Report on
The 1995 International Workshop on Temporal Databases

Arie Segev

Haas School of Business
University of California, Berkeley &
Lawrence Berkeley National Lab
Berkeley, CA 94720-1900, USA

Christian S. Jensen

Department of Math and Computer Science
Aalborg University
Fredrik Bajers Vej 7E
DK-9220 Aalborg Ø, DENMARK

Richard T. Snodgrass

Department of Computer Science
University of Arizona
Tucson, AZ 85721
USA

Abstract

This paper provides an overview of the 1995 International Workshop on Temporal Databases. It summarizes the technical papers and related discussions, and three panels: "Wither TSQL3?", "Temporal Data Management in Financial Applications," and "Temporal Data Management Infrastructure & Beyond."

1 Introduction

The International Workshop on Temporal Databases was held in Zürich, Switzerland, September 17–18, 1995, immediately following the 21st Conference on Very Large Databases, also held in Zürich. This arrangement was chosen to facilitate the widest possible international participation by enabling many participants to attend both events.

The Workshop was, as the first workshop ever, held in cooperation with the VLDB Endowment, Inc. It was jointly sponsored by ARPA, the National Science Foundation, Aalborg University, and ETH Zürich. The Workshop

The workshop had three main objectives. The first objective was to provide a follow up forum to discuss and evaluate infrastructure-related progress since the ARPA/NSF-sponsored International Workshop on Infrastructure for Temporal Databases that was held in Arlington, Texas, in June 1993 (the report of that workshop was published in SIGMOD RECORD, Vol. 23, No. 1, March 1994). The second objective was to discuss future directions related to the advancement of temporal database infrastructure in commercial database technologies and applications. The third objective was to discuss research papers containing material not necessarily related to infrastructure issues in the shorter term.

The above objectives were achieved through the presentations and discussions of an excellent set of technical papers and panels. Section 2 provides more details about the submissions profile and summarizes the papers and related discussions. The three panels are summarized in Sections 3 through 5.

The temporal database infrastructure discussions spanned multiple sessions. They were part of technical papers that dealt with the evaluation of TSQL2; a panel on TSQL3; and a panel on temporal database infrastructure that summarized that state of TSQL2, the glossary, and temporal database implementations. They made it evident that the temporal database field has made significant and accelerated progress as reflected by the 1993 first book on temporal databases, the 1993 infrastructure workshop, and the subsequent glossary and TSQL2 efforts.

The concluding general discussion (which is detailed in Section 5) focused on the state of the temporal database community, problems that it might face, and its interaction with other groups. Issues related to research directions were also raised. Overall, the partipants felt that it was a good and worthwhile workshop, and, following a vote, the majority expressed their interest in another workshop in two years.

2 Summary of Paper Presentations

2.1 Overview

The Call for Papers aimed at attracting submissions from the global temporal database community of up-to-date research results of the highest quality.

A total of 55 paper submissions were received from 17 countries on five continents, clearly illustrating the global nature of temporal database research. Using double-blind reviewing, 18 of these papers were accepted for publication in the proceedings and for presentation at the workshop. The proceedings, entitled "Recent Advances in Temporal Databases" and edited by James Clifford and Alexander Tuzhilin, were published by Springer-Verlag (1995, 360+x pages, ISBN 3–540–19945–4) in the Workshops in Computing series, which is a collaborative effort with the British Computer Society and with C. J. van Rijsbergen as series editor.

The research topics and approaches represented in the proceedings were quite diverse, ranging from quite theoretical work to work with a focus on the use of temporal database technology for solving practical, real-world problems. Some work addressed data modeling issues and some was concerned with query language issues. Among this work, three papers were devoted to different aspects of constraints. Other work dealt with implementation-related aspects such as access methods, transaction, and schema versioning. And yet other work was concerned with temporal data application development and was motivated by applications that pose challenges to temporal database technology.

