A. Gal (editor):

Temporal Databases

S. Jagodia, S. Sripada, O. Etzion (organizers)

Dagstuhl-Seminar-Report; ??
23.06.-27.06.97 (9726)

Contents

FILL IN THE DAGSTUHL INFORMATION

Temporal Databases

Organizers: S. Jagodia, S. Sripada, O. Etzion

Temporal databases incorporate the concept of time to create high-level abstractions useful in database applications. This has been an active area of research for about twenty years. In the last few years the importance of the temporal database area has been recognized by the international scientific community. This recognition came in part in the form of the ARPA/NSF sponsored International Workshop on Temporal Database Infrastructure in 1993, a VLDB-affiliated temporal workshop in 1995, a special section of the IEEE Transactions on Knowledge and Data Engineering on temporal and real time databases published in August 1995, and the incorporation of temporal constructs, proposed by the temporal database community, in the soon-to-be standardized SQL3 language.

This report arose out of the Dagstuhl seminar that was organized by us during June 23–27, 1997. This seminar focused on the future directions of this discipline, with respect to both research issues and the means to incorporate temporal databases into main-stream application development. List of topics discussed at this seminar included:

1. Temporal data models: relational, object-oriented, deductive and hybrid models. Where do the temporal capabilities fit in?

2. Temporal languages: TSQL2 and beyond. Update and retrieval languages for various types of temporal data models.

3. The inter-relationships between temporal databases and other disciplines: spatial databases, active databases, deductive databases, real-time databases, information uncertainty, belief revision, etc.

4. Implementation issues in temporal databases. Issues that arise from experience of implementors and users and the agenda for research into these areas and transition to use in practice.

5. Strategic discussions about the future of "temporal databases" as a discipline. Evaluation of the current state of the art and "call for action" to the community.

The Dagstuhl seminar brought together researchers that have dealt with different perspectives of temporal databases: temporal data models, temporal retrieval and update languages, inter-relationships between temporal databases and other database technologies (e.g., spatial databases, active databases, real-time databases), and inter-relationships between temporal databases and temporal reasoning in artificial intelligence. Some of the invited participants have also been involved in the standardization activities of the temporal community. Having a diverse group that shared a focus on temporal information processing ensured critical evaluation of the activities that have occurred thus far, and enriched the discussions.

As any Dagstuhl seminar, the participants respresented a select group of prominent researchers in the subject area. We also solicited submission for a book, aimed to include high-quality original papers about the state-of-the-art in the temporal database area. The book entitled "Temporal databases - research and practice" will be published by Springer-Verlag.

Temporal Databases with Multiple Granularities

Claudio Bettini
Universita di Milano
Dipartimento di Scienze Informazione
Via Comelico 39, I-20135 Milano, Italy
bettini@dsi.unimi.it
Sushil Jajodia, X. Sean Wang
George Mason University
Information & Software Systems Eng. Dept.
Mail Stop 4A4, 4400 University Drive, Fairfax VA 22030 - 4444, USA
jajodia@gmu.edu, xywang@isse.gmu.edu

There is a rich variety of time granularities, and users as well as applications often require the flexibility of viewing the same temporal data in terms of different time granularities. The increased awareness of this requirement by the database community is evident from the growing research literature on this subject and from the inclusion of constructs for handling multiple granularities in SQL-92, the relational query language standard, and TSQL2, a temporal extension of SQL-92. In spite of this, the fact remains that whenever there is a difference between the way the information is stored in the database and the way it is required by the users, it is up to the users to understand these differences and specify appropriate operations in their queries to reconcile them.

The objective of this project is to provide a flexible framework for supporting multiple time granularities. The targeted use of the framework is for automatic evaluation of user queries and for discovering temporal patterns (i.e., data mining) in an environment where either the user queries or temporal patterns involve granularities that do not match the granularity of the stored data. The basic idea is to add the necessary functionalities so that the database system is able to understand and reason about information involving multiple time granularities. Algorithms for efficiently evaluating user queries and discovering temporal patterns are also being investigated, and an experimental prototype is being built.

We are also investigating networks having temporal constraints on the distances among event occurrences. When multiple granularities are used in the distance specification, new techniques and algorithms for consistency checking and for deriving solutions are required. We are considering applying

the results that we have obtained in this area to database integrity constraint checking and to trigger condition evaluation in active databases.
Selected Publications:

- X. S. Wang, C. Bettini, A. Brodsky, and S. Jajodia, "Logical design for temporal databases with multiple granularities," ACM Trans. on Database Systems, Vol. 22, No. 2, June 1997, pages 115–170.
- C. Bettini, X. S. Wang, S. Jajodia, and J. Lin, "Discovering temporal relationships with multiple granularities in time sequences," IEEE Trans. on Knowledge and Data Engineering, To appear. A preliminary version appeared as "Testing complex temporal relationships involving multiple granularities and its application to data mining," Proc. 15th ACM PODS Symp., Montreal, Canada, June 1996, pages 68-78.
- C. Bettini, X. S. Wang, and S. Jajodia, "Temporal semantic assumptions and their use in database query evaluation," IEEE Trans. on Knowledge and Data Engineering, To appear. A preliminary version appeared in Proc. ACM SIGMOD Int'l. Conf. on Management of Data, San Jose, CA, May 1995, pages 257–268.
- C. Bettini, X. S. Wang, and S. Jajodia, "A general framework for time granularity and its application to temporal reasoning," Annals of Mathematics and Artificial Intelligence, To appear.
- Claudio Bettini, X. Sean Wang, Sushil Jajodia, "Satisfiability of quantitative temporal constraints with multiple granularities," Proc. 3rd I'ntl. Conf. on Principles and Practice of Constraint Programming, Schloss Hagenberg, Austria, October-November 1997, To appear.
- C. Bettini, X. S. Wang, and S. Jajodia, "A general framework and reasoning models for time granularity," Proc. 3rd Int'l. Workshop on Temporal Representation and Reasoning, Key West, FL, May 1996.
- X. S. Wang, S. Jajodia, V. S. Subrahmanian, "Temporal Modules: An Approach Toward Federated Temporal Databases," Information Sciences, Vol. 82, 1995, pages 103–128. A preliminary version appeared in Proc. ACM SIGMOD Int'l. Conf. on Management of Data, Washington, DC, May 1993, pages 227–236.

