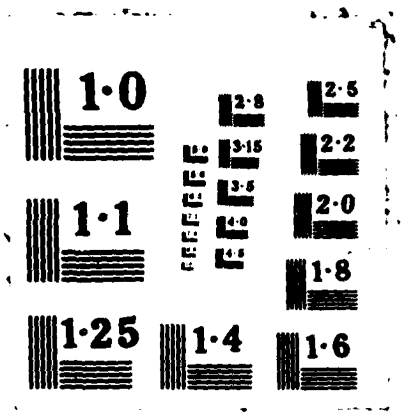
AD-A188 963 THE TEMPIS PROJECT: CURRENT STATUS(U) NORTH CAROLINA UNIV AT CHAPEL HILL DEPT OF COMPUTER SCIENCE R SNODGRASS 19 JUN 87 UNC-TEMPIS-15 N00014-86-K-0680 1/1

UNCLASSIFIED F/G 12/5 NL


1·0
2·8
2·5
3·15
2·2
3·5
2·0
1·1
4·0
4·5
1·8
1·25
1·4
1·6

DTIC FILE COPY

AD-A188 963

The TEMPIS Project

The TEMPIS Project: Current Status

Richard Snodgrass

June 19, 1987

Abstract

We give the status of the TEMPIS project, identifying our initial goals, evaluating our progress in light of these goals, and listing future work.

TEMPIS Document No. 15

Copyright © 1987 University of North Carolina

Department of Computer Science
University of North Carolina
Chapel Hill, NC 27514

DTIC
ELECTE
DEC 1 1 1987
S D
E

This document has been approved for public release and sale; its distribution is unlimited.

87 11 10 080

RE: Copyright Notice: The Defense Technical Information Center and the National Technical Information Service have permission to reproduce and sell this report.

DEFENSE LOGISTICS AGENCY
DEFENSE TECHNICAL INFORMATION CENTER
CAMERON STATION
ALEXANDRIA, VIRGINIA 22304-6145

11/24/87

REPLY REFER TO DTIC-DDA

25 NOV 1987

SUBJECT: Distribution Statement on Technical Reports

TO: OFFICE OF NAVAL RESEARCH
ATTN: CODE 1133, DR. CHARLES HOLLAND
ARLINGTON, VA 22217

1. DTIC has received the report referenced below without an approved distribution statement.

2. Every unclassified report must have an approved distribution statement. Classified reports may have a distribution statement, but do not assign Statement A to classified reports. Distribution statements are required by DoD Directive 5230.24, Distribution Statements on Technical Documents, 20 Nov 84. Approved distribution statements are listed on the reverse of this letter. If you have any questions regarding these statements, call DTIC's Selection Section, (202)274-6837, Autovon 284-6837.

3. Please indicate the proper distribution statement and return this form to the above address, ATTN: DTIC-DDAC. We request your reply within 20 working days. Delay in responding will result in denying availability of this report to qualified users.

FOR THE ADMINISTRATOR:

Richard A. Matthews

RICHARD A. MATTHEWS
Chief, Acquisition and
Selection Branch

SOURCE: (Prepared by) UNIVERSITY OF NORTH CAROLINA, CHAPEL HILL, NC

TITLE: THE TEMPIS PROJECT: CURRENT STATUS

A

CONTRACT/GRANT NO: N00014-86-K-0680

REPORT NO: TEMPIS DOCUMENT NO. # 15

DATE OF REPORT: 87/06/19

DTIC FL-182, Feb 85

file

In this brief report, we look to the past, the present, and the future of the TEMPIS project.

1 Initial Goals

The goal of this project ~~was and~~ is a formalization and implementation of a temporal database management system (TDBMS). In the past three years, we have concentrated on the TQuel language, its definition, formalization, and initial implementation. We originally subdivided the main goal into six fairly substantial tasks, that we thought we could accomplish by this time:

1. Develop a formal semantics of TQuel in the tuple calculus.

2. Define an historical relational algebra.

3. Formalize this algebra and prove that the algebraic expression associated with any TQuel statement preserves the semantics defined by the tuple calculus.

4. Implement an historical DBMS based on TQuel.

5. Develop algebraic optimizations and prove that they preserve the semantics.

6. Instrument the implementation to determine the actual performance with and without the optimizations.

We partitioned TQuel into three distinct aspects:

1. the core language, which supports valid time,

2. the portion of the language supporting aggregates, and

3. the portion of the language concerned with temporal indeterminacy.

