

basis of chosen enterprise policy. The application can be used by managers for acceptance of the investment decisions in the theme of commercial enterprise development strategy.

The work on improvement the system is continued. The decision support system, using OLAP-technologies, will be proposed.

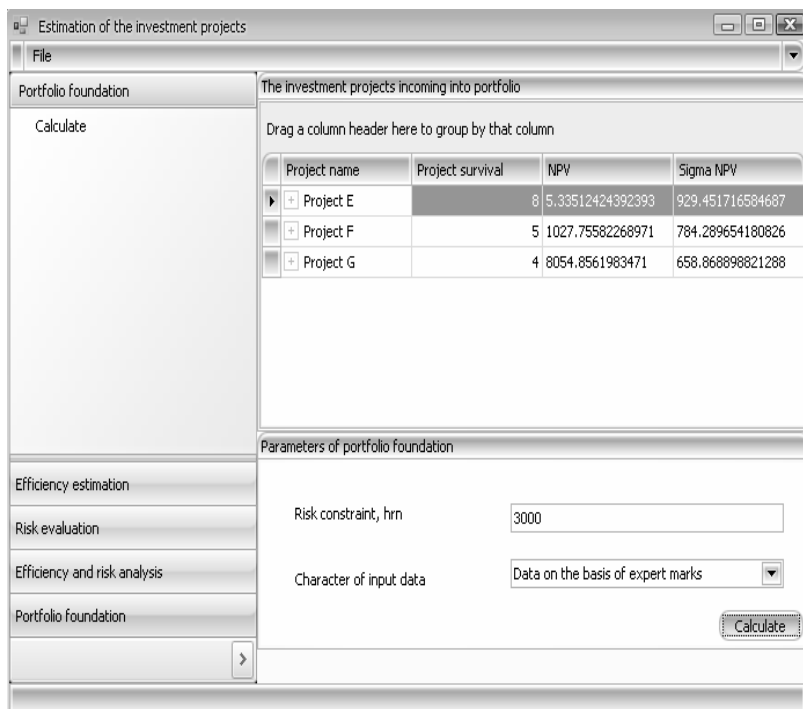


Fig. 6. Screen form of investment projects portfolio foundation

**References:** 1. G. Birman, S. Shmidt Economic Analysis of the Investment Projects/Transl. from eng. edit. L.P. Belykh. – M.: Banks and Exchanges, UNITI, 1997. – 631 p. 2. D.A. Endovitskiy Complex Analysis and Control of the Investment Activity: methodology and practice / edit. prof. L.T. Gilyarovskaya. – M.: Finances and Statistics, 2001. – 400 p. 3. A. Kofman Introduction into Fuzzy Sets Theory. – M.: Radio and Communication, 1982. – 432 p. 4. G. Booch, J. Rambaugh, I. Jacobson The Unified Modeling Language User Guide, Addison-Wesley, 1998. 5. P.P. Ed Chen Entity-Relationship Approach to Information Modeling and Analysis, Amsterdam, North-Holland, 1993. 6. J. Richter CLR via C# Programming on C# Microsoft.Net Framework 2/0 platform, 2007. – 656 p.

Поступила в редколлегию 27.02.08

УДК 681.3.06

А.А. БЛАЖКО, канд. техн. наук, ОНПУ, Одесса, Украина  
Дж.Т. Д. АЛЬСАФФАДИ, аспирант, ОНПУ, Одесса, Украина

## ТИРАЖИРОВАНИЕ ОПЕРАЦИЙ СОГЛАСОВАНИЯ В ГЕТЕРОГЕННЫХ РАСПРЕДЕЛЕННЫХ БАЗАХ ДАННЫХ

У статті розглядається проблема проектування розподілених баз даних з тиражуванням, які складаються з множини локальних баз даних, що містять логічно різні, але семантично еквівалентні структури даних, та знаходяться під керуванням систем баз даних різних виробників. Запропоновано розширення моделі процесу тиражування даних, яке включає: відповідності між БД з різними структурами, правила перетворення даних та функціональні правила обміну даними та між цими базами даних.

There is described development problem of replication databases, which consist of local databases with logically different, but semantic equivalent data structures, and managed by database systems of different vendors. There is offered expansion replication data process model, which includes: schema matching between databases with different structures, rules data conversion and functional rules data exchange between databases.