Logical Foundations of Temporal Databases

Renato Busatto
Database Systems Research Group
Department of Computer Science
Aalborg University
Denmark

My research activities are mostly centered on the logical foundations of temporal databases, with particular emphasis on temporal data models, ontologies of time and dynamic timestamping.

Temporal data models: formal definition of temporal data models, both at the data structure and algebraic levels, and the application of such conceptual framework to the study of properties of temporal query languages and to temporal database design.

Ontologies of time: axiomatic definition of abstract data types for time points and time intervals, and their application to temporal database design.

Dynamic timestamps: formal semantics of dynamic timestamps, i.e. those whose denotation varies with time, such as CURRENT and NOW; identification of various axiomatic systems that capture the subtleties of the meaning of such timestamps in natural language; study of the computational complexity associated with each of such formal representations; and impact of each approach on temporal database design.

Spatiotemporal Databases and the Expressiveness of Temporal Query Languages

Jan Chomicki
Department of Computer Science
Monmouth University
Howard Hall, West Long Branch NJ 07764-1898, USA
chomicki@moncol.monmouth.edu

SPATIOTEMPORAL DATABASES: The technology of constraint databases has a lot to offer in this area because it is based on a sound mathematical model. I am interested in studying the theoretical foundations of spatiotemporal databases and the more practical issues in the interoperation of temporal, spatial and spatiotemporal database models:

1. J. Chomicki, P. Revesz "Constraint-Based Interoperability of Spatiotemporal Databases," Proc. 5th International Symposium on Large Spatial Databases, July 1997, Berlin, Germany, pp. 142-161.
2. J. Chomicki, D. Goldin, G. Kuper "Variable Independence and Aggregation Closure," Proc. 15th ACM Symposium on Principles of Database Systems, June 1996, Montreal, Canada, pp. 40-48.

FOUNDATIONS OF TEMPORAL DATABASES: The project of putting the semantics of temporal databases and query languages on a firm theoretical foundation is now largely complete. What remains to be done is resolving the remaining questions about the expressiveness or complexity of specific query languages and apply the formal framework in practice:

1. J. Chomicki, M. Boehlen, R. Snodgrass and D. Toman "Querying TSQL2 Databases with Temporal Logic," LNCS 1057, pp. 325-341, March 1996, Springer-Verlag.
2. A short paper based on my Dagstuhl presentation "'Temporal' Considered Harmful in Temporal Database Design" is an instance of work drawing on the above-mentioned formal framework. I try to show in it that "less" (fewer concepts, definitions etc.) is often better than "more."

Activities in Temporal Databases

Hugh Darwen
IBM Ltd.
HCI United Kingdom Strategy & Design
P.O. Box 31, Warwick CV34 5JL, Great Britain
darwen@vnet.ibm.com

My interest in temporal database was aroused in 1994, when the first moves were made towards inclusion of temporal extensions in the international standard for SQL.

I have focussed almost exclusively on the relational aspects of the subject, being primarily interested in the provision, in relational database languages, of a general-purpose collection of operators on intervals. I see such provision as being a key enabler of temporal and other applications, including generic applications such as "time series", "valid time history" and "transaction time history".

I have devoted much effort to the discovery and promulgation of those research results that claim to show that solutions to all of the problems we encounter in temporal databases are expressible in first order logic, hence in primitive relational algebra or calculus (I admit recursion, too, if necessary), hence in current database languages based on such theory. If these claims are true, then every desired new operator in, for example, SQL can be specified by *syntactic substitution*, an approach to growth that is found generally agreeable in the language design community.

In the past two years I have collaborated with J.M. Sykes in the United Kingdom on major contributions to "SQL/Temporal", an extension to the SQL standard that is in the early stages of drafting. These include operators for "coalescing" tables over interval columns, and "temporal difference", the syntax being based closely on IXSQL (Lorentzos et al., 1994). These operators, and the new scalar operators on intervals based on Allen's algebra, are all specified by syntactic substitution.

Transaction-time World-Wide Web (WWW) Server

Curtis Dyreson
James Cook University
Dept. of Computer Science
Townsville Q4811, Australia
curtis@cs.jcu.edu.au

The goal of this research is to add transaction time and valid time support to SQL-like WWW query languages (e.g., WebSQL). At DagStuhl we presented the design of a 'transaction-time' World-Wide Web (WWW) server.

Unlike traditional database servers, WWW servers have no update protocol for records (pages) that they service. Users insert, update, and delete pages without notifying the server; the server is only used to retrieve pages. We outlined a 'lazy' update protocol that enables a server to detect, in most cases, that a page has been modified. The server can then archive the outdated page. Rather than storing the entire outdated page, only a 'diff' file is stored since most page edits are small.

We also presented strategies for supporting transaction timeslice in a WWW server, via URL modifications and cgi-bin scripts. A key benefit of this research is that it provides transaction-time support as a transparent extension of existing servers; no changes to legacy pages, HTTP, or HTML is required.

We are currently working on implementing the protocol and adding valid time support to an SQL-like WWW query language.

Design and Implementation Options for Incorporating Time Series in Temporal Databases

Ramez Elmasri
Department of Computer Science and Engineering
The University of Texas at Arlington
Box 19015, Arlington, Texas 76016, USA
email: elmasri@cse.uta.edu

My research interests in temporal databases are currently focused in the following areas:

1. Integrating time series and version-based temporal models: Most temporal database models concentrate on version-based temporal data, and hence describe how an object's properties change over time. However, in economic and financial applications, temporal data that are represented as time series are prevalent, where values that describe certain events in sequence are stored. These two types of temporal data have different characteristics, and different types of queries. Our research in this area concerns developing data models and languages that allow both types of temporal data to be represented and accessed within a unified framework.

2. Comparison of temporal data models' representation power: The power of representing temporal information is limited depending upon which time dimensions are represented in a temporal data model - valid time only, transaction time only, or bitemporal. We compare the representational and querying capabilities of various temporal models, and develop a pure append only model that allows retroactive and proactive queries and updates, and differentiates between corrections and changes in plans.

3. Indexing techniques for temporal data: We develop indexing techniques and search strategies that allow rapid retrieval based on temporal search conditions. Our techniques are developed for valid time, transaction time, and bitemporal databases. Different indexing structures are developed for each type of temporal databases to account for their differences.

