

УДК 004.658.2

Ю.А. КОРОЛЬ, А.Б. ЛЕЩЕНКО, Ю.А. ЛЕЩЕНКО

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина

ИСПОЛЬЗОВАНИЕ БИТЕМПОРАЛЬНЫХ БАЗ ДАННЫХ В ИНФОРМАЦИОННЫХ УПРАВЛЯЮЩИХ СИСТЕМАХ

Проанализированы подходы к реализации темпоральных систем, особенности и методы создания битемпоральных баз данных. Определены основные способы представления темпоральных систем, использующих современные базы данных: реализация темпоральной поддержки на уровне приложения и расширение нетемпоральной модели до темпоральной. Предложен подход к реализации битемпоральной модели на основе объектной модели данных, которая содержит как модельное, так и транзакционное время. Представлены фрагменты реализации битемпоральной модели с использованием системы управления базами данных Caché.

Ключевые слова: темпоральная база данных, битемпоральная модель данных, транзакционное время, модельное время.

Введение

С ростом информатизации общества всё большее значение приобретают системы, контролирующие и ограничивающие доступ пользователей к информационным ресурсам. Процесс совершенствования систем безопасности информационных ресурсов изменяется с развитием информационных технологий и техники. Это развитие позволяет усовершенствовать использование информационных ресурсов, где хронологическим данным соответствуют некоторые характеристики, которые раньше были невозможны, например, динамики изменения параметров, практически неограниченный срок хранения данных и объем хранимой информации.

Законодательство Украины, а также многих стран обязывает хранить все бизнес-данные в публичных компаниях за определенный срок, что и обуславливает необходимость хранения информации, поступающей в информационные системы.

Темпоральные базы данных (ТБД), помимо обеспечения гибкой работы с хронологическими данными оперативных систем, играют большую роль в аналитических системах, так как накопленные исторические данные представляют огромную ценность для принятия решений, выявления тенденций развития, планирования и прогнозирования деятельности предприятий. Поэтому тема представленной публикации является достаточно актуальной, так как в ней предложено использование битемпоральных баз данных в информационных управляющих системах.

Постановка задачи исследования

Если рассматривать данные, представленные в БД, в качестве некоторого отражение текущего со-

стояния действительности о моделируемом мире, каждая запись может быть воспринята как некоторый факт, который является истинным в определенный момент времени. При переходе к ТБД для каждого факта можно указать промежуток времени, в течение которого этот факт будет истинным в моделируемом мире. Представление времени, когда с данными связывается промежуток времени их актуальности (с точки зрения моделируемого мира), называется модельным или действительным временем. Другим типом линии времени в теории ТБД является транзакционное время. В любой системе управления базами данных (СУБД) каждой записи базы данных можно сопоставить некоторый промежуток времени, когда данная запись была создана и/или удалена из БД.

Несмотря на то, что многие приложения успешно функционируют на основе традиционных СУБД, существующих возможностей для сохранения и обработки исторических данных недостаточно для информационных систем, динамика изменения информации в которых является одним из ключевых моментов. В работах [2, 3] представлен подход к реализации темпоральности, используя только модельное время на основе реляционной базы данных, что обладает рядом недостатков, которые сдерживают их применение на практике. Использование этих моделей в системах, в которых со временем необходимо изменять структуру данных, приводит к увеличению избыточности хранения информации, а также к разрастанию базовой реляционной модели, в которой усложняется составление, выполнение запросов и адаптация базы данных к новым условиям. В предложенной работе рассматривается новый метод построения битемпоральных баз данных (БТБД). Битемпоральная модель данных оперирует

как модельным, так и транзакционным временем. Именно битемпоральная модель является наиболее востребованной моделью в большинстве информационных управляющих систем (ИУС) (рис. 1).

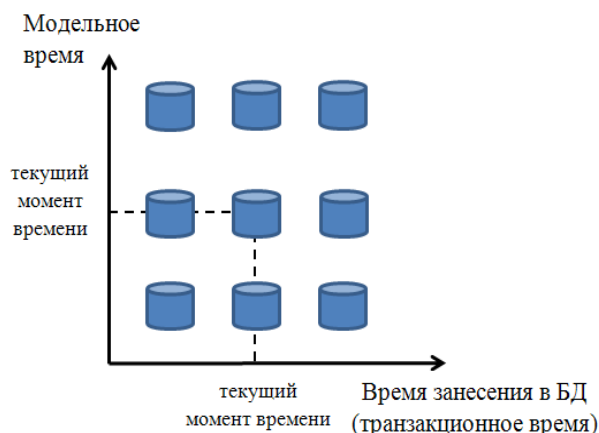


Рис. 1. Битемпоральное представление данных

Целью данной работы является – формирование нового метода создания БТМД на основе объектной модели, которая обеспечивает высокопроизводительную обработку данных в ИУС.

