Krassimir Markov, Vitalii Velychko, Oleksy Voloshin
(editors)

Natural and Artificial Intelligence

ITHEA
SOFIA
2010

Krassimir Markov, Vitalii Velychko, Oleksy Voloshin (ed.)

Natural and Artificial Intelligence

ITHEA®

Sofia, Bulgaria, 2010

ISBN 978-954-16-0043-9

First edition

Recommended for publication by The Scientific Concil of the Institute of Information Theories and Applications FOI ITHEA

This book is engraved in prof. Zinoviy Lvovich Rabinovich memory. He was a great Ukrainian scientist, co-founder of ITHEA International Scientific Society (ITHEA ISS). To do homage to the remarkable world-known scientific leader and teacher this book is published in Russian language and is concerned to some of the main areas of interest of Prof. Rabinovich.

The book is opened by the last paper of Prof. Rabinovich specially written for ITHEA ISS. Further the book maintains articles on actual problems of natural and artificial intelligence, information interaction and corresponded intelligent technologies, expert systems, robotics, classification, business intelligence; etc. In more details, the papers are concerned in:conceptual problems of the natural and artificial intelligent systems: structures and functions of the human memory, ontological models of knowledge representation, knowledge extraction from the natural language texts;network technologies;evolution and perspectives of development of the mechatronics and robotics;visual communication by gestures and movements, psihology of vision and information technologies of computer vision, image processing;object classification using qualitative characteristics;methods for comparing of alternatives and their ranging in the procedures of expert knowledge processing;ecology of programming – a new trend in the software engineering;decision support systems for economics and banking;systems for automated support of disaster risk management;and etc.

It is represented that book articles will be interesting for experts in the field of information technologies as well as for practical users.

General Sponsor: Consortium FOI Bulgaria (www.foibg.com).

Printed in Bulgaria

ISBN 978-954-16-0043-9

C\o Jusautor, Sofia, 2010

ТИРАЖИРОВАНИЕ ДАННЫХ В ДИНАМИЧЕСКИ НАСТАИВАЕМЫХ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

Людмила Лядова, Михаил Стрелков

***Аннотация**: Рассматривается подход к тиражированию данных в распределенных информационных системах, допускающих динамическую реструктуризацию данных, расширение функциональности системы в ходе ее эксплуатации. Описывается модель информационной системы, управляемой метаданными, приводятся описания схемы и алгоритмов тиражирования, обсуждаются некоторые особенности их реализации. Предлагаемые средства позволяют реплицировать данные, размещенные в локальной базе данных системы-источника, в соответствии со схемой, определяемой пользователем, во внешние подсистемы-приёмники в формате, пригодном для тиражирования данных в гетерогенных системах. Восстановление данных в системе-приёмнике происходит в автоматическом и «ручном» режиме.*

***Keywords**: репликирование данных, схема тиражирования, распределенные информационные