Dimensional Data

Shashi K. Gadia
Computer Science Department
Iowa State University
Ames, IA 50011
gadia@cs.iastate.edu

Major Interests: Temporal, spatial, belief, security, statistical and incomplete data; database models, type hierarchy, languages, user interfaces, optimization, implementation and access methods; pattern matching in spatio-temporal data.

Current Research: The term dimensional data covers databases for data over a dimensional space such as time, space and user beliefs. A uniform approach for dimensional data assures upward compatibility. Thus when one migrates from lower forms of dimensional data to higher forms, e.g. ordinary data to spatio-temporal data, the existing application software would not need to be redeveloped. Such a framework necessitates careful study of pragmatic issues such as database optimization.

Temporal Active Databases

Avigdor Gal
MSIS Department
Rutgers University
94 Rockafeller Road, Piscataway, NJ 08854-8054
gal@everest.rutgers.edu
Opher Etzion
IBM Haifa Research Lab and
The Faculty of Industrial Engineering & Management, Technion - Israel Institute of Technology
Haifa, Israel
OPHER@HAIFA.VNET.IBM.COM

Conventional databases are like children, in a sense. Mostly leaving the here and now, many things are given as ground facts, there is a clear distinction between right and wrong, between true and false, and there is always an answer to a question. As we grow up, at least some of us realize that the world is not a fairy-tale. We start thinking ahead, we understand some of the rules of life that get us going, we understand that there are many shapes of gray between black and white, and we even learn how to say: "I dont know" or "it might be either x or y," without feeling really foolish.

TALE, a Temporal Active Language and Execution model is part of the tale of databases growing up, and hopefully not a fairy-tale. The combination of temporal and active database technologies creates a powerful modeling tool that is beyond the basic capabilities of these two technologies. TALE consists of a language for a temporal active database and the execution process of it. The properties of the model, including expressive power, consistency, update minimality and parallelism are discussed in Avigdor's thesis, available through his home page (http://everest.rutgers.edu/~gal/publications.html).We have devised a language of temporal composite events that include long-duration activities. Each event can have a "valid time", which designates when the event happened, and a "transaction time" which designates when the event has been reported. Compositions can be based on both time dimensions.

Another branch of research is the issue of schema versioning, by which a rule can relate to any version of the schema, and a rule that is applied over an interval can be applied to several schema versions during this interval. A parallel processing using "temporal agents" was applied to optimize multiple versions. The "temporal stabilizers" project deals with the automatic creation of repairs to integrity constraints with the added complexity of temporal database.

All the rules components (events, conditions and actions) may have different valid times with partial overlappings. In current project we build the semantics of simultaneous valid rules, and the implementation of alternative database values that are applicable simultaneously.

Temporal Object Models

Iqbal A. Goralwalla and M. Tamer Ozsu
Laboratory of Database Management Systems, Department of Computing Science
University of Alberta
Edmonton, Alberta T6G 2H1, Canada
{iqbal,ozsu}@cs.ualberta.ca

The relational and object-oriented approaches to modeling temporal information have led to the definition and design of a multitude of temporal models. Many of these assume a specific set of temporal features, and therefore do not incorporate sufficient functionality or extensibility to meet the varying temporal requirements of today's applications. Instead, similar functionality is re-engineered every time a temporal model is created for a new application. This suggests a need for combining the diverse features of time under a single infrastructure that allows design reuse. Towards this end, we present an object-oriented framework that provides such a unified infrastructure. An object-oriented approach allows us to capture the complex semantics of time by representing it as a basic entity. Furthermore, the typing and inheritance mechanisms of object-oriented systems directly enable the various notions of time to be reflected in a single framework. The objectives of this work are threefold. The first objective is to identify the design dimensions that span the design space for temporal models. This will classify design alternatives for temporal models. The design space is then represented by exploiting object-oriented features to model the different aspects of time. The second objective is to show how the temporal framework can be tailored to accommodate the temporal needs of different applications, and derive existing temporal models by making a series of design decisions through subclass specialization. The framework can also be used to derive a series of new more general temporal models that meet the needs of a growing number of emerging applications. The third objective is to use the framework to compare and analyze different temporal object models with respect to the design dimensions. This helps identify the strengths and weaknesses of the different models according to the temporal features they support. The framework is currently being implemented in C++ in the context of the ObjectStore object-oriented database management system.

Another avenue we are pursuing is using temporality to model other object database features such as schema evolution. The issues of schema evolution and temporal object models are generally considered to be orthogonal and are handled independently. This is unrealistic because to properly model applications that need incremental design and experimentation (such as CAD, software design process), the evolutionary histories of the schema objects should be traceable. We propose a method for managing schema changes by exploiting the functionality of a temporal object model. The result is a uniform treatment of schema evolution and temporal support for many object database management systems applications that require both.

Extending Temporal Database Concepts to the World Wide Web

Fabio Grandi and Maria Rita Scalas
C.S.I.TE. - C.N.R. and
Dipartimento di Elettronica - Informatica e Sistemica
Viale Risorgimento 2, I-40136 Bologna, Italy
{fgrandi, mrscalas}@deis.unibo.it

The development of the World Wide Web technology on the Internet is a major achievement of computer research in recent years. The great availability of easy-to-access on-line hypermedial documents has been a real revolution in the information world, also for its important social, cultural and economical consequences. However, a scarce attention has been devoted so far to the temporal aspects of the World Wide Web definition and technology. Moreover, although the management of time-varying information is a consolidated research issue in the database field, no integrations between the two worlds has been attempted yet.

In our latest research project, we try to make a first step towards a temporal extension of the World Wide Web, by considering the impact of well-known temporal database concepts on the design and management of Web documents, starting from the very first notions of transaction time and valid time. By means of a reference application (a virtual Web museum of fine arts), we will try to explore the potentialities of the approach and sketch research directions in which new solutions have to be sought.

In particular, the integration of transaction time into the World Wide Web would allow, in a transparent way, the management of successive versions of the same documents. For instance, by acting on transaction time within a "virtual museum," it will be possible to visit different temporary special exhibitions dressed in different times at the museum (i.e. to navigate through successive museum's versions).

On the other hand, the adoption of valid-time versioning would allow the explicit definition of temporal information within documents, whose contents will be selectively indexed on the basis of the validity coded within them. By acting on valid time, it will be possible to navigate through time in a given environment; for instance, to visualize the evolution of an archaeological site through successive ages, or to cut personalized visit routes for a specific epoch in a museum.