In addition to this quite satisfying diversity, the technical program also exhibited some common themes. In the survey of the paper presentations that follows, space restrictions preclude a thorough coverage of each individual paper—for that, the reader is instead referred to the proceedings. Rather, we explore some of the themes and variations among the papers.

2.2 Timestamps Attributes—How Special Should They Be?

To understand this first theme, observe that a temporal database relation may, roughly speaking, be thought of as a conventional snapshot relation with one or more timestamp columns added. Values of these columns then may encode the valid or transaction times of the tuples they stamp.

As one approach, timestamp attributes may in principle be no different from other attributes. In SQL–92, periods may be encoded using start and stop attributes of type `TIMESTAMP` ("datetime"); or one can add a `PERIOD` column type and associated operations to the data model and achieve better support. In an object-oriented context, this corresponds to supporting time by extending the available class library with temporal types. This approach is desirable in that it represents a minimal and relatively straightforward extension of current data models and query languages.

Another approach is to accord the timestamp attributes some degree of special semantics in the data model and query language. It has been argued that providing built-in support for time in this fashion yields queries that are more easily formulated, understood, and proven correct. This is the typical approach chosen by temporal data models, including TSQL2.

In TSQL2, a single "implicit" timestamp attribute encodes both valid time and transaction time. The transaction-time aspects of queries and updates are almost completely built into the data model and are essentially handled by the system. There is also built-in defaults, e.g., snapshot reducibility, and special syntax for valid time, e.g., for timestamp referencing, update, schema definition, etc.

As a final approach, it may be argued that simply adding a timestamp column, no matter how implicit, is not an adequate foundation for providing support for the management of temporal data. Simply adding columns yields a temporally ungrouped model. In contrast, it is possible to achieve a temporally grouped, or history oriented, model by carefully building time into the attribute values of tuples. In this type of model, histories are first-class objects. To achieve this desirable property, temporally grouped models, although based on the relational model, represent a substantial departure from the relational model.

This theme was discussed among the workshop participants and was touched upon by a number of papers, from both practical and more theoretically-oriented perspectives.

2.3 Critical Evaluations of TSQL2

The development of the consensus temporal query language TSQL2 was initiated at the ARPA/NSF International Workshop on an Infrastructure for Temporal Databases which was held in Arlington, Texas, in June 1993. The language design was subsequently completed in September 1994, and a book documenting the language became available just before the Workshop ("The TSQL2 Temporal Query Language", Kluwer Academic Publishers, 1995, 674+xxiv pages, ISBN 0-7923-9614-6) and was given to each participant as part of the registration package.

One of the objectives of TSQL2 was to serve as an infrastructure for temporal database research. Several papers provided different insights into and perspectives on TSQL2, or used the language as the context for new research contributions.

One paper reports on a comprehensive study where natural-language queries are translated into TSQL2 queries. This study revealed that some language aspects, e.g., to do with multiple granularities, remain to be properly documented or worked out. This lead to proposals for clarification and extensions to TSQL2. A different paper makes the point that any query language that is an extension to SQL–92 is inappropriate for some users and instead proposes a graphical query language as an alternative language for these users. Yet another paper investigates TSQL2 with respect to several completeness notions. It was shown that while TSQL2 satisfies some completeness properties, there are other notions that it does not satisfy. Finally, a paper provides a critique of the idea of building temporal semantics into a data model and query language, exemplified with TSQL2, and proposes to instead leave the relational model as is and instead handle the temporal data management aspects solely in the application.

2.4 Challenges from Non-typical Applications

Another theme was the challenges to temporal database technology posed by applications that reach beyond the typical administrative data management applications. Several papers concerned temporal aspects of such applications. Some of these, where much research is still warranted, are briefly mentioned in what follows.

In geographic information systems, time is an essential concept when understanding and modeling spatial phenomena in diverse applications such as bio-physical sciences; epidemiological research; social, economic, and political sciences; and various real-time applications for, e.g., management and planning.