**Введение.** Корпоративная информационная система (ИС), которая разрабатывается на протяжении десятилетий, часто управляет данными, расположенными на множестве автономных (локальных) баз данных (ЛБД). При этом все БД могут содержать логически разные, но семантические эквивалентные структуры данных и поддерживаться системами управления (СУБД) от разных производителей. Для обеспечения согласованности данных в ИС необходимо поддерживать тиражирование операций согласования данных в гетерогенной распределенной базе данных (РБД), что является трудоемкой задачей при отсутствии соответствующих инструментальных средств. Существующие системы тиражирования *OpenSource*-типа, которые являются расширением стандартных библиотек доступа к БД, требуют перепроектирования существующих программ [1], а специализированные не поддерживают гетерогенности [2]. Для объединения свойств этих систем была разработана система [3]. При увеличении размерности ЛБД и числа типов СУБД процесс администрирования гетерогенной РБД становится трудоемким. В статье предлагается механизм сокращения трудоемкости за счет формализации процессов гетерогенного тиражирования операций согласования ЛБД.

**1. Расширенная модель процесса гетерогенного тиражирования данных.** Для решения поставленной задачи предлагается на основе модели

РБД создать формализованную структурную модель процесса гетерогенного тиражирования данных, которая должна включать описания:

- соответствия между ЛБД с разными структурами, которые входят в РБД;
- правил преобразования данных между ЛБД с разными структурами на основе их структурной модели соответствия;
- функциональных правил обмена данными между ЛБД под управлением СУБД разных производителей без учета встроенных систем тиражирования.

Представим расширенную модель процесса гетерогенного тиражирования операций согласования ЛБД в виде семерки

$\langle LS, DS, RS, SM, TT, AQ, MQ, \rangle$ ,

где  $LS$  (*Local Schema*) – взвешенный граф описания структуры ЛБД;

$DS$  (*Distributed Schema*) – множество элементов-описаний РБД;

$RS$  (*Replication Schema*) – множество элементов-описаний схемы тиражирования;

$SM$  (*Schema Matching*) – множество элементов-описаний соответствия между логическими структурами ЛБД;

$TT$  (*Trigger Template*) – множество элементов-описаний шаблонов программного кода триггеров СУБД, которые согласовывают ЛБД;

$AQ$  (*Asynchronous Queue*) – упорядоченное множество элементов-описаний операций асинхронного согласования ЛБД;

$MQ$  (*Meta Query*) – множество элементов-описаний *SQL*-запросов к СУБД для получения мета-описания логической структуры ЛБД.

Элемент  $ls \in LS$  представим в виде тройки

$\langle R, L, SR \rangle$

где  $R$  – множество вершин описаний структур таблиц БД, элемент  $r_i = \langle rname, aname, atype, ftype \rangle$ , где  $rname$  – имя таблицы,  $aname$  – имя атрибута,  $atype$  – тип атрибута,  $ftype$  – признак вхождения атрибута в первичный ключ;

$L$  – множество дуг-связей между таблицами БД, элемент  $l_{ij} \in L$ , если  $\exists r_i \in R_i$  и  $r_j \in R_j$ ;  $ftype_i = PK$ ,  $ftype_j = FK(r_i)$ , где  $PK$  – *Primary Key*,  $FK$  – *Foreign Key*;

$SR$  – упорядоченное множество описания иерархической зависимости номеров таблиц, которое определяет порядок выполнения запросов на внесение данных в таблицы (или удаление данных из таблиц) и учитывает связи между таблицами из множества  $L$ .

Элемент  $sr \in SR$  представим в виде двойки  $\langle r_i, level \rangle$ , где  $r_i \in R$ ,  $level$  – порядковый номер, определяющий уровень иерархической зависи-

мости таблиц, при этом таблица, которая не связана с другими таблицами по связи *Foreign Key*, характеризуется нулевым уровнем.

Элемент  $ds \in DS$  представим в виде двойки  $\langle dbname, dbtype \rangle$ , где  $dbname$  – физическое имя ЛБД и сетевой адрес расположения ее СУБД;  $dbtype$  – тип СУБД (например, *oracle, postgresql, mysql, firebird*).