Анализ особенностей и методов создания битемпоральных баз данных.

В отличие от традиционных моделей данных, обеспечивающих хранение лишь “мгновенного снимка” объектов предметной области, темпоральные модели данных позволяют хранить информацию об эволюции объектов: для любого объекта, который был создан в момент времени T_1 и закончил свое существование в момент времени T_2 , в базе данных (БД) будут сохранены все его состояния на временном интервале $[T_1, T_2]$ [4].

Темпоральная модель данных (ТМД) – это модель данных, ориентированная на хранение темпоральных данных, все аспекты которой также должны быть темпоральными. Традиционную модель данных можно представить в виде трёх компонент:

$$M = (DS, OP, C),$$

где DS – структура данных, OP – операции, C – ограничения целостности.

Темпоральная модель данных [5]:

$$MT = (DST, OPT, CT)$$

должна поддерживать все понятия, входящие в каждую из трех компонент, с учетом изменений данных во времени (T), и хранить темпоральные данные. Алгебра и операции модификации должны быть переопределены, используя темпоральные обозначения. Дополнительно, для каждого ограничения целостности в нетемпоральной модели данных (M),

темпоральная модель данных (ТМД) должна поддерживать темпоральный аналог нетемпорального ограничения.

Разработка ТМД предполагает формирование следующих составляющих:

- темпоральной структуры данных;
- темпорального ограничения целостности, ключей;
- темпоральных запросов и модификаций данных;
- темпоральной алгебры.

Для исследования ТМД выделим три фундаментальных типа темпоральных данных:

- момент времени (instant) (событие, которое произошло или произойдет в определенный момент времени, например, сейчас или 1 августа 2013 года в 17.57);
- интервал времени (interval) (длительность временного отрезка, например, 2 года, начиная с указанной даты);
- период времени (period) (конкретный отрезок времени, например, с 23 апреля 2011 года по 1 августа 2013 года).

На основе выделенных типов темпоральных данных могут быть разработаны смешанные типы данных (например, моменты времени с определенной периодичностью).

Фактически, на практике, существуют два принципиальных подхода к реализации ТМД:

- реализация темпоральной поддержки на уровне приложения;
- расширение нетемпоральной модели данных до темпоральной.

Метод реализации темпоральности на уровне приложения предполагает разработку специальных собственных средств поддержки темпоральности на уровне приложения. Однако на практике данный подход приводит к существенным проблемам, например, когда требуется изменить или заменить часть кода в приложении. Темпоральная семантика, в таком случае, проектируется каждым разработчиком заново. Темпоральная логика, реализованная на уровне приложения, может быть удобным сиюминутным решением, но не дальновидной стратегией проектирования ИС.

Расширение нетемпоральной модели данных до темпоральной модели означает, что для спецификации темпоральных понятий используются основные концепции, поддерживаемые нетемпоральной моделью данных. Язык запросов и алгебра расширяются дополнительными операциями для того, чтобы иметь возможность описывать темпоральные операции с данными. На практике этот подход расширения схемы данных наиболее широко используется

для построения ТМД. Преимуществами такого метода является изменение лишь отдельных частей модели, например, языка запросов или ограничения целостности. Метод доступа к информации и структура данных остаются неизменными.

В рамках данного подхода были предложены различные ТМД. Принципиальные отличия этих моделей друг от друга можно разделить по следующим критериям:

- тип темпоральных данных (дискретное или интервальное представление времени);
- обеспечение темпоральности на уровне отдельных атрибутов или на уровне кортежа.

Первый критерий говорит о том, что темпоральные данные могут быть связаны как с дискретным представлением времени – моментом времени, так и с интервальным представлением.

Второй критерий построения темпоральных моделей данных приводит к появлению нескольких различных моделей данных.

