The 8th International Conference on Computer Supported Cooperative Work in Design Procceedings

Bitemporal Extensions to Non-temporal RDBMS in Distributed Environment

Yong Tang, Lu Liang, Rushou Huang, Yang Yu
Computer Science Department of ZhongShan University, GuangZhou 510275,P.R.China
pianoll@tom.com , issty@zsu.edu.cn

Abstract

Temporal information processing is the key and advanced technology of new generation of database and information system. Based on the research on Temporal Information Model (TIM), we think the Embedding Temporal Application mode is an important and proper method to implement temporal application at present. So we use temporal extended middleware to accomplish temporal processing. After deep researching on ATSQL2 query language and the translation algorithm from ATSQL2 to standard SQL used in TimeDB, we have developed a bitemporal extended software platform upon non-temporal RDBMS in distributed environment, which lays the foundations of adding more temporal knowledge expression, query and reasoning into TIM.

1. Introduction

Time is an essential attribute of information. Neither the traditional relational database (RDB) nor the object-oriented database (OODB) specially manipulates temporal data. Both of them are lack of the capability to record and handle temporal information. As the latest development of database and information technology, the requirement of temporal information processing is more and more exigent in many application systems. Temporal information processing has become a key technology of the new generation of database and information system.

Many researchers have done a lot of work on temporal database, and generated many temporal data models and query language according to them. In the middle 80's of last century, about one hundred approaches have been brought forward, which can add the function of handling temporal information into database management system (DBMS). After competitive research in several years, these models have been concluded into thirteen models [1]. Each of them has built a set of term, conception and mathematic model to form theoretics. Their focus is mainly on extending the data structures and/or the query language. But hardly any of these temporal data models were implemented by any commercial DBMS, even in the form of prototype systems [2,3]. Up to the present, the standardization of temporal data model has not succeeded. There is no mature temporal model and software product so that the traditional technology is still used in most of the temporal applications in practice, which means that the temporal function is accomplished in application program instead of database. So, in order to put the temporal database technology into real-world application more quickly, it is regarded as a more feasible approach that builds temporal extensions to DBMS.

In the process of studying and researching temporal database, we have brought forward a Temporal Information Model (TIM) in paper [4]. TIM includes three parts: TDM（Temporal Data Model）、TKM（Temporal Knowledge Model）、TKDM（Temporal Knowledge/Data Model）. We use TimeDB (an extended database software supporting bitemporal treatment) for reference, and have recently built a TDM prototype, which is a middleware platform for temporal processing and is based on temporal extensions to traditional RDBMS and Remote method Invocation (RMI) technology.

2. Temporal Information Processing

2.1. The evolution of Temporal Application

Because non-temporal data model and database management system only handle an instant-time state of a object during its whole life time, and not consider the object's history and future. So researchers bring forward the technology of temporal database, in order to solve the problem about how to handle temporal information in the real world. Now, application systems, which can provide the function of temporal information processing, are in a more and more mature development. According to the manner and capability of handling temporal attribute in the system, we think that there are three evolution stages of temporal application, as depicted in Fig 1: Mixed Temporal Application, Embedding Temporal Application, and Pure Temporal Application.