Для обеспечения работы системы тиражирования в условиях гетерогенности структур ЛБД предлагается в каждой ЛБД создавать копии-таблицы из других ЛБД. Процесс преобразования содержания операций согласования между таблицей-издателем и таблицей-подписчиком с разными структурами зависит от свойства СУБД, управляющей ЛБД-издателем.

Если СУБД обладает механизмами перехвата с передачей в другие СУБД, то выполняется следующая последовательность действий:

- преобразование операции для согласования копии таблицы-подписчика в ЛБД с использованием преобразующего (гетерогенного) триггера (П-триггер);
- передача операции согласования между таблицей-копией в ЛБД-издателя и таблицей-подписчиком ЛБД-подписчика с использованием согласующего (гомогенного) триггера (С-триггер).

Если СУБД обладает механизмами перехвата без передачи в другие СУБД, то выполняется следующая последовательность действий:

- преобразование операции для сохранения в универсальном журнале хранения операций с использованием триггеров журналирования;
- запаздывающее восстановление операции согласования из журнала;
- перенос операции в таблицу-копию в ЛБД-подписчика с использованием С-триггера;
- преобразование операции для согласования таблицы-подписчика в ЛБД с использованием П-триггера.

Элемент  $rs \in RS$  представим в виде четверки

$\langle pubdb, subdb, ro, trtype \rangle$ ,

где  $pubdb \in DDB$ ,  $subdb \in DDB$ , – ЛБД-издатель, ЛБД-подписчик, соответственно;

$ro \in R$  – имя тиражируемой таблицы ЛБД;

$trtype$  – метод передачи операций согласования (синхронный, асинхронный-*pull*, асинхронный-*push*).

Процесс согласования РБД при синхронном методе передачи операций допускает использование протокола двухфазной фиксации изменений, когда у ЛБД-издателя операция модификации завершается одновременно с операциями согласования, отправленными в ЛБД-подписчики. В асин-

хронном-*pull* методе инициатором процесса согласования является ЛБД-подписчик, а при асинхронном-*push* методе – ЛБД-издатель.

Элемент  $sm \in SM$  представим в виде двойки  $\langle sa, AM \rangle$ , где  $sa \in R$  – атрибут БД-подписчика;  $AM$  – упорядоченное множество преобразований с элементами  $F(pa)$ ,  $pa \in R$  – атрибут ЛБД-издателя,  $F$  – функция преобразования атрибута.

Элемент  $tt \in TT$  представим в виде тройки  $\langle rtype, trtype, def \rangle$ , где  $rtype$  – тип СУБД (например, *oracle*, *postgresql*, *mysql*, *firebird*);  $trtype$  – метод передачи операций согласования (синхронный, асинхронный);  $def$  – описание шаблона программного кода триггера. Элемент  $mq \in MQ$  представим в виде тройки  $\langle mqtype, rtype, metaSQL \rangle$ , где  $mqtype$  – подкласс метаописания  $mqtype = \{ "struct", "link" \}$ , ("struct" – мета-описание структуры таблицы, "link" – мета-описание связей между таблицами);  $rtype$  – тип СУБД;

**2. Автоматизация процессов управления.** Для автоматического заполнения элемента  $LS$  заданной ЛБД, управляемой СУБД типа  $type$ , предлагается алгоритм со следующей последовательностью шагов.

Шаг 1. Получение из множества  $MQ$  элемента  $mq = \langle mqtype, rtype, metaSQL \rangle$ , для которого  $rtype = type$  и  $mqtype = "struct"$ .

Шаг 2. Выполнение запроса из  $metaSQL$  и сохранение результата в  $result$ .

Шаг 3. Формирование множества вершин  $R$  на основании  $result$ .

Шаг 4. Получение из множества  $MQ$  элемента  $mq = \langle mqtype, rtype, metaSQL \rangle$ , для которого  $rtype = type$  и  $mqtype = "link"$ .

Шаг 5. Выполнение запроса из  $metaSQL$  и сохранение результата в  $result$ .

Шаг 6. Формирование множества дуг  $L$  на основании  $result$ .