Temporal and Real-Time Issues in Active Database Applications

Dr. Heinrich Jasper
Hochschule Vechta
Institut für Umweltwissenschaften
Postfach 15 53, D-49364 Vechta, Germany
Heinrich.Jasper@uni-vechta.de

Active databases support events, i.e. notifications about changes in the database or the external application. Temporal information is always associated with this notion of event. Every detected event has subsidiary information on the point of time it occurred. Additionally, events describing time-stamps or time intervals (so-called time events) can be specified. Besides other application areas, several of our experiments demonstrated the usefulness of active capabilities in the area of computer based process support. For example, information systems for workflow management make extensively use of database technology for storing all kind of data and additionally must respond to events that occur in the workflow in a timely manner.

Database technology with provision for temporal and real-time support is necessary in the aforementioned application domains. Temporal information is associated with the events that set up processes. Furthermore, timely reactions are necessary in process domains, i.e. the respective applications have some real-time characteristics. Temporal and real-time aspects may be interrelated, for example due dates (for real-time reactions) are computed from temporal data in the database combined with temporal data of some event that occurred within the actual process. The complete control flow as well as some temporal aspects of processes can be specified within the event-condition-action (ECA) rules of active database languages. Furthermore, the event detection and action execution mechanism supports the execution of processes. However, there is a severe lack in database technology regarding the integration of temporal and real-time capabilities that are necessary to support the aforementioned application domains. Topics are for example the use of temporal data for scheduling of real-time actions and real-time behaviour under heavy transaction loads. One major observation is the need for joined work on temporal and real-time capabilities of databases that are used in process oriented domains.

TimeCenter and CHOROCHRONOS

Christian Jensen
Aalborg University
Dept. of Comp. Science
Fredrik Bajers Vej 7E, DK-9220 Aalborg Ost, Denmark
csj@cs.auc.dk

The bulk of my current research in temporal databases falls within the projects "TimeCenter" and "CHOROCHRONOS." The latter project is an umbrella project for more specific projects, including the projects "TimeDesign" and "Foundations of Technology-transferable Database Technology."

TimeCenter: Database applications that manage time-varying data, as most do, may benefit substantially from built-in temporal support in the database technology used. In spite of this and although temporal databases has been an active area of research for more than a decade, temporal database technology has so far had little impact on practice. The facts that much of past research ignored industry standards, assumed that a temporal DBMS must be implemented from scratch, and was incomplete in its coverage provide some of the reasons for this state of affairs. In TimeCenter, I collaborate with colleagues to establish the foundations for developing database technology that is technology-transfer friendly. The outset is a novel set of requirements that explicitly take technology transferability into account. Based on these, we aim to develop database concepts and techniques fundamental to the development of temporally enhanced database technology. The specific objective of TimeDesign is to develop concepts and techniques that support the modeling of time-dependent information.

CHOROCHRONOS: a research network devoted to the design, implementation, and application of database technology for the handling of spatiotemporal information. In CHOROCHRONOS, I collaborate with researchers from 10 European research groups. My focus is on conceptual modeling, query languages, and query processing.

References:

- TimeCenter Home Page.
  URL: http://www.cs.auc.dk/general/DBS/tdb/TimeCenter/
- CHOROCHRONOS Home Page.
  URL: http://www.dbnet.ece.ntua.gr/projects/chorochronos/chorochronos.html

ORES

Nikos Lorentzos
Informatics Laboratory, Agricultural University of Athens
Iera Odos 75, GR 118 55 Athens, Greece
lorentzos@auadec.aua.ariadne-t.gr

I had the honour to be one of the people invited at the Dagstuhl Seminar on Temporal Databases, 23-27 June, 1997. I was given the opportunity to make

two presentations:

1. 'ORES: Design and Implementation of a Temporal DBMS', concerning the development of a temporal relational DBMS and an associated language, VT-SQL.

2. 'Remarks on the TSQL2 Approach', where three questions were addressed: (i) whether TSQL2 achieves a uniform temporal data modelling, (ii) Whether TSQL2 is really dedicated to the management of temporal data, and (iii) whether 'implicit' columns (inherent in the TSQL2 approach) are really needed. It was shown that the answer to all these questions was negative. It was also shown that TSQL2 provides only partial support to the so-called 'sequence' queries. Moreover, an alternative more powerful approach for such queries was presented, making use of only 'explicit' (i.e. ordinary) attributes. The presentation aimed at initiating a discussion in the hope that this author's conclusions would be disproved. Unexpectedly, the feedback obtained was rather positive. The validity of the conclusions was also supported indirectly by the fact that other research efforts were also presented, which do make use of only explicit attributes.

The seminar enabled to become aware fairly quickly with the research effort of many specialists in various areas of temporal databases. Many thanks are due to the organisers of the seminar, O. Etzion, S. Sripada and S. Jajodia, they really did an excellent job. In the Seminar's closing speech, Sury combined successfully the objectives of the Seminar with the Theory of Relativity and gave all us a good lesson regarding fruitful collaboration and dissemination of research results in the area. The Centre's personnel was outstandingly helpful and the facilities we were provided with were beyond our expectations.

Modal Queries about Partially Ordered Events in the Event Calculus

Angelo Montanari (joint work with Iliano Cervesato and Massimo Franceschet)
Dipartimento di Matematica e Informatica, Universita' di Udine;
Via delle Scienze 206, 33100 Udine. Italia
montana@dimi.uniud.it

We consider a hierarchy of modal event calculi to represent and reason about partially ordered events. These calculi are based on the model of time and change of Kowalski and Sergot's Event Calculus (EC): given a set of event occurrences, EC allows the derivation of the maximal validity intervals (MVIs) over which properties initiated or terminated by those events hold. The formalisms we analyze extend EC with operators from modal logic. They range from the basic Modal Event Calculus (MEC), that computes the set of all current MVIs (MVIs computed by EC) as well as the sets of MVIs that are true in some/every refinement of the current partial ordering of events (Diamond/Box-MVIs), to the Generalized Modal Event Calculus (GMEC), that extends MEC by allowing a free mix of boolean connectives and modal operators. We analyze and compare the expressive power and the complexity of the proposed calculi, focusing on intermediate systems between MEC and GMEC, called ICMEC and ECMEC, respectively. The queries of ECMEC (Modal Event Calculus with External Connectives) allow combining computations of MVIs, Box-MVIs and Diamond-MVIs by means of boolean connectives. The approach followed in ICMEC (Modal Event Calculus with Internal Connectives) is dual: boolean combinations of MVI computations can be prefixed by either Box or Diamond. Both ECMEC and ICMEC retain enough of the expressive power of GMEC to allow a faithful and usable representation of numerous common situations. However, while the problem of evaluating an ICMEC query is NP-hard (as for GMEC), ECMEC admits polynomial implementations, making this formalism an ideal candidate for the formalization of a number of applicable domains.