Applications such as financial trading and analysis systems, medical treatment plans, and ecological monitoring may also benefit from improved temporal support in the DBMS. These applications require support for incomplete data and quite complex temporal queries. Typical queries involve relative relationships between future events and past events with unspecified absolute occurrence times. An example is "Which stocks and bonds produce a series of coupons with a given, possibly irregular, cycle?"

While largely ignored so far by temporal database researchers, time series management applications also are prime candidates to benefit from built-in temporal support in the DBMS. However, the requirements from time series applications are quite different from the traditional ones and thus are largely unmet. They include the integrated support for calendars and time series, also including multivariate time series and groups of time series. It is particularly important to note that a "time series" does not denote a uniform type of object, but spans very diverse types of objects. An essential and apparently simple task such as preprocessing of time series, e.g., for homogenization purposes, is one of the difficult and largely unmet challenges. Among other requirements, advanced facilities for aggregate and statistics queries are needed.

Next, audio and video data posses inherent temporal properties. Video segments, streams of frames, and audio segments from video recordings are combined into derived, or virtual, video documents. As a complicating factor, different segments may have different associated time coordinate systems and frame rates. As the human interpretation of a video segment is very context dependent, the management of contexts, with both spatial and temporal properties, is crucial. While clearly quite different from traditional applications, video data management applications are candidates for benefiting from temporal DBMS support.

Finally, work has focused on the support for more "traditional" temporal data management applications using existing commercial DBMSs. Applications include pharmaceutical product control, sewer system management, social benefits management, inventory control, decision support, and system-performance monitoring. This interesting work provides concrete examples of how and to what extent existing DBMSs support temporal data management.

3 Panel: Whither TSQL3?

TSQL2 was designed to be a minimal temporal extension of SQL-92. As such, the language does not use or exploit the constructs and facilities being added to SQL-92 to produce SQL3, which is expected to become an international standard in 1997. This panel considered a new language, *TSQL3*, which was initially envisioned two years ago to be a temporal object-oriented query language (TOOQL), as an extension of SQL3.

The panel began with the moderator, Richard Snodgrass, outlining the various languages in question. Figure 1 illustrates these languages, along with interactions between them. SQL-92 is the starting point; it is already the international standard (languages that are standardized or are expected to be standardized are indicated with darker borders). TSQL2 was completed in October, 1994, adding *temporal tables* to SQL-92. In parallel, SQL3 is being defined, also as an upward-compatible extension of SQL-92. SQL3 adds *abstract data types (ADTs)* as column values in relations, *inheritance* among these ADTs, and *object identifiers*, also as column values in relations.

SQL/Temporal was added to the SQL3 definition as a new part in July, 1995, with the `PERIOD` predefined data type being the first aspect of TSQL2 to become part of SQL3. The intention is to move more of TSQL2's concepts and functionality into SQL/Temporal. The next aspect to be proposed is that of temporal tables. This extension is compatible with the other aspects of SQL3: such tables could have ADTs or object identifiers as column values within rows. One issue before the panel was to what extent is SQL3 + SQL/Temporal already an object-oriented temporal query language?

Once the SQL3 language (including the SQL/Temporal part) becomes a standard, the standardization effort will focus on the next iteration, SQL4. While SQL4 will be a strict superset of SQL3, the basis for TSQL3 was another issue before the panel. Should the basis be SQL3 without SQL/Temporal (i.e., start from a clean slate), should the basis be SQL3 with SQL/Temporal, or should the basis be yet another language, such as ODMG's Object Query Language?

Finally, when and if TSQL3 is defined, what should be the relationship between it and SQL4? Should SQL4 be extended along the lines of TSQL3, yielding yet another extended language, SQL4'?

With the context set, the panelists each stated their position on some aspect.

