraint Programming (CP), International Workshop on Database Programming Languages (DBPL, PC member), Workshop on Foundations of Models and Languages for Data and Objects (DEMM).
- **Memberships in Professional Organizations:**
  Association for Logic Programming, Association for Computing Machinery, Czech Computer Science Society (CSIS, founding member), Upsilon Pi Epsilon Computer Science Honorary Society.

References:

Available upon request.

Teaching Statement

David Toman

Teaching Experience:

Assistant Professor
Department of Computer Science, University of Waterloo, Canada (1998-).

- CS234 Algorithms Data Structures (sophomore class; Fall 2000, Fall 2001),
- CS448 Introduction to Database Systems (senior class; Winter 1999, Spring 2000 (x2), Spring 2002 (x2)),
- CS748M Topics in Database Systems (graduate seminar; Fall 1998),
- CS748M Topics in Database Systems (graduate seminar; Fall 2000)
  Publications: Optimizing Temporal Queries: Efficient Handling of Duplicates (with Ivan T. Bowman). Proc. *TIME 2001, International Symposium on Temporal Representation and Reasoning*, Cividale del Friuli, Italy, 93-100, 2001. Invited to special issue of DKE (Elsevier Science).
- CS740M Database Engineering (tier II graduate class, Fall 2001)
  Publications: Efficient Handling of Nested XML Queries using Relational Technology. (with Mariano P. Consens and David DeHaan). TR-CS-2002-12, University of Waterloo.

Visiting Professor
DAIMI, Department of Computer Science, University of Aarhus, Denmark (Fall 1997, Fall 1999, Spring 2001).

During my visits to BRICS, Centre for Basic Research in Computer Science, I taught the following classes at BRICS PhD School and DAIMI:

- Temporal Databases (BRICS PhD Mini-course). BRICS Newsletter 7, 3-4, 1997.
  Publications: Temporal Logic in Information Systems (with Jan Chomicki). In *Logics for Databases and Information Systems* (Chomicki and Saake eds.), Kluwer Academic publishers, 31-70, 1998.
- Introduction to Database Systems (senior course).

The introductory database class also offered the students the possibility of becoming a certified *IBM DB2 Application Developer*.

Teaching Assistant
Department Computing and Information Sciences, Kansas State University (1993-1995).

Duties associated with the assistantship included grading papers and examinations (in most cases I was solely responsible for the grading of the class) and occasional lecturing.

Teaching Philosophy:

I believe that teaching and supervising students is an integral part of academic life. Moreover, supervising students and, in general, interaction with the students often leads to exploring new and unforeseen topics and research directions.

While training students to be good programmers is an important aspect of every computer science curriculum, I believe that a computer science and computer engineering degrees need to offer more than mere training of programmers: they must promote the *understanding* of principles on which computing is based, many of which are deeply rooted in *mathematics*. This way the graduates should be able to understand not only the immediate functions and features of their programs but also the long-term ramifications of their design decisions and tradeoffs that they will be often making after they get into the "real world".

It is also my belief that while the practical up-to-date skills, hands-on experience, and familiarity with the cutting-edge technology play an important and irreplaceable role in the computer science and engineering curriculums and may give graduates immediate chance to get a job (a common measure of the quality of a degree), the thorough understanding of the foundations of computing may help them to *keep* their jobs later, when the technology moves on and the skills tied to particular tools of today cease to be relevant.

Teaching Interests:

I would feel comfortable teaching most introductory courses in computer science and a variety of advanced courses, including but not limited to:

- **Undergraduate level:**
  Introduction to Database Systems, Design and Implementation of Database Systems, Programming Languages, Formal Methods in Software Engineering, Compiler Construction, Operating Systems, Theory of Formal Languages and Automata, Introduction to Non-imperative Programming Languages, Introduction to Complexity Theory, Set Theory and Logic.
- **Graduate level:**
  Advanced Implementation Techniques for Database Systems, Theory of Database Systems, Distributed Databases and Concurrency Control, Semantics of Programming Languages, Formal Methods in Computer Science and Software Engineering, Advanced Compiler Design.

March 2002

Research Statement

David Toman

Relational database systems enjoy wide-spread acceptance and provide the basis for many every day applications. Their success is based on two integral properties of relational technology:

- sound and well-understood *foundations*, and
- efficient *implementation techniques* suitable for large amounts of data.

On the other hand, there are many limitations of the classical relational model and its implementations, in particular

- its inability to deal elegantly with interpreted data, e.g., time and space; and
- its close ties with large, monolithic DBMS implementations that cannot be used in embedded systems;

My research focuses on extensions of the relational model that alleviate the shortcomings.