Log-structured History data Access Method (LHAM)

Peter Muth
Universität des Saarlandes
Fachbereich 14 - Informatik
Postfach 15 11 50, 66041 Saarbrücken, Germany
muth@cs.uni-sb.de

There are numerous applications that require on-line access to a history of business events. Ideally, both historical and current data should be logically integrated into some form of temporal database, also known as a multi-version database, historical database, or rollback database. The underlying access method should support the migration of old record versions onto inexpensive write-once media, different forms of "time-travel" queries, and potentially high rates of update events. In my talk, I proposed a new access method for history data, called the . The basic principle of LHAM is to partition the data into successive components based on the timestamps of the record versions, and to employ a rolling merge process for efficient data migration between components. Preliminary experimental performance results of LHAM are very promising. The number of I/Os required to insert and update data closely matches the analytical expectations. Compared to a TSB-tree, LHAM requires much less I/O and much less disk space. We are currently performing more detailed experiments and comparisons, also on the performance of different kinds of queries.

Conceptual Modelling of Temporal Databases and their Implementation in Database Systems

Thomas Myrach
University of Bern, Institute of Information Systems
Engehaldenstrasse 8, CH-3012 Bern, Switzerland
myrach@ie.iwi.unibe.ch

So far the discussion about temporal data models has been primarily 'technical'. Design issues for temporal data schemas have been scarcely addressed. A usual approach for designing data schemas is the modelling of the relevant mini-world with a semantic data model and the subsequent transformation of the results to, e.g., the relational model. We pursue the adoption of this approach to the design of temporal data schemas. One of the main points in modelling temporal data on a conceptual level is to express whether data is time-varying or not. In this respect, one may distinguish two fundamental approaches: On one hand, it is assumed that facts are generally time-invariant and time-varying facts are expressed by adding timestamps. On the other hand, one may assume that every fact in principle is temporal and, therefore, may change over time. Thus, every fact type is implicitly associated with a timestamp. We evaluate both approaches within the framework of Object Role Modeling (ORM) which does not anticipate a grouping of facts as in methods based on the Entity Relationship Model. Besides the work on conceptual modelling we are also interested in the implementation of temporal data schemas. This may be done in one of the proposed temporal data models, e.g., the consensus language TSQL2. Even though the research on temporal databases has reached a level of relative maturity, the availability of commercial products is not to be expected in the short and middle run. Thus, pragmatic solutions are needed which support temporal functionality on top of existing (relational) technology. We have investigated ways to use triggers and stored procedures for this purpose which are provided by modern database systems. This work proved that it is on principle possible to implement temporal functionalities by these means even though there are some limitations and possible difficulties with respect to performance. Both research areas may be combined in developing routines for the automatic generation of appropriate temporal modules from the conceptual schema. This is planned as a future activity.

Handling of Events and Alternatives in Temporal Databases

N. L. Sarda
Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai, 400 076 India
nls@cse.iitb.ernet.in

Planning for the future is an important activity both at individual and organizational levels. Planning consists of defining alternative actions to handle various events in the future. The alternatives arise because of uncertainties in the outcomes of events (e.g., a research proposal may or may not be approved). A plan consists of a sequence of actions to be carried out for each possible outcome of an event. In the context of database modeling, the actions are operations on a database (e.g., hire technical staff if research proposal is approved). A database management system should not only enable its users to define events and alternatives, but allow them to interrogate the database under different alternatives (possibly to evaluate different plans). The existing temporal data models treat future analogous to past or present; they provide for one future path (in the sense that facts valid at some future time can be stored), and they do not provide support for alternatives in future.

We have developed a model for incorporating events and alternatives by extending the temporal data model to support 'branching' time. The extended model permits definition of events, their inter-dependencies and associated actions. The events that affect an object are modeled by a tree, permitting an object to have different states at same valid time but under different alternatives. The event tree shows inter-dependencies among events. The 'branching' time paradigm is obtained by superimposing linear valid time on the event tree. We extend the temporal relational algebra and the Temporal SQL2 for supporting branching time data model. The paper also briefly deals with the uncertainties associated with future planning as well as probabilities of possible event outcomes. Finally, we sketch an implementation strategy for the branching time data model.

Bulk Loading Temporal Index Structures

Bernhard Seeger
Universität Marburg
FG Informatik
Hans-Meerwein-Str., 35032 Marburg, Germany
seeger@informatik.uni-marburg.de

(Joint work with Jochen van den Bercken (Univ. Marburg) and Peter Widmayer (ETH Zurich).)

Recently there has been an increasing interest in supporting bulk operations on advanced index structures. Bulk loading refers to the process of creating an initial index structure for a presumably very large data set (which is not very uncommon for temporal data bases). For bulk loading B+-trees there is a well-known algorithm which is implemented in almost all commercial database systems. This algorithm first sorts the data set and then, builds up the index from bottom to top. In contrast to B-trees, temporal index structures like the multiversion B-tree (MVBT) support two dimensions, the time dimension and the key dimension. Due to the multidimensionality of temporal data the standard approach of B+-trees is not applicable anymore to temporal index structures.

We presented a generic algorithm for bulk loading which is applicable to a broad class of multi-dimensional index structures, in particular to temporal index structures like the MVBT. Our approach avoids sorting temporal data according to a one-dimensional order. Instead, our approach is based on the standard routines for splitting and merging pages. These routines are available without changes from the target index structure. Second, in contrast to inserting records one by one, multiple records are inserted simultaneously during bulk loading. As an example we demonstrated in this talk how to apply our technique to the MVBT. It can be shown for the MVBT that our generic algorithm is asymptotically I/O optimal. Empirical results demonstrate that performance improvements are also achieved in practice without sacrificing search efficiency of the index. To the best of our knowledge, the problem of bulk loading temporal index structures has not been addressed so far in the literature.