As each aspect is relevant to each of the tasks enumerated above, we thus identified 18 subtasks that we hoped to address.

Soon after starting the project, we determined that there are many types of time to be stored in a database, and we precisely characterized what we view to be the three primary types [Snodgrass & Ahn 1986]. The first kind of time, termed *valid time*, involves recording the history of the enterprise being modelled in the database. Our original proposal dealt only with valid time. The second kind of time, termed *transaction time*, involves recording the history of updates to the database. The third, *user-defined time*, is not interpreted by the database management system, and is easy to support. Supporting transaction time was added to our study, increasing the number of subtasks to 24.

2 Current Status

In Table 1 we give the current status of the subtasks. We have completed about half of these subtasks, have made substantial progress on another four, and are actively working on all but five subtasks. Support for temporal indeterminacy has thus far been given short shrift. We view this aspect as somewhat less interesting (and less difficult) than the other aspects, which is the reason it remains uninvestigated; nevertheless, we plan to begin working on it next year. Finally, we have made some progress in the display of temporal relations [Shannon 1986] and in the use of TQuel for monitoring complex systems [Snodgrass 1985, Snodgrass 1986A].

To put our activity in context, there are approximately two dozen research projects investigating issues of databases and time [Snodgrass 1986B]. About a dozen query languages involving time have been defined [Snodgrass 1987]. There are four properties that we deem essential for a query language to adequately support time.

- It must be *well defined*, in that it has a formal retrieval semantics.
- It must *support historical queries*, and hence valid time, in that queries can be formulated that derive information (i.e., tuples) valid at a point in time from information in underlying relations valid at other points in time, much as conventional query languages can derive information concerning entities or relationships for information in underlying relations concerning other entities or relationships.
- It must *support rollback*, and hence transaction time, in that queries can be formulated that retrieve information stored in a previous transaction (and possibly removed in a subsequent transaction).
- It must be *implementable*, as demonstrated formally through a semantics based on a well defined algebra, or through an actual implementation.

Only TQuel currently satisfies all four of these properties.

	Core Language	Supports Aggregates	Supports Temporal Indeterminacy	Supports Transaction Time
Semantics Developed	√[1]	√[2]	P[6]	√[1]
Algebra Defined	√[3]	√[3]	□	√[4]
Algebra Formalised	√[3]	√[3]	□	√[4]
Implementation	√[5]	□[9]	□	√[5]
Optimisations Devel.	P[7,8,10,11]	□[9]	□	□[8]
Impl. Instrumented	P[5,8]	□	□	P[5,8]

"√" denotes completion
"P" denotes substantial progress
"□" denotes as yet minimal progress
[1] [Snodgrass 1987]
[2] [Snodgrass & Gomez 1986]
[3] [McKenzie & Snodgrass 1987A]
[4] [McKenzie & Snodgrass 1987B]
[5] [Ahn & Snodgrass 1986]
[6] [Snodgrass 1982]
[7] [Snodgrass 1986A]
[8] [McKenzie 1988]
[9] [Valiente 1987]
[10] [Ahn 1986A]
[11] [Ahn 1986B]

Table 1: Status of Subtasks

Accession For
NTIS GRA&I ☒
DTIC TAB □
Unannounced □
Justification per ltr
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

On another scale, TQuel is thus far the only query language possessing the five properties that we feel provide a minimal definition of *temporal completeness*:

1. It must be *snapshot complete*, in that the language, when applied to a snapshot of the database, is at least as powerful as existing conventional query languages that are complete according to Codd's original definition.

2. It must be well defined, which is necessary to prove snapshot completeness.

3. It must have a *well defined update semantics*, which is also necessary to prove snapshot completeness.

4. It must support historical queries.

5. It must support rollback.

3 Future Work

We feel that more work is needed in the areas of physical storage structures and algebraic optimizations for temporal databases. Graphical display of temporal relations and support for temporal indeterminacy also require further investigation. Fortunately, temporal databases are on a firm formal basis, which will aid future research.

In the short term (the next six months), we plan on completing the prototype (termed the TempIS prototype), by adding support for aggregates, temporal indeterminacy, and perhaps an evolving schema. We also want to more fully integrate Equel and the graphical user interface with the prototype, thereby generating a fairly complete extension of the Ingres system to handle temporal data. This prototype will demonstrate functionality as well as providing limited feedback on efficiency. At this point, we will have met most of the 24 goals set three years ago.