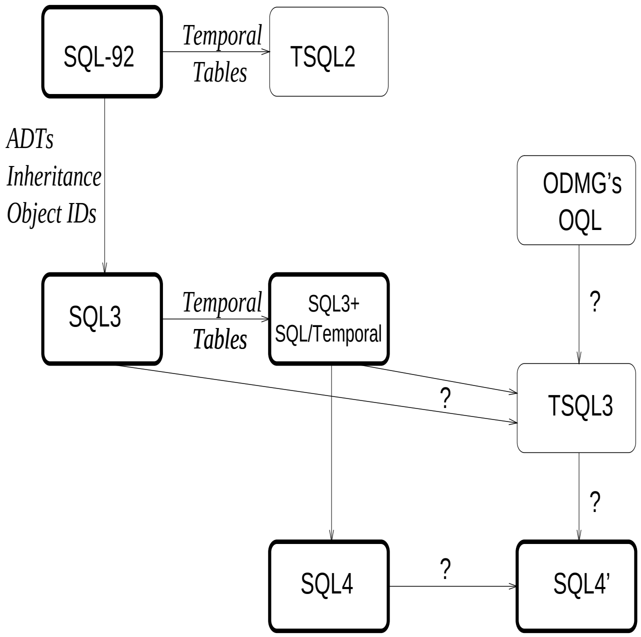

Figure 1: Relationship between Existing and Future Query Languages

Umeshwar Dayal, from Hewlett-Packard Labs in Palo Alto, was the co-designer of OODAPLEX, an object-oriented extension of DAPLEX which can be used effectively for temporal applications. His position is that the guiding principles for TSQL3 should be the following.

- *parsimony* Use the power of the type system to avoid introducing special constructs for temporal data.
- *extensibility* Use the extensibility inherent in the "object-oriented" type system to support a wide range of application semantics.
- *completeness* Cover *all* aspects of SQL where time might be important.

The benefits of this approach are several. A smaller, more elegant language design would result, which would be easier to use, uniform for temporal and non-temporal data, and consistent with existing query optimization frameworks. The expressiveness of other aspects of SQL should be relied upon for useful temporal operators such as temporal modalities and temporal aggregates. This approach would support a wider range of application semantics than currently feasible with TSQL2, including multiple time dimensions, time series, topological versus metric relationships, and interpolation functions. OODAPLEX serves as an existence proof of this approach. As a first step, the community should try to influence the standards bodies to include other, ostensibly non-temporal features that would be useful for TSQL3, including partial orders, tables and functions as collection types, and temporal triggers, constraints, and inheritance.

Ramez Elmasri, from the University of Texas at Arlington, has done significant work in adding time to Extended Entity-Relationship Models, which provides insight into the facilities that should be in a TOOQL. He began by stating that the concepts that will make up SQL3 are slowly being finalized and approved. At this point, it seems likely that provisions for defining abstract data types, with inheritance, will be part of the SQL3 model and language. This will provide a method of allowing various researchers and language developers with diverse ideas to implement their temporal data models/query languages as abstract data types.

A lot of effort was spent in the design of TSQL2, and an implemented prototype exists. However, there is still very limited experience in using it in practice. Hence, his position is that TSQL2 should also be implemented using the abstract data typing mechanism of SQL3. Others researchers with alternative temporal models/query languages could also implement abstract data types for their proposals if they want their models to be available for temporal database users/researchers. If several options exist and are available to the users, then it is possible that the marketplace will decide if any particular approach is easier to use and more complete in general. It is also possible that no one approach can cover the wide range of temporal database applications, and that different applications will find it best to utilize different approaches.

Fabio Grandi, from the University of Bologna, has worked for several years to define language constructs that support what he terms *history orientation*. His position is that temporal groupedness (history-orientation) is absolutely vital to any advanced temporal query language. Temporal data models and query languages have been shown to be formally more expressive that ungrouped ones. It has also been shown how they can be more clear and friendly for human users as they support the concept of history as a first-class object of discourse. Furthermore, they represent the most natural extension of the snapshot relational approach.