Для автоматического заполнения множества  $SR$  заданной ЛБД предлагается алгоритм, который рекурсивно просматривает содержимое множества  $L$ , начиная с таблиц из множества  $RM = \{rm_i\} \subseteq R: \exists l_{ij} \in L$ . Для указанных таблиц устанавливается  $level = 0$ . Для таблиц, имеющих связи с данными таблицами, устанавливается  $level$  на единицу больший.

Для автоматического создания программного кода триггеров в таблице  $table$  заданной ЛБД, управляемой СУБД типа  $type$ , предлагается алгоритм со следующей последовательностью шагов.

Шаг 1. Получение из множества  $TT$  описания заголовка программного кода триггера для  $rtype=type$ .

Шаг 2. Получение из множества  $TT$  для  $rtype=type$  описания шаблона операции сравнения атрибутов и внесения в  $AQ$ -очередь.

Шаг 3. Получение из множества  $R$  списка атрибутов с  $rname = table$ .

Шаг 4. Для каждого атрибута формирование операции сравнения атрибутов и внесения в  $AQ$ -очередь на основании шаблона.

Шаг 5. Получение из множества  $TT$  для  $rtype=type$  описания завершения программного кода триггера.

Шаг 6. Регистрация программного кода триггера в СУБД.

**3. Система тиражирования в гетерогенной базе данных.** На основе предложенной модели разработано программное обеспечение системы гетерогенного тиражирования операций согласования, схема работы которого представлена на рисунке 1. Анализ программного обеспечения доступа в БД (*ODBC*, *JDBC*, *DBI*, *BDE*, *ADO*), выполненный по критерию кросс-платформенности, *OpenSource*, производительности и поддержки в программном коде триггеров СУБД, определил в качестве программной архитектуры библиотеку *DBI* языка *Perl*, которая предоставляет программные шлюзы для большинства СУБД. При этом лишь СУБД *PostgreSQL* обладает *OpenSource*-механизмами перехвата операций модификации ЛБД с передачей в другие СУБД. Остальные СУБД (*non-PostgreSQL*) обладают *OpenSource*-механизмами перехвата операций модификации ЛБД без передачи в СУБД других типов.

Система гетерогенного тиражирования включает модули:

- модуль согласования РБД методом «мгновенного снимка» (*snapshot*);
- модуль генерации программного кода триггеров по перехвату операций модификации в ЛБД, которые обеспечивают асинхронный-*pull* метод согласования на основе множеств  $LS$ ,  $DS$ ,  $SM$ ,  $RS$ ,  $TT$ ;
- модуль генерации программного кода синхронного тиражирования в виде триггеров по перехвату операций модификации в ЛБД и их синхронной передачи в удаленные ЛБД на основе множеств  $LS$ ,  $DS$ ,  $SM$ ,  $RS$ ,  $TT$ ;
- модуль согласования ЛБД по асинхронному-*push* методу операциями из множества  $DQ$ ;
- модуль согласования ЛБД по асинхронному-*pull* методу операциями из множества  $AQ$ , которые были отложены на заданный период или из-за недоступности ЛБД-подписчика.
- БД шаблонов программного кода триггеров содержит описания:
- синхронного и асинхронного-*push* методов передачи операций согласования для СУБД *PostgreSQL* на основе библиотеки *Perl DBI-DBD*;
- асинхронного-*pull* метода передачи операций согласования для СУБД *Oracle v.10g*, *PostgreSQL v.8.2*, *MySQL v.5.0*, *FireBird v.2.0*.