In the medium term (twelve–eighteen months), we will step back and start the design of a second-generation temporal database management system called the TEMPIS system. TEMPIS will involve several innovations, including

- a new, incremental temporal algebra,

- support for temporal indeterminacy,
- support for optical storage devices,
- support for conventional and temporal aggregates,
- logical and physical distribution of the prototype.

Our initial design efforts will focus on the graphical user interface, incremental query evaluation, and concurrency control and crash recovery of relations supporting transaction time.

Our long-term goals are less precisely defined. While the overall objective remains the implementation of a TDBMS, we realize that many aspects of such an implementation are mundane yet resource-intensive. Our approach will be to build on the work of others whenever possible (the approaching availability of extensible DBMS prototypes is especially exciting), while identifying and solving the fundamental problems confronting such an objective.

4 Acknowledgements

This research was supported in part by NSF grant DCR-8402339, ONR contract N-00014-86-K-0680, an IBM Faculty Development Award, and a UNC junior faculty award.

5 Bibliography

[Ahn 1986A] Ahn, I. *Towards an Implementation of Database Management Systems with Temporal Support*, in *Proceeding of the International Conference on Data Engineering,* IEEE Computer Society. Los Angeles, CA: IEEE Computer Society Press, Feb. 1986, pp. 374-381.

[Ahn 1986B] Ahn, I. *Performance Modeling and Access Methods for Temporal Database Management Systems.* PhD. Diss. Computer Science Department, University of North Carolina at Chapel Hill, July 1986.

[Ahn & Snodgrass 1986] Ahn, I. and R. Snodgrass. *Performance Evaluation of a Temporal Database Management System*, in *Proceedings of ACM SIGMOD International Conference on Management of Data,* Ed. C. Zaniolo. Association for Computing Machinery. Washington, DC: May 1986, pp. 96-107.

[McKenzie 1986] McKenzie, E. *Bibliography: Temporal Databases. ACM SIGMOD Record,* 15, No. 4, Dec. 1986, pp. 40-52.

[McKenzie & Snodgrass 1987A] McKenzie, E. and R. Snodgrass. *Supporting Valid Time: An Historical Algebra and Evaluation.* Technical Report TR87-008. Computer Science Department, University of North Carolina at Chapel Hill. June 1987.

[McKenzie & Snodgrass 1987B] McKenzie, E. and R. Snodgrass. *Scheme Evolution and the Relational Algebra.* Technical Report TR87-003. Computer Science Department, University of North Carolina at Chapel Hill. May 1987.

[McKenzie 1988] McKenzie, E. *An Incremental Temporal Relational Algebraic Language (in progress).* PhD. Diss. Computer Science Department, University of North Carolina at Chapel Hill, 1988.

[Shannon 1986] Shannon, K.P. *The Display of Temporal Information.* Master's Thesis, Computer Science Department, University of North Carolina at Chapel Hill, July 1986.

[Snodgrass 1982] Snodgrass, R. *Monitoring Distributed Systems: A Relational Approach.* PhD. Diss. Computer Science Department, Carnegie-Mellon University, Dec. 1982.

[Snodgrass 1985] Snodgrass, R. *A Relational Approach to Monitoring Complex Systems.* Technical Report TR85-035. Computer Science Department, University of North Carolina at Chapel Hill. Dec. 1985.

[Snodgrass 1986A] Snodgrass, R. *Filtering Monitoring Data.* Technical Report TR86-001. Computer Science Department, University of North Carolina at Chapel Hill. Jan. 1986.

[Snodgrass & Ahn 1986] Snodgrass, R. and I. Ahn. *Temporal Databases. IEEE Computer,* 19, No. 9, Sep. 1986, pp. 35-42.

[Snodgrass & Gomez 1986] Snodgrass, R. and S. Gomez. *Aggregates in the Temporal Query Language TQuel.* Technical Report TR86-009. Computer Science Department, University of North Carolina at Chapel Hill. Mar. 1986.

[Snodgrass 1987] Snodgrass, R. *The Temporal Query Language TQuel. ACM Transactions on Database Systems (to appear),* 12, No. 2, June 1987.

[Snodgrass 1986B] Snodgrass, R. (ed.) *Research Concerning Time in Databases: Project Summaries. SIGMod Record,* 15, No. 4, Dec. 1986, pp. 19-

39.

[Valiente 1987] Valiente, J. *Implementing TQuel Aggregates (in progress).* Master's Thesis, Computer Science Department, University of North Carolina at Chapel Hill, 1987.

END

FILMED

MARCH, 1988

DTIC