Research Contributions (Temporal Data):

While virtually every database application has to handle time in one form or another (and usually does so using ad-hoc techniques), past research in this area failed to produce a widely accepted and available solution. This failure can be attributed to the use of complicated approaches without solid theoretical foundations. My research proposes an alternative way of adding temporal capabilities to a relational system. The main contributions in this area are:

- A separation (with respect to expressive power) of first-order temporal logic from two-sorted first-order logic (independently Abiteboul et al., PODS'96). Therefore, first-order temporal queries cannot be composed of simpler subqueries over the universe of single- (or fixed-) dimensional temporal relations; a common choice in most of the current proposals for temporal extensions of the relational model and/or SQL.
- A definition of a sound and natural temporal extension of SQL including an efficient query evaluation procedure over a compactly encoded temporal databases. The proposal provides both a clean and sound theoretical foundation and a well-defined implementation path.

The work on a compact representation of large and potentially infinite sets of time instants in temporal databases connects my research with the area of Constraint Databases. I have studied a representation that uses periodic sets in conjunction with order constraints and efficient query evaluation procedures for *deductive queries* over such a representation. I have built an experimental prototype of a query evaluation engine to measure the relative performance of the individual query evaluation procedures.

The results obtained for temporal databases feed directly into the exploration of *expiration* and aging of data in such systems. I have developed a new technique for *query-driven* expiration of database histories that, for an arbitrary fixed set o first-order (SQL) queries, compacts the existing history in such a way that the answers to the queries are preserved while the size of the result is independent of the length of the history (it only depends on the size of the *active domain* of the history; this was conjectured not to be possible. On the other hand I have shown that thete cannot be a expiration technique with the above property for queries with counting (here, the space bound depends logarithmically on the length of the history).

The work on expiration of temporal data is closely related with efficient enforcement of temporal integrity constraints over histories of standard relational databases. I have built a prototype implementation of a temporal integrity enforcement method based on ECA rules commonly present in active database systems. The implementation is based on the *active features* of the INGRES and IBM's Starburst Database Management Systems.

Research Contributions (Database Technology for Embedded Systems):

An embedded control program can be often viewed as a main-memory database system tailored to suit the needs of a particular application. For performance reasons, the program will usually define concrete low level data structures to encode the database, which in turn must be understood by anyone who needs to develop or modify the program. This is in contrast with the data independence that can be achieved by using a database system. However, because of space and performance requirements, the use of current database technology is not likely to be feasible in this setting. To explore one obstacle to this, we have developed a query optimizer that compiles queries on a conceptual schema to native Java or C code that navigates low level data structures. Of crucial significance is that an arbitrary collection of such structures, perhaps already devised for an earlier version of the control program, can be given as a part of the input to the optimizer.

The main contributions in this area are:

- A query optimizer that, with the help of *schematic information* that describes main-memory data structures and/or interfaces of a embedded control system, transforms a high-level declarative query (e.g., in SQL) to C or JAVA code that can be executed within the embedded system. The DEMO system based on this approach has been verified by experimental means on top of the Linux operating system.
- The development of a *schema definition language* based on *description logic* that is used by the DEMO system, study of its computational properties (in particular, the relationship between various dialects of description logics and formulas with the *Ackermann quantifier prefix*), and its use for query optimization.

While the embedded control programs themselves provide a rich application area, the DEMO system can be easily adopted to other uses, e.g., to handling network management or to processing XML documents. (the work on the DEMO project this is joint work with Grant E. Weddell at the University of Waterloo; the DEMO project is funded by NORTEL Networks and NSERC, CITO, and CFI/OIT grants).

Future Research Plans

My current research plans are closely tied to the above results, in particular to resolving several open questions in the area of temporal data:

- what are the tradeoffs between expressive power of a temporal query language and the amount of data needed to answer a fixed set of queries over a database history?
- how is the above situation affected by retroactive updates?

- is there a relationship with approaches used for *garbage collection* in programming languages (or vice versa)?
- can *schematic information* be used to reduce the amount of data to remember even further?

and in the area of embedded systems:

- what data structures can be described by a schema language? (e.g., can the DEMO system be extended to handling *inductive data types*)
- what are the limits on expressive power of the queries?
- what is the impact of ordering of results? (this is particularly pertinent to handling XML queries).

In all cases I am more interested in the foundational principles of the problems and in the development of techniques to support and promote correct implementation, rather than in quick-n-dirty solutions.

March 2002