Transitioning Temporal Support in TSQL2 to SQL3

Richard Snodgrass
Department of Computer Science
University of Arizona
Tucson AZ 85721, USA
rts@cs.arizona.edu

I have recently been involved in implementation of temporal databases (especially the stratum approach and implementing coalescing, temporal joins, and aggregation), infrastructure (in particular notations for temporal queries and for temporal granularity), and furthering temporal databases in the SQL standard.

I presented the valid time and transaction time change proposals that have be submitted to the ISO SQL3 committee. These constructs are variants of those first defined in TSQL2. I started with a brief chronology of work by the temporal database community. TSQL2 was finished in 1994. I started working with the ANSI and ISO SQL3 committees late that year. The first step was to propose a new part to SQL3, termed SQL/Temporal. This was formally approved in July, 1995. Jim Melton agreed to edit this new part. The valid-time and transaction-time proposals were unanimously accepted for forwarding to ISO, where they have yet to be voted on.

I then outlined the three kinds of queries (views, cursors, modifications, constraints, assertions) supported by these extensions: temporal upward compatible, sequenced, and nonsequenced. Temporal upward compatibility ensures that non-temporal legacy applications will continue to work when history is retained in one or more of the underlying tables. Sequenced queries are extensions of conventional queries that request the history of the desired information. Nonsequenced queries simply treat the timestamp as another column. I showed how to use these facilities allow one to migrate smoothly from a conventional relational system to one encompassing temporal support, and allow one to easily express queries of the form "List the history of...".

Temporal Relational Databases

Abdullah Uz Tansel
Baruch College - CUNY
Box E-0435, 17 Lexington Avenue
New York NY 10010, USA
uztbb@cunyvm.cuny.edu

Tansel's early work in temporal databases extends the relational data model by attaching timestamps to attribute values by using set valued attributes. An attribute value is modeled as a temporal atom, $\langle$[l, u),v$\rangle$. A temporal atom asserts that the value v is valid for the time period represented by the interval [l, u). Hence, temporal data is represented as a cube where time comprises the third dimension. This is a powerful approach in modeling temporal data and supports non-homogeneous tuples, regrouping of temporal data and temporal reduction. The temporal relational algebra the operations are defined at the attribute level to directly manipulate the three dimensional structure.

Tansel later generalized his work to arbitrarily nested relations and defined temporal algebra and calculus languages. Temporal relational algebra includes slice, temporal atom decomposition and formation operations in addition to the traditional algebraic operations. Temporal relational calculus includes set membership test and a set constructor formula. He also shows that these two languages have the same expressive power when temporal relational algebra is augmented with a looping construct. Tansel used the temporal relational calculus as a yardstick in evaluating the expressive power temporal query languages and showed that all the other proposed temporal relational query languages and their models can be expressed by his temporal relational algebra.

Tansel designed two temporal query languages, HQUEL and Time-By-Example. He also explored temporal aggregates. Tansel was the head of the editorial board that published the first book in temporal databases.

Corporate Memory Systems

Babis Theodoulidis
TimeLab, Department of Computation
UMIST
P.O. Box 88 Manchester M60 1QD, United Kingdom
babis@co.umist.ac.uk

In the information society today, database management systems play a very prominent role. They store, update and retrieve information, knowledge, facts and figures about our Universe of Discourse. It is not surprising that they are the second largest market for software after operating system software. Database systems are expected to resemble more and more living organisms, i.e. to interact with each other, to form federations with other systems which share similar interests, to continually evolve and more importantly, to behave intelligently. Part of our research at UMIST is focusing on the development of techniques to build "intelligence" or "consciousness" in a database system. The one technique we have investigated more extensively, is to build "memory" mechanisms in the database system in order to provide it with the ability to recall the experience and history of events much the same as humans do to form consciousness (Scientists offer computer with mind of its own, Stuart Millar, The Guardian, 17th of December 1996). This work is often referred to as temporal databases and its main objective is to extent current database technological products and applications with add-on components for modelling and reasoning about time and time-varying information. The end result is that one is able to reason about the past e.g., to know what is the sales history of the Timperley branch and to be able to interpret time correctly by understanding that "31-8-93" is the same as 31st of August 1993, thus avoiding the mishaps caused by the 2000 problem (Doomsday 2000, Matthew Holbrook, Computer Shopper, March 1997). We have experience in applying our techniques to applications involving medical records such as patient history and liver transplant management, to applications involving human resources such as allocation of employees to projects and task planning, to applications involving student records such as admissions management. However, the application domains were this technology can be applied are virtually everything because "nothing remains permanent except change" and everything needs to know about the past in order to be able to reason about the future.

Temporal Extensions of SQL Revisited

David Toman
Department of Computer Science
University of Toronto
10 King's College of Road, Toronto Ontario M5S 1A4, Canada
david@cs.toronto.edu

Current proposals of temporal extensions to SQL commonly use interval-based representation of time as the basis for their data models (e.g., TQUEL or TSQL*). This choice causes severe problems when defining a single-dimensional semantics for such languages: it often turns out to be very cumbersome. We propose an alternative and more natural point-based data model as the basis for a temporal extension of SQL. The proposed language extends the syntax and semantics of SQL/92 in a very natural way: by adding a single data type to represent time. Such an extension has many advantages:

- It vastly simplifies the semantics of the proposed temporal query language,
- The users can write temporal queries in a natural declarative fashion,
- It supports meaningful duplicate semantics and aggregation, and
- Queries in the proposed language can still be efficiently evaluated over a compact encoding temporal relations.

The query evaluation is based on a compilation technique that translates the extended language to SQL/92. In this way existing database systems can be used for managing temporal data. We substantiate this claim by proposing an experimental version of a compiler of the extended language to SQL/92 to serve as a front-end for DB2.

A side-effect of this work is to make the ISO/ANSI standardization committees aware of potential pitfalls in the current proposals (e.g., limited expressive power, typing violations, representation dependencies, or the need for DEXPSPACE operators) and of the possibility of alternative solutions that avoid these pitfalls.