0-7803-7941-1/03/$17.00@2003 IEEE

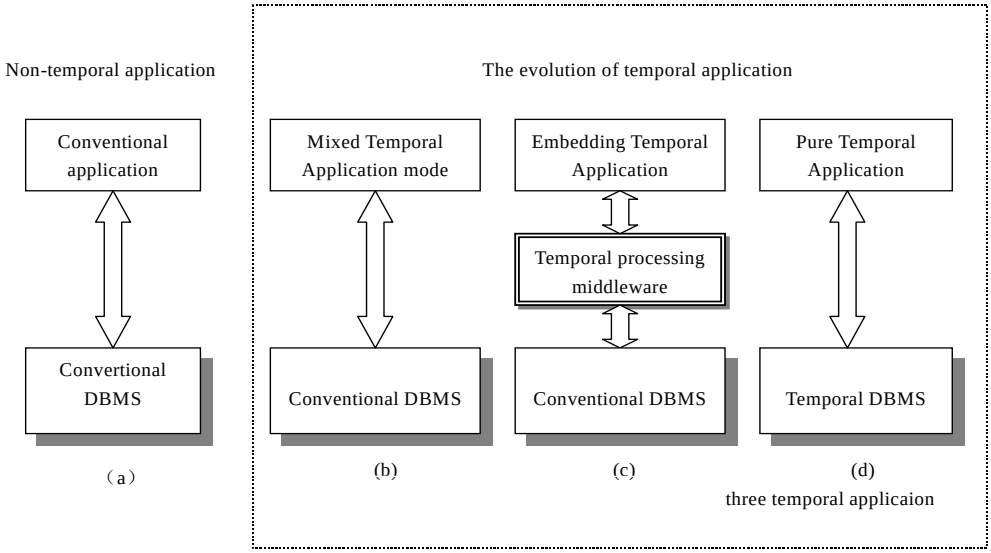


Fig 1. The architecture of three temporal application

2.2. Embedding Temporal Application

Embedding Temporal Application mode is a method to implement temporal application. In this mode, some temporal development tools (named as temporal processing middleware) are used to accomplish the function of temporal information processing in application systems, and the bottom database is still managed by the traditional DBMS, as depicted in Fig1(c). Nowadays this kind of system is the direction of developing temporal application. Temporal processing software is required to support this mode of application, such as TimeDB of Time Consult company, a distributed bitemporal extended RDBMS software platform developed by us.

Embedding Temporal Application Software can be logically decomposed into three layers[4], as depicted in Fig1 (c):

1) Temporal application layer: the interface to contact with the users. It provides temporal pretreatment tools for users to definite the temporal data views, to edit the temporal knowledge base, to use the service which temporal middle layer provide to accomplish the temporal information manipulation.

2) Temporal middle layer: provide temporal information processing service on web, including temporal query, temporal reasoning etc.

3) Bottom database layer: accomplish the expression of temporal information on traditional database.

2.3. Bitemporal Extensions to Non-temporal RDBMS

Because relational data model is based on complete theoretics, many commercial DBMS manufacturers have given great support to the model, and have implemented corresponding relational DBMS (RDBMS). RDB and RDBMS have become the mainstream of database application domain. Therefore, the research of temporal processing middleware based on RDBMS in Embedding Temporal Application mode, is a key point in the field of temporal database.

According to the design process of database application program and the required function of temporal data processing, we consider that a successful temporal RDBMS should depend on a set of complete and mature extended SQL language, and do relevant extensions on three aspects as follow:

1) Extends the data structure, including defining new data type such as interval, in order that the different states of database can be identified.

2) Provides a set of relevant table-level temporal operations and column-level temporal operations to support temporal calculating.

3) Do temporal extensions to constraint, in order to satisfy more abundant semantic requirement and keep the consistency of temporal database.

3. ATSQL2 and TimeDB

3.1. ATSQL2 Language

ATSQL2[5,6] is an extension to SQL to support the management of time-varying data. It was designed by an international group of researchers. ATSQL2 includes not only a bitemporal query language, but also a bitemporal

modification, data definition and constraint specification language. It is the result of integrating three different approaches: TSQL2 (a temporal query language based on SQL), ChronoLog (introducing the concept of temporal completeness), and Bitemporal ChronoSQL (featuring that any query may be evaluated with respect to valid time, transaction time or both).