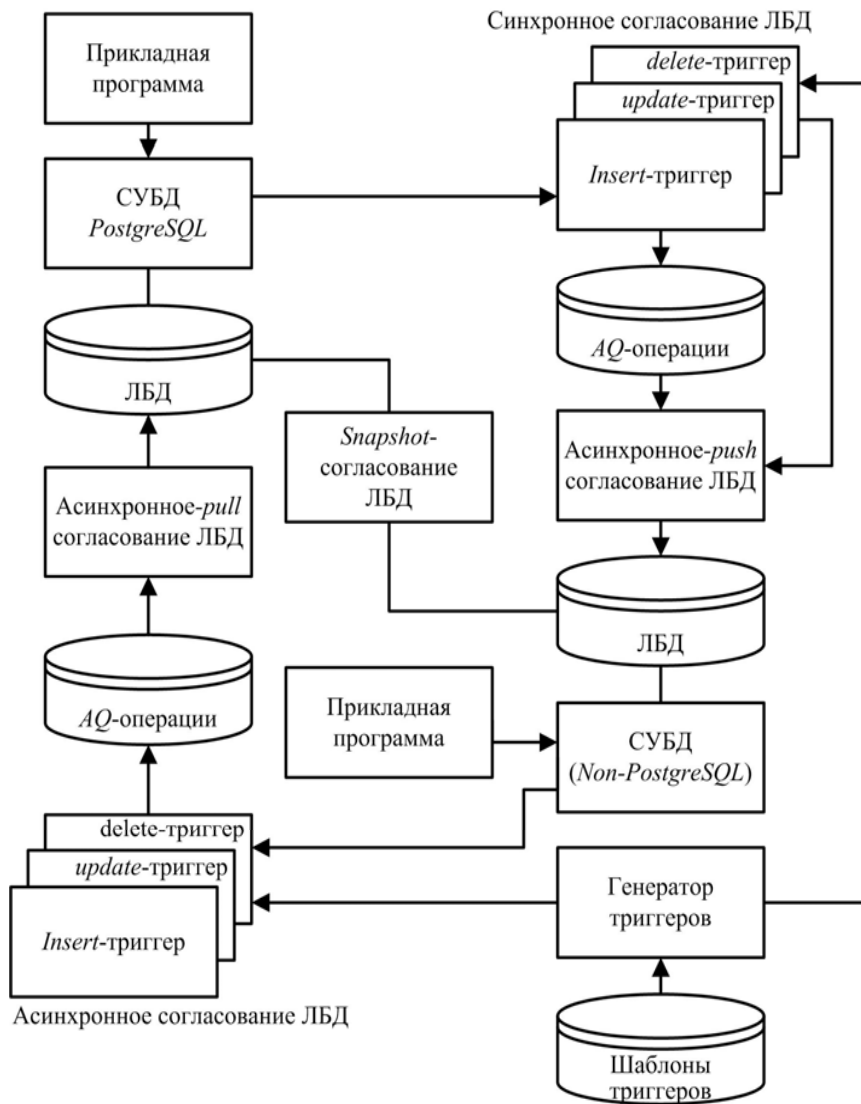


Рис. 5. Система тиражирования гетерогенной базы данных

**Выводы.** Разработанная система проходит тестовую эксплуатацию в ИС Вуза ОНПУ, которая содержит программы под управлением разных СУБД: «Отдел кадров» (СУБД PostgreSQL v.8.2), «Библиотека» (СУБД Oracle v.10g), «Бухгалтерия» (СУБД FireBird v.2.0). Для автоматической генерации кода преобразующих триггеров согласования РБД потребовалось ручное формирование соответствия между структурами данных БД «Отдел кадров», «Библиотека» и «Бухгалтерия». Структурная формализация процесса гетерогенного тиражирования данных, предложенная в работе, позволила провести унификацию программных компонент тиражирования для СУБД разных производителей и создать основу для последующего создания алгоритмов согласования ЛБД с разными структурами. Использование системы не требует перепроектирования прикладных программ, работающих с ЛБД, поэтому работа системы обеспечивает прозрачность тиражирования.

В дальнейшем для автоматизации процесса установления соответствия между структурами данных ЛБД предполагается разработать алгоритм анализа SQL-запросов формирования объектов тиражирования в РБД.

**Список литературы:** 1. C. Plattner, G.Alonso. Ganymed: Scalable Replication for Transactional Web Applications // The Proc. of 5th ACM/IFIP/USENIX International Middleware Conference, 2004. 2. S.Wu and B. Kemme. Postgres-R(SI): Combining Replica Control with Concurrency Control Based on Snapshot Isolation // Proc. 21st International Conference on Data Engineering. - P. 422 – 433. 3. Джайяб Т.Д. Альсаффади, О.А.Блажко Підтримка гетерогенних розподілених баз даних з тиражуванням // The Procs. of International Conferences on Computer Science and Information Technologies. – Lviv, Ukraine, 2007. – С. 237–240

Поступила в редколлегию 19.02.08