Temporal Hashing and a Notation for Spatiotemporal Queries

Vassilis J. Tsotras
Dept. of Computer Science
Polytechnic University
Brooklyn, NY 11201
email: tsotras@aegean.poly.edu

Hashing has been traditionally used in database systems as a fast method to answer membership queries (for example: given a dynamic set of objects find whether a given object k is in the set). Among the various proposed hashing methods, Linear Hashing has been the prevailing approach in databases. Here we extend this work and address the problem of Temporal Hashing. In our setting the set of objects evolves over time and the membership queries have a temporal predicate: "find whether a given object k was in the set at time t". We propose a new approach, namely Persistent Hashing, to address such queries and compare it with various straightforward methods (like traditional linear hashing). Our comparisons show that Persistent Hashing outperforms all other approaches. In particular, Persistent Hashing behaves as if a separate linear hashing function was used on each state assumed by the evolving set over time. However, the space used by our method remains linear to the total number of changes in the evolution.

We also present a new notation for Spatiotemporal queries. Temporal, spatial and spatiotemporal queries are inherently multidimensional, combining predicates on time dimension(s) with predicates on explicit attributes and/or several spatial dimensions. In the past there was no consistent way to refer to temporal or spatiotemporal queries. In an attempt to eliminate this problem, we propose a new notation for such queries. Our notation is simple and extensible and can be easily applied to multidimensional queries in general. For our purposes a spatiotemporal object is characterized by six entries: a key attribute, a valid-time interval, a transaction-time interval and three spatial attributes. We propose the following basic notation to classify spatiotemporal queries: Key//X_dimension/Y_dimension/Z_dimension//Valid/Transaction. A "//" (double slash) distinguishes the explicit attribute(s) from the temporal qualifiers and from the spatial qualifiers. Each qualifier can take a number of different values based on the query presented (S for single value, R for range of values, E for a set of value ranges, '*' for any value and '-' for no applicable value).

Pattern Discovery in Temporal Data Models and Query-Driven Simulations

Alexander Tuzhilin
New York University
Stern School of Business, Dept. of Information Systems - Rm. 9-78
44 West 4th Street, New York NY 10012, USA
atuzhili@stern.nyu.edu

1. Pattern Discovery in Temporal Databases (Data Mining): Temporal logic is used to specify temporal patterns and to express temporal patterns, that are processed by methods I have developed for discovering the most frequent and the most unexpected patterns in temporal data. Unexpectedness of a pattern is when the actual and expected numbers of occurrences in a temporal database differ significantly from each other. I have been using temporal logic programming for the discovery of frequent patterns expressed in temporal logic and developing efficient search methods for the discovery of unexpected patterns.

2. "Grouped" vs. "Ungrouped" Temporal Data Models: a joined work with Jim Clifford and Al Croker on attribute vs. tuple timestamping in temporal relational data models. We showed that the former type is more expressive than the latter and argued that grouped models have several other important advantages over the ungrouped models.

3. Applications of Temporal Databases to Query-Driven Simulations: Simulation programs are executed and appropriate statistics about simulation outcomes are collected in order to answer end-user questions about simulation models. P. Balasubramanian and I proposed an alternative approach to simulations, called the Query-Driven Simulation (QDS) approach, where the user asks a query about outcomes of simulations expressed in a temporal query language followed by appropriate simulations and events necessary to answer the query. After the simulation runs are completed, trace files are converted into a temporal database format, and the temporal query is evaluated on the temporal database containing these trace files. The QDS approach provides a nice application of temporal databases to simulations because temporal databases lie at the heart of this approach.

On Mining Temporal Dependencies

Jef Wijsen
Department of Computer Science
Free University Brussels
STARlab - Building F-G/10 - Room 10G730
Pleinlaan 2, B-1050 Brussels, Belgium
jwijsen@vub.ac.be

In earlier work, we studied several types of temporal dependencies. These dependencies were primarily used for temporal databases integrity. In recent work, we looked at temporal dependencies from a data mining perspective.

As a contribution to this Dagstuhl seminar, we presented a paper concerning the computational complexity of mining certain trends in temporal databases. A simple example of such trend might be "In general, salaries of employees do not decrease." The trends considered are formalized by the construct of trend dependency (TD). TD's can compare attributes over time by using operators of <,=,>,<=,<>,>=. TD satisfaction is characterized by a support and confidence. As TD's may express meaningful trends, mining them is significant. The TD mining problem studied is the following task: Given a temporal database, find the TD of a specified form that holds with the highest confidence and with support greater than or equal to a specified minimum threshold.. This problem is called TDMINE. Unlike most other work in data mining, we primarily focus on the computational complexity of the TDMINE problem – rather than on the performance of algorithms to solve it. Both the number of tuples (cardinality) and the number of attributes can be taken as the "size" of TDMINE. TDMINE can be solved in quadratic time if the cardinality is taken as the size. If the time requirements are expressed in function of the number of attributes – rather than the cardinality – then the problem turns out NP-complete. We discuss the practical implications of this result.

Dynamic Attributes and Temporal Triggers in Active Databases

Ouri Wolfson
University of Illinois at Chicago
Dept. of Electrical Engineering and Computer Science
851 South Morgan Street, Chicago, IL 60607, USA
wolfson@eecs.uic.edu

1. Dynamic attributes: Existing database management systems (DBMS's) are not well equipped to handle continuously changing data, such as the position of moving objects. The reason for this is that in databases, data is assumed to be constant unless it is explicitly modified. For example, if the salary field is 30K, then this salary is assumed to hold (i.e. 30K is returned in response to queries) until explicitly updated. Thus, in order to represent moving objects (e.g. vehicles) in a database, and answer queries about their position (e.g., How far is the vehicle with license plate RWW860 from the nearest hospital?) the vehicle's position has to be continuously updated. This is unsatisfactory since either the position is updated very frequently (which would impose a serious performance and wireless-bandwidth overhead), or, the answer to queries is outdated. To address this problem, we proposed a data model called Moving Objects Spatio-Temporal (or MOST for short) for databases with dynamic attributes, i.e. attributes that change continuously as a function of time, without being explicitly updated.