Ввиду отсутствия на сегодняшний день полноценных битемпоральных СУБД [2, 3], для реализации битемпоральных возможностей в рамках ИУС, программистам приходится разрабатывать специальные средства, расширяющие и дополняющие существующие реляционные модели. Весьма распространенной проблемой разработки таких приложений является отсутствие полного понимания того, каким образом и на каком уровне должна быть осуществлена поддержка темпоральности в БД. Многими разработчиками, реализующими битемпоральность в ИС, не учитывается тот факт, что за несколько десятилетий работы в данной области исследований, накоплено множество различных подходов и методик, изучение которых помогло бы избежать многих традиционных ошибок и заблуждений.

В данной работе за основу была принята модель представления темпоральных данных, описанная Р. Снодграсом [6].

Пусть битемпоральное отношение BR имеет набор атрибутов A_1, \dots, A_n, T , где T — темпоральный атрибут, определенный на множестве темпоральных элементов. Тогда битемпоральное отношение можно записать в следующем виде:

$$BR = (A_1, \dots, A_n, T_s, T_e, V_s, V_e).$$

Набор дополнительных атрибутов T_s, T_e, V_s, V_e состоит из атомарных битемпоральных атрибутов времени, содержащих дату начала и окончания транзакционного (T_s, T_e) и модельного времени (V_s, V_e) . Данная модель данных является самым естественным и наиболее часто используемым способом представления битемпоральных отношений.

Описание математической модели

В качестве модели данных предлагается использовать открытую модель с абстрактным идентификатором объекта, представленную на рис. 2. В этой модели жизненный цикл объекта описывается через жизненные циклы всех его свойств, определенных в разных отношениях и имеющих временные атрибуты $\{T_s, T_e\}$, определяющие соответственно время начала и окончания жизненного цикла.

Предполагается, что объект битемпоральной системы ОВ характеризуется уникальным абстрактным идентификатором K^s , набором атрибутов АО и может быть представлен в виде:

$$OB = \{K^s, AO\}.$$

Набор атрибутов АО, в свою очередь, делится на множество статических атрибутов SA^s , не подверженных изменениям во времени, и множество динамических атрибутов DA, изменяющихся во времени:

$$AO = SA^s \cup DA.$$

В модели не будем рассматривать множественные атрибуты, которые присутствуют в множествах SA^s и DA, так как это в рассматриваемом нами контексте не принципиально и только приведет к излишнему усложнению модели.

Представим битемпоральную базу данных ИУС в виде совокупности взаимосвязанных классов:

$$BD = \{BR^s, BR_1^d, BR_2^d, \dots, BR_n^d, \\ \text{TablKernelMod}, \text{TablKernelTran}\},$$

где $BR^s = \{K^s, SA^s, MO, MD, MS\}$ – родительский класс, K^s – абсолютный идентификатор объекта, SA^s – статические атрибуты объекта и методы работы с битемпоральными данными: MO – добавления, MD – удаления, MS – изменения данных;

$BR_1^d, BR_2^d, \dots, BR_n^d$ – наследуемые классы от абстрактного класса BR^s , описывающие дискретно изменяющиеся динамические атрибуты предметной области $DA = \{DA_1, DA_2, \dots, DA_n\}$ и имеющие схемы $BR_i^d = \{K^s, K_i^d, DA_{i1}, \dots, DA_{ik}, \text{TempProperty}_i, \text{TempValue}_i, \text{TempType}_i\}$, $i=1, 2, \dots, n$, в которых (K^s, K_i^d) – первичный ключ (K_i^d – часть первичного ключа BR_i^d), $\{DA_{i1}, \dots, DA_{ik}\} \subseteq DA$ – дискретные динамические атрибуты. В классах BR_i^d определены:

- параметр $\text{TemporalProperty}_i$, который описывает темпоральные свойства объекта

$$\text{TemporalProperty}_i \subseteq \{DA_{i1}, \dots, DA_{ik}\};$$

- параметр TemporalType_i , содержащий типы темпоральности каждого из свойств, описанных в $\text{TemporalProperty}_i$. Каждое свойство может принимать одно из указанных типов темпоральности $\{\text{instant}, \text{interval}, \text{period}\}$;

- параметр TemporalValue_i, содержащий значение по умолчанию каждого атрибута описанного в TemporalProperty_i (например, "0", текущее время, вычисление значения, взять из базы данных и т.д.).