Temporal groupedness is in some way inconsistent with SQL-92 and its temporal extension TSQL2. Fabio feels that it cannot be added to TSQL2 as an optional additional feature, as it basically represents the "right" and only way of adding time, in his opinion. For example, users should be allowed to use system-defined data types or to define new data types with the option to make them temporal or not; when such option is checked, the temporal representation is (1) automatically managed by the system via built-in functionality and (2) based on the grouped approach. Any other kind of time additions will be considered as user-defined time dimensions, unrelated to the temporal (valid- and transaction-time) representation.

Therefore, support for that should be carefully considered in developing new SQL standards. His strong recommendation is that the effort to develop a temporal extension to SQL3 be based on the temporally grouped approach, so that it will better meet the user requirements and the modeling needs of temporal information.

Finally, Arie Segev was the co-designer of the temporal object-oriented model TOODM and language TOSQL. His position is that SQL3 is not fully object-oriented. Temporal support is needed at all levels, e.g., rules, schema, etc. Supporting a time sequence type directly will be useful, and so will be the support of calendars as ADTs. From the temporal database community point of view there are three main options.

1. Extend TSQL2, making it more object-oriented.

2. Design TSQL3 as a new extension to (an evolving) SQL3.

3. Design a Temporal Object-Oriented Language (TOOL) and try to influence OQL, SQL3, and TSQL2 with extensions to support it, as well as develop mappings to those languages (and possibly other).

His opinion is that option 3 (which includes aspects of the first two) is preferable.

An energetic discussion followed these position statements. Concerning the degree of additional temporal functionality, the panelists were in agreement that a TOOQL should be full-featured, and go significantly beyond SQL3, in both temporal and object-oriented aspects. All but Arie felt that SQL3 should be the basis of TSQL3; Arie felt that a separate TOOL should be designed from scratch. However, all but Fabio felt that it was premature to begin language development of TSQL3 now. Instead, users should use the ADT, object ID, and temporal table facilities in SQL3+SQL/Temporal in disparate applications. Only when there is sufficient user experience should additional language facilities be proposed. Fabio disagreed, stating instead that the notion of temporal groupness was sufficiently compelling to be required of TSQL3. The panel ended on an admonishment from the audience to not have the TSQL3 language design sap the creative energies of the temporal database community. While TSQL2 seems to have been accepted as important infrastructure, the community should now refocus on research and application before starting yet another consensual language design effort. There was little enthusiasm for commencing a TSQL3 effort in the near term.

4 Panel: Temporal Data Management in Financial Applications

The panel, moderated by Arie Segev, consisted of three panelists: Duri Schmidt from Union Bank of Switzerland (UBS), and Roman Barnert and Guido Schmutz from RBA Service. The primary objective of the panel was to use the context of financial applications as an example of applications that may benefit substantially from temporal database support, and to examine how the requirements are being satisfied in that context. The ensuing discussion, however, also emphasized the point that there is an important class of applications (time series data management) that has requirements which are not met (or are likely to be met) by the main-stream temporal database infrastructure efforts.

Arie Segev opened the panel discussion with the following four general statements.

- Not all financial applications are the same (different requirements)

- Some requirements can be satisfied by current commercial database management systems

- Those that cannot be satisfied are not getting enough attention from the temporal database community

- Some functional features, if supported, will benefit non-financial applications as well

Duri Schmidt made a distinction between temporal data management and time series management (TSM). That is, the general temporal database research and development does not focus on the specialized requirements of TSM. At UBS, many projects develop their own solutions in the areas of "time dimensions in databases," "storage of data with temporal structure," and "effects of time in logical data models." In general, the bank deals with issues related to the ontology of time, how to store temporal data in RDBMS's, archiving, and new ideas concerning temporal queries. In the area of temporal data management, the recommendation regarding useful research was to work on conceptual database design, logical and physical database design, and query facilities, in a way which works satisfactorily with nowadays RDBMS's and covers a reasonable subset of TSQL2.