2. Temporal Triggers in Active Databases: We are building a rule-processing component that can be incorporated into commercial-off-the-shelf Database Management Systems. The component will add monitor and control capabilities to existing DBMS's. The target applications consist of a continuously changing database that represents the status of a real system (e.g. a battlefield). Using the rule-system the user will be able to specify conditions that need to be monitored in real-time over the changing database, and actions to be taken upon occurrence of these conditions. The novel aspect of our rule-processing component is the rule language which, in contrast to existing languages, allows the concise specification of temporal conditions (i.e., conditions on the way the database evolves over time) and temporal actions.

Dagstuhl Seminar on Temporal Dsatabases: Summary of the Closing Session

Chair: S. Sripada
S. Sripada: Four discussion topics were suggested for the closing session:

1. Introduction of ideas to the industry.
2. Future Meetings.
3. Future Research Directions.
4. Further Collaboration.

INTRODUCTION OF IDEAS TO THE INDUSTRY
H. Darwen: You may not think that enough progress has been achieved in industrial implementations. I was one of the recipients of material about temporal databases, and I found that I had to re-think on the concepts presented using my terms, which are shared among many people in this area. Since the temporal extension to SQL was frozen until 1999, this community has two years to reflect upon what to present and how to present to the industry. I would like to get clearer picture, we will be interested to see the community consensus, but please don't try to poison us.
C. Jensen: SQL standardization is a political process run by major database vendors that have big commercial interests in SQL. It is very difficult for independent scientists to be part of the national and international standards bodies and take part in standardization. And the scientific rewards that can be obtained are few. I advocate that we, as researchers, focus on doing what we do best, namely research. For temporal databases to achieve impact, I consequently advocate that we target the potential users of our technology: By making functionality available to the user communities and thus creating a customer demand, we can hope that temporal database startups will emerge and that the major vendors will provide temporal support in their products.
B. Thoedoulidis: There are two things that should be done: devising the "best practice" of industrial applications and composing a "white paper". For the standards issue, we should put there a minimal thing. We can sell parts and not the entire idea.
S. Sripada: There have been two approaches to standardization. One is supplying several data types, the other is to supply not only the foundations,

but the design as well. These alternatives should be carefully considered. The benefit

of minimality is that it is safer, but we may prefer the bigger picture.
A. Gal: We are over-shadowed by the attempt of linkage with the industry. We create an atmosphere that is unclear for the entire community. If we will do a good research, the industry will reach us.
T. Ozsu: There are two unique features to this community:

1. There is no success in getting to the industry. This is a cloud that lies above this community.

2. There is a desire to speak in an homogenous voice.

I don't understand why we should speak in a homogenous voice, and why it is a big deal, there is beauty in diversity. I worked in the industry, and I am currently 14 years in the academia, and realized that there is not a unique model of how things should be done. We should do what we know how to do. Not everybody can work on language design issues, but if it is important to people, we can see three alternative proposals. It is not important to me. What is missing in this community is that we spend too much time on modelling issues, and not on proof of concepts implementations. Build systems! It took IBM a plenty of time to build a relational database. I don't want to speak in homogenous voice, I just want to do the best research I can. If temporal options will be picked, they will be picked. Each vendor has a list of required options. For part of them, temporal issues is higher in the list, and for part of them it is lower.
S. Gadia: I agree to what Tamer Ozsu said. I suffered from dealing with temporal databases.
H. Darwen: I also agree with Tamer Ozsu. If you don't want to go to the industry, the industry will come to look for you. About the minimality issue, there is a pyramid, the willingness to minimality exists in the language level.
R. Snodgrass: This meeting proved that we have a wonderful community, that there are people from different backgrounds who are able to communicate. There is great diversity, with a wide range of topics investigated. We should celebrate the strong scientific legacy our community has created.
FUTURE MEETINGS:
T. Ozsu: I am against annual meeting. This creates conflicts with the regular conferences. These paper should be integrated with the main conferences.

B. Theodoulidis: The benefit of a workshop is the interaction. In the big conferences there is no interaction. I prefer an annual meeting.
J. Chomicki: The relationships with other communities is weak. We should improve these relationships and the interaction with other communities, by participating in other conferences.
S. Sripada: There is a place to two types of activities. Workshops with the need to compete for papers are problematic. The question is what events should be organized in the future.
B. Theodoulidis: One of the alternatives is to do a workshop next year linked with either VLDB or CAISE. We can invite people from the spatial community to increase the interaction. There is also a possibility for interaction with the AI community in TIME.
A. Montanary: TIME is a workshop that takes place every year in Florida. There are some papers from the database community, for example the issue of events or temporal constraints.
A. Gal: I participated in TIME, and I wonder if the AI people will be interested in database issues.
C. Bertini: Some AI topics are relevant.
A. Montanary: We should check with the organizers. In any event, it cannot be the main event for this community.
V. Tsotras: In my opinion we should do a series of conferences. The spatial database community has an annual conference that is considered a good conference. Anyone with nice results should go to SIGMOD or VLDB, it is good if some papers are presented there. More specific papers should be in a specialized annual conference. Our standards should be high.
S. Sripada: We should devise a mailing list for the temporal database community, and make proposals for further events in this mailing list.
R. Snodgrass: I am willing to coordinate this mailing list.
FUTURE RESEARCH DIRECTIONS:
C. Jensen: There is a European project on spatio-temporal databases. I believe that application-driven research, where a new and demanding application sets novel challenges, is a promising approach to developing new and useful technology.
H. Darwen: Query Optimization.
J. Wijsen: Some topics have been neglected. For example: optimization that is not algebraic, but cost-driven. There is a big gap between the language and the data structures.

R. Elmasri: At the Temporal Workshop in Zurich, the topic of time series was mentioned. This is a topic that we should work on. Also, the extension of object-oriented databases to temporal databases.
A. Gal: Issues of data mining. The combination of temporal and active databases should be further investigated, we just scratched the surface.
C. Jensen: Modularization of temporal support where temporal functionality is separated into modules that can be adopted independently, gradually, and according to the particular needs.
J. Chomicki: This area needs theoretical foundations.
D. Toman: There was a transformation in programming languages to languages that are more mathematically-based.
T. Ozsu: In the active database community there was a recommendation not to invent new language, but to deal with the semantics of existing ones, this may also apply to temporal databases.
FURTHER COLLABORATION:
S. Wang: I propose a working-group in the area of granularities and calendars.
M. Soo: I am coordinating a working-group on benchmarking and performance issue.
S. Sripada: The community should keep in touch.
Summarized by O. Etzion