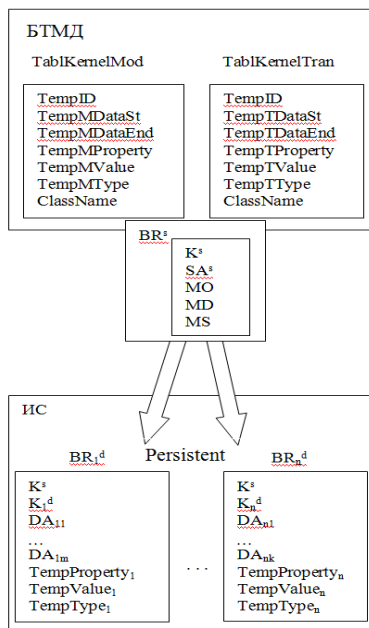


Рис. 2. Модель данных с абстрактным идентификатором объекта

Класс TablKernelMod хранит модельные данные:

- время начала (TempMDataSt) и время окончания модельного времени (TempMDataEnd),
- хранимое свойство (TempMProperty) из класса ClassName,
- значение (TempMValue) свойства хранимого в TempMProperty,
- тип темпорального атрибута (TempMType) из TempMProperty, содержащегося в классе ClassName.

Класс TablKernelTran имеет структуру, аналогичную классу TablKernelMod и хранит транзакционные данные.

Формирование битемпоральных данных осуществляется в момент обращения пользовательских приложений к соответствующим классам (BR₁^d ... BR_n^d). При доступе, записи, изменении и удалении пользовательских данных вызываются методы доступа, записи, изменения и удаления данных, которые вносят соответствующие значения в БТБД.

Особенности реализации битемпоральной модели данных

Основой для создания метода БТМД были приняты рассмотренные выше принципы, особенности построения ТБД и математическое описание модели.

1. Создание абстрактного класса обработчика (BR^s):

```
Class DATA.CORE.Kernel Extends (...) [ Abstract, ... ]
Property OpenLog [ MultiDimensional, Private ];
Property recordDate As %Date(ZENHIDDEN = 1, ZENLABEL = "Дата записи") [ InitialExpression = {+$H} ];
Method %OnOpen() As %Status {...}
Method %OnAfterSave(insert As %Boolean) As %Status {...}
ClassMethod %OnDelete(oid As %ObjectIdentity) As %Status {...}
```

2. В абстрактном классе переопределяем методы:

добавления данных (MO):

- добавления данных (MO):

```
Method %OnOpen() As %Status [ Private,
ServerOnly = 1 ] { s classname = ..%ClassName(1)
Set cdef = ##class(%Dictionary.ClassDefinition).%OpenId(classname)
s pp = $zobjclassmethod(classname, "%Open", ..%Oid())
// get list of properties
Set count = cdef.Properties.Count()
For i = 1:1:count { s prop=cdef.Properties.GetAt(i).Name
if !.#TemporalProperty[prop {
s ..OpenLog(..%Oid(),cdef.Properties.GetAt(i).Name)= $zobjproperty(pp, prop)}}
Quit $$$OK;
```

- изменения данных (MS):

```
Method %OnAfterSave(insert As %Boolean) ...
// get list of properties
Set count = cdef.Properties.Count()
if insert=1 {
For i = 1:1:count {
s prop=cdef.Properties.GetAt(i).Name
s val = $zobjproperty(pp, prop)
if !.#TemporalProperty[prop {
s KernelMod =
##class(DATA.CORE.TablKernelMod).%New()
s KernelMod.TempClass=classname
s KernelMod.TempID=ID
s KernelMod.TempDataSt=ts
s KernelMod.TempProperty=prop
s KernelMod.TempValue=val
s KernelMod.TempType="toch"
do KernelMod.%Save()
s KernelTran =
##class(DATA.CORE.TablKernelTran).%New()
s KernelTran.TempClass=classname
s KernelTran.TempData=ts
s KernelTran.TempID=ID
s KernelTran.TempProperty=prop
s KernelTran.TempType="i"
s KernelTran.TempValue=val
do KernelTran.%Save()
}}
else { ...}
Quit $$$OK;
```

- удаления данных (MD):

```
ClassMethod %OnDelete(oid As %ObjectIdentity) As %Status [ Private,
ServerOnly = 1 ]
...
// get list of properties
Set count = cdef.Properties.Count()
For i = 1:1:count {
s prop=cdef.Properties.GetAt(i).Name
if !.#TemporalProperty[prop {
s val = $zobjproperty(pp, prop)
s KernelTran =
##class(DATA.CORE.TablKernelTran).%New()
s KernelTran.TempClass=classname
s KernelTran.TempDataEnd=$zdt("99999,9999",3)
s KernelTran.TempID=ID
s KernelTran.TempProperty=prop
s KernelTran.TempType="d"
s KernelTran.TempValue=val
do KernelTran.%Save() }... Quit $$$OK}.
```

4. От абстрактного класса наследуем дочерние

классы (BR_i^d), что позволяет наследовать методы для добавления, изменения и удаления темпоральных данных в битемпоральной базе данных.