In the area of TSM, time series in the bank are used in portfolio management, risk management, and trading. They are managed with RDBMS's and commercial TSM systems. The statement was made that there are many problems in TSM that are awaiting solutions, but they are not provided by the temporal database research community since most efforts concern general temporal (mostly relational) data management.

In conclusion, users, such as UBS, cannot wait for TSQL2 implementations. They need solutions for current RDBMS's. Consequently, the issue of migrating from these solutions to TSQL2 has to be addressed.

Roman Barnert and Guido Schmutz from RBA (which provides information processing services for Swiss regional banks) described applications whose temporal features can be implemented on current commercial DBMS (Oracle versions 7.1–7.3, in this case). They stressed the importance of temporal data in marketing and controlling applications, and they described their 3-phase approach to providing such functionality. In the first phase, the temporal concepts were explained to the database designers, followed by the design of a suitable bitemporal model in the second phase. The third phase involved the implementation of the temporal model on Oracle; since the mapping between those two phases is not one-to-one, an adaptation of the temporal model was needed at the Oracle database level. A description of a GUI-based client-server architecture was also given.

Two important summary points were made. The first point was that since modeling is left to the database designers, education on the subject of temporal data models is fundamental and missing. The second point related to their general needs in the area of temporal databases: accepted standards, referential integrity (rule activation), and interaction with the Temporal DB community.

5 Panel: Temporal Data Management Infrastructure & Beyond

The objectives of this panel were to report on the status of temporal database infrastructure efforts and to have a general discussion among the workshop participants. The panel's moderator, Arie Segev, gave a brief history of the temporal database community's efforts that led to this workshop (see the introduction to this report). He then presented an overview diagram of how temporal database research can impact practice. Three main avenues were mentioned: standards activities, vendors' extensions, and "temporally smart" shells. It was stressed that the research should be motivated by needs of applications. The opening overview was followed by a survey of temporal prototypes and a report on the temporal database glossary by Michael Böhlen and Christian Jensen, respectively, from Aalborg University, and a report on TSQL2 by Richard Snodgrass.

Michael Böhlen summarized the state of temporal database implementations. Rather than being very specific about each system, he provided an indication of the functionality together with pointers to additional information. In order to include as many prototypes as possible, a broad definition for temporal database systems was adopted. As a consequence, this summary not only includes descriptions of systems that qualify as temporal database systems in the first place but, also descriptions of systems that are related to temporal database systems, e.g., a temporal database generator.

Besides general descriptions, the survey classifies each system according to traditional selection criteria. An analysis of these criteria reveals interesting properties, and "non-properties," of actual implementations. As an example of the latter, most systems have not been evaluated against large databases. Also, traditional database features like persistence, transactions, and concurrency are not always provided. This raises the question of what a system has to provide in order to be rated as a temporal database system. Another interesting point is that the set of query operations that has been investigated is highly unbalanced. This probably means that many query languages only provide limited functionality. Further details are given in the paper which follows this report. Finally, the paper contains information only on those implementations that were reported prior to the workshop. Readers are encouraged to contact Michael Böhlen with information about non-reported implementations.