3.2. TimeDB

TimeDB[7] is an implementation of the bitemporal relational database system by Andreas Steiner etc [8]. TimeDB is based on the generalisation approach, however only to a limited extent. Temporal relations, for example, are extensions of the non-temporal relations with timestamp attributes. TimeDB supports the temporal query language ATSQL2. TimeDB runs as a frontend to the commercial RDBMS Oracle. ATSQL2 statements (queries, updates, and assertions) are compiled into (sequences of) SQL-92 statements which are executed by the backend. TimeDB provides bitemporal statements, and handles valid time and/or transaction time. It excels in a seamless integration of time into databases by supporting upwards compatibility[9] and temporal upwards compatibility[10].

4. A Distributed Bitemporal RDBMS Platform

Based on the Embedding Temporal Application mode, we have brought forward Temporal Information Model (TIM)[4]. Then we use temporal extension algorithm of TimeDB for reference, and develop a distributed bitemporal RDBMS platform based on the Remote Method Invocation (RMI) technology as a elementary implement of TDM prototype. The primary idea of this software is that add a temporal compiled extension layer on traditional RDBMS to build temporal RDBMS.

4.1. System Architecture and Data Flow

The distributed bitemporal RDBMS platform is programmed in Java language, and use Remote Method Invocation (RMI) technology to implement the Web communication. The software platform comprises two parts: server and client, and construct a distributed data model of C/S mode. The server handles the logic which is relevant to database access, such as, receive ATSQL2 statement, translate statement and calculate result, and return the result to client. The client uses the interface provided by server to send ATSQL2 query request, and gives corresponding treatment on the return result. The work process of client and server is depicted in Fig2.

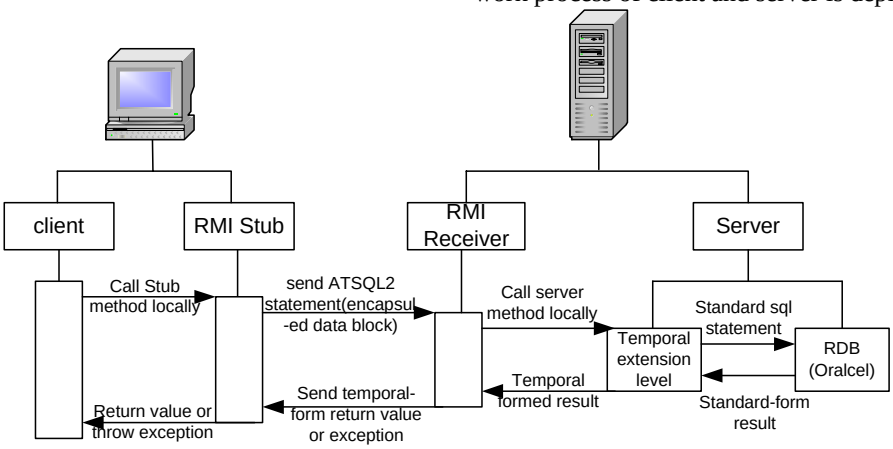


Fig 2. The system architecture of distributed bitemporal RDBMS platform

4.2. Key Functional Components of Server

The server is a bitemporal extended RDBMS, which use the Embedding Temporal Application Mode. The backend RDBMS is ORACLE8.1.6-.0.0. The frontend is a temporal process middleware, which actually is a compiled module to implement the translation from ATSQL2 statement to standard SQL statement. The backend and frontend is connected by JDBC. Fig 3 depicts the main modules of compile function. The translation algorithm of TimeDB is used in the compile modules, which implements a set of temporal extended algebra operations using standard SQL statements [8]. The extended algebra operations include unitemporal algebra operations and bitemporal algebra operations to orthogonally treat valid time and transaction time.

As shown in Fig 3, when a ATSQL2 statement is passed into the frontend temporal processing middleware, firstly the module scanner analyses the words to generate a token stream, which is passed into the parser next. The module parser then builds a syntax tree on the token stream, and at the same time use translation rules, which

are some operations defined in the module Algebra, SQL and Coalescing, to transform the special syntax form of temporal data into the standard form of SQL (implement the translation from ATSQL2 to standard SQL). In the translating process, the object query is required to store the query statement in the memory, and checks whether it is syntax-valid. When a syntax exception is checked, an error message is thrown out and the translation is stopped. In addition, if there are subqueries relevant to temporal characteristics, which are the views and derived tables of the clause from, or the subqueries of the clause where, some assistant tables are generated to store these subqueries result, which then will do relational-calculation on the main query relation. Concerning the relational calculation, this temporal middleware does not calculate the correlative objects once they are generated. In stead, it uses two static object defined in module stmt to point to the stmt objects (structured in a chain) generated in translating process. One object stores the SQL statements, which are used to built the assistant tables and do temporal relational calculations. The other object stores some delete statements, which can atomically delete such assistant tables at a proper time. When the translation finishes, the operations defined in module accessDB are used to access the backend RDBMS by executing the standard SQL statements in stmt object chain. Thus, the final generated query result according to the original ATSQL2 statement is returned from database. At last, the delete statement stored in the second stmt object are executed to release the assistant tables.

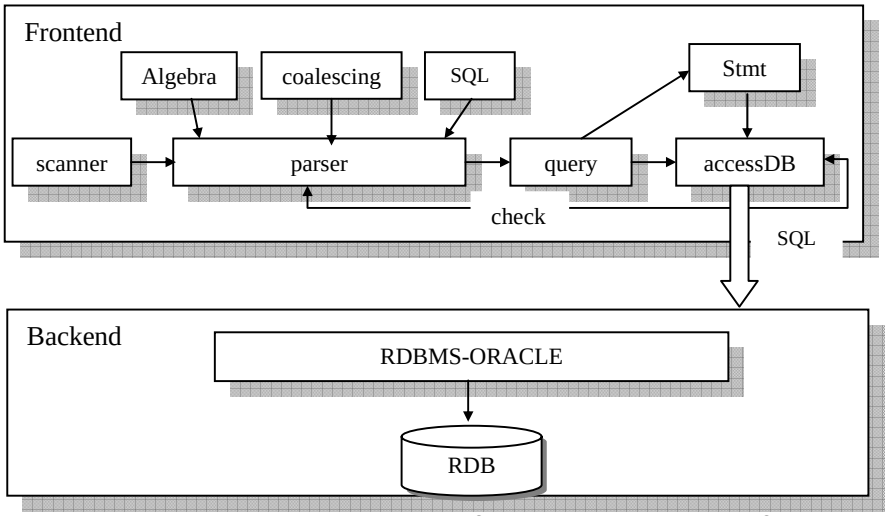
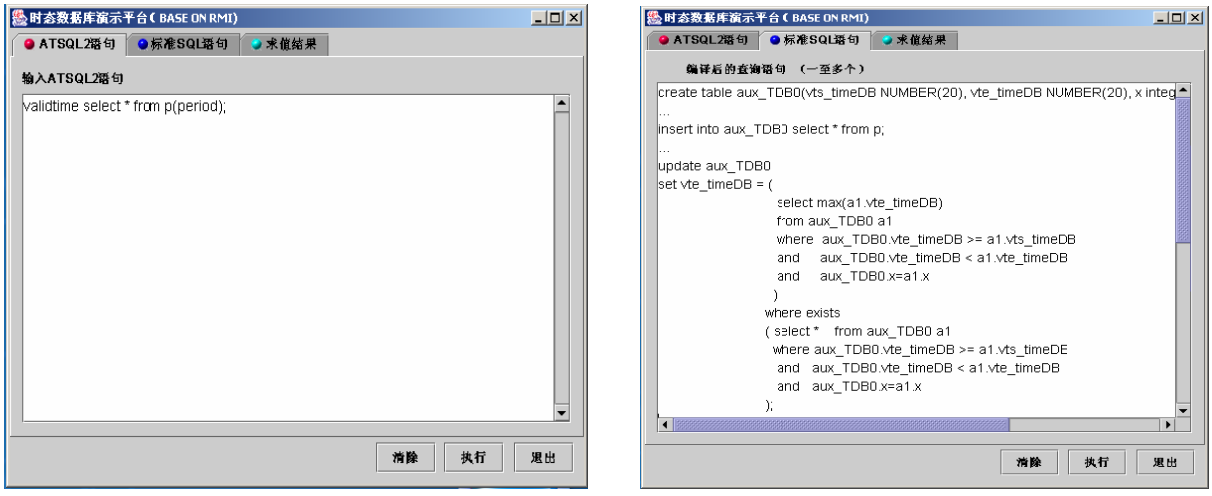


Fig 3: The key compile components of bitemporal RDBMS platform



(a) Input ATSQL2 statements **(b) Output standard SQL statements**

Fig 4. A demo interface of bitemporal RDBMS platform

4.3. Client Interface

The client interface comprises three parts: input interface of ATSQL2 statement (Fig4(a)), display interface of generated standard SQL statement (Fig4 (b)), and output interface of final query result. The SQL statement interface is an intermediate result, which is just used to display the translation result to the user.

5. Summary

It is described in this paper that an important approach to implement temporal application. Based on Temporal Information Model (TIM)[4], we have developed a database middleware for handling temporal data, which is a distributed temporal extended software platform on RDBMS. This research is just early preparative work in order to implement a pure temporal database in the future. Next, we will try to improve the algebra operation algorithm of temporal relational databases, change the implementations of algebra operations with more efficient ones, and add special indexing techniques for faster query evaluation, or develop extensions of indexing techniques for temporal data. On the other hand, we will extend TSQL language, and add temporal knowledge expression, query and reasoning into TIM in real-world application.

6. Acknowledgement

This work is supported by the National Natural Science Foundation of China Grant NO.60373081, the Guangdong Provincial Science and Technology Foundation of China Grant NO.A10203, and Guangzhou Science and Technology Foundation of China Grant NO. Z2-D3041.

13. References

[1] Tansel A. ,Clifford J. ,Gadia S. ,et al. Temporal Database—Theory, Design and Implementation. The Benjamin/ Cummings Publishing Company, 1993.

[2] Tang CJ. The achievement, deficiency and future work in temporal database. Computer Science, 1999,26(3): 63-65 (in Chinese with English abstract).

[3] He XG, Tang CJ, Li L, Liu YS. Special Type Database Technology. Beijing: Science Press, 2000(in Chinese).

[4] Tang Yong, Tang Na, Ye Xiaoping, Feng Zhisheng, Xiao Wei. A Unified Model of Temporal Knowledge and Temporal Data. Journal of Software, Vol.14 Supplement 2003: 74-78 (in Chinese with English abstract).

[5] R. T. Snodgrass, M. H. Bohlen, C. S.Jensen, and A. Steiner. Adding Transaction Time to SQL/Temporal,SQL/ Temporal Change Proposal , ANSI X3H2-96-502r2, ISO/IEC JTC1/SC21/WG3 DBL MAD-147r2, November 1996.

[6] R. T. Snodgrass, M. H. B ohlen, C. S. Jensen, and A. Steiner. Adding Valid Time to SQL/Temporal, SQL/ Temporal Change Proposal, ANSI X3H2-96-501r2, ISO/IEC JTC1/SC21/WG3 DBL MAD-146r2, November 1996.

[7] http://www.timeconsult.com/

[8] A. Steiner. A Generalisation Approach to Temporal Data Models and their Implementations, PhD thesis, 1998.

[9] M. Bohlen, C.Jensen, and R. Snodgrass. Evaluating the completeness of TSQL2. In J. Clifford and A. Tuzhilin, editors, Recent Advances in Temporal Databases,1995, pages 153-172.

[10] M. Bohlen and R. Marti. On the Completeness of Temporal Database Query Languages. In Proceedings of the First International Conference on Temporal Logic, July 1994, pages 283-300.