5. В дочернем классе (BR_i^d), где необходимо обеспечить битемпоральность, добавляем параметры, содержащие список темпоральных свойств объекта, тип темпоральности и значение по умолчанию:

```
Parameter TemporalProperty As %String;
Parameter TempValue As %String;
Parameter TemporalType As %String.
```

Заключение

В результате выполненных исследований была представлена и реализована новая битемпоральная модель данных для ИУС, которая включает в себя как модельное, так и транзакционное время.

Преимуществом предложенной битемпоральной БД является возможность интеграции модельной и транзакционной БД. Это позволяет получить полный доступ к истории изменений и оригинальным значениям атрибутов, что особенно важно для современных ИУС предприятий.

Поступила в редакцию: 3.09.2013, рассмотрена на редколлегии 11.09.2013

Рецензент: д-р техн. наук, проф., проф. каф. программной инженерии И.В. Шостак, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

ВИКОРИСТАННЯ БІТЕМПОРАЛЬНИХ БАЗ ДАНИХ В ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМАХ

Ю.О. Король, О.Б. Лещенко, Ю.О. Лещенко

Проаналізовано підходи до реалізації темпоральних систем, особливості та методи створення битемпоральних баз даних. Визначено основні способи подання темпоральних систем, що використовують сучасні бази даних: реалізація темпоральної підтримки на рівні додатку і розширення нетемпоральної моделі до темпоральної. Запропоновано підхід до реалізації битемпоральної моделі на основі об'єктної моделі даних, яка містить як модельний, так і транзакційний час. Представлено фрагменти реалізації битемпоральної моделі з використанням системи управління базами даних Cache.

Ключові слова: темпоральна база даних, битемпоральна модель даних, транзакційний час, модельний час.

USE BITEMPORAL DATABASE IN INFORMATION MANAGEMENT SYSTEM

Y.O. Korol, O.B. Leshchenko, Y.O. Leshchenko

Approaches to the implementation of temporal systems based on object database, as well as the features and methods for creating bitemporal databases. The basic ways of presenting temporal systems using modern object databases: the implementation of temporal support at the application level and the expansion nontemporal object model to temporal. An approach to the implementation of the bitemporal model based on object data model that contains both the model and the transactional time.

Keywords: temporal database, bitemporal data model, transaction time, range time.

Король Юлія Александровна – аспірант каф. інформаційних управляючих систем, Національний аерокосмічний університет ім. Н.Е. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Лещенко Александр Борисович – канд. техн. наук, доцент, доцент каф. інформаційних управляючих систем, Національний аерокосмічний університет ім. Н.Е. Жуковського «Харківський авіаційний інститут», Харків, Україна, e-mail: lesch@xai.edu.ua.

Лещенко Юлія Александровна – інженер каф. інформаційних управляючих систем, Національний аерокосмічний університет ім. Н.Е. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Литература

1. Закон України о защите личных данных [Электронный ресурс]. – Режим доступа: <http://zakon.rada.gov.ua/cgi-bin/laws/main.cgi?nreg=2297-17>. – 03.05.2012.
2. Балдин, А.В. Обзор способов построения темпоральных систем на основе реляционной базы данных [Текст] / А.В. Балдин, Д.В. Елисеев, К.Г. Агаян // Научно-техническое издание «Наука и образование». – 2012. – №8. – С. 309-318.
3. Панченко, Б.Е. Рекурсивные связи и темпоральность в реляционном каркасе - маски сущностей-объектов [Текст] / Б.Е. Панченко // Проблемы управления и информатики. – 2013. – №2. – С. 92-104.
4. Snodgrass, R. Developing Time-Oriented Database Applications in SQL [Text] / R. Snodgrass. – Morgan Kaufmann Publishers, 1999. – 504 p.
5. Jensen, C.S. Unifying Temporal Data Models Via a Conceptual Model [Text] / C.S. Jensen, M.D. Soo, R.T. Snodgrass // Information Systems. – 1994. – Vol 19, No. 7. – P. 513-547.
6. Snodgrass, R. A Taxonomy of Time in Databases: In Proceedings of ACM SIGMOD [Text] / R. Snodgrass, I. Ahn, 1985. – 90 p.