Richard Snodgrass summarized the situation in 1993, when the temporal infrastructure workshop was held. The previous fifteen years of research had yielded many results, including more than two dozen temporal relational query languages and one dozen temporal object-oriented query languages. However, the lack of a common data model or query language was hampering both research and commercial development of temporal databases. The TSQL2 effort was initiated immediately after the workshop, in July, 1993. An initial draft was released to the research community in the March, 1994 *ACM SIGMOD Record*, and the final language design was released in September, 1994. In July, 1995 the ISO SQL3 committee voted unanimously to accept a new part: SQL/Temporal; the base document for this part initially contains the `PERIOD` data type from TSQL2. In August, the TSQL2 book was published, and at the workshop a prototype implementation developed by Michael Böhlen and Andreas Steiner was made available. Future plans involve introducing portions of TSQL2 to SQL3/Temporal as change proposals to the base document. Initially valid-time tables will be proposed, encompassing extensions for schema specification and adding temporal upward compatibility and snapshot reducibility to the semantics. Support for non-sequenced queries and updates will be next proposed. Later proposals will concern event tables, storing "now" in the database, transaction time, temporal granularity, aggregates, temporal indeterminacy, schema versioning, and vacuuming.

Christian S. Jensen described the glossary initiative. He initially provided an overview of its background, current state, and next steps. Then he characterized challenges involved in working with the glossary.

Fundamentally, the glossary associates terms with concepts and defines consistent temporal-database-specific terminology. It is controlled by, developed by, and is for the temporal database community. Specifically, anybody in the temporal database community may participate in maintaining the glossary, which is controlled by the participants. In this sense, it is a consensus document.

He emphasized that the glossary is not an introduction to temporal databases or an historical account of the development of temporal databases. It is to be used as a reference, with the index being the primary entry point. Consequently, a uniform, dictionary-like format, composed of small, self-contained entries has been adopted.

Next, it was stated that the glossary has been received well and has had a positive impact. It helped make the temporal database book mentioned earlier more coherent, it has created awareness about temporal database concepts and terminology, and its terms are being used widely. It was pointed out, however, that a list of 24 unresolved, proposed entries exist, that additional previously suggested concepts probably merit inclusion, and that there is a need for a revision of certain current glossary entries.

As a next step, work on a new release of the glossary is scheduled to start later this winter. The kick-off will be announced on dbworld, and the contributions of the community are essential. Finally, these challenges were emphasized as important.

- Deciding what concepts to include in the glossary
- Making small, self-contained, and precise definitions for the glossary
- Finding good terms for the concepts included
- Ensuring incrementality of the glossary
- Encouraging the community to use the glossary

The general discussion that followed focused on issues related to the meaning of "consensus" in the glossary and TSQL2, the degree of effort that should be devoted to standards-related activities, and the possibility (or reality) of limiting the scope of the research areas and the exclusion or lack of involvement of other groups in the database field. The issue of theoretical versus applied research was also touched upon. Some people felt that too much of the community's energy has been devoted to standards-related activities and that exclusion of other topics is occurring. It was also stated that this group should open up more and integrate other parts of the database community into its efforts (an example was the area of active databases). How real versus perceived these issues are is not clear, but like many other aspects of life, perceptions can also be a cause for concern. Consequently, the authors of the report believe it is important that the community be sensitive to these statements. Following a constructive discussion of those issues, it seemed that a consensus opinion was expressed that TSQL2 and its extension into TSQL3 (if that occurs) should not limit other research, and that the word "consensus" as applied to TSQL2 and the glossary should be interpreted in its context, that is, it does not mean that these are the best language or glossary, or that other efforts (when motivated by real-world needs) should not take place.

Overall, the partipants felt that it was a good and worthwhile workshop, and, following a vote, the majority expressed their interest in another workshop in two years (opinions were evenly divided between full papers and short papers).

Acknowledgements

We thank Mario Nascimento for supplying notes taken during the workshop. Together with our own notes and the proceedings, these notes were helpful in writing this report. We also thank Ramez Elmasri and Fabio Grandi for providing their position statements, included here, and Michael Böhlen, Umeshwar Dayal, and Duri Schmidt for providing copies of their slides. Finally, we gratefully acknowledge James Clifford's significant involvement in the workshop as program committee chair. Jim was responsible for all aspects of the technical program, which he organized before his untimely passing in August. It was unfortunate that Jim was not able to attend this workshop; we all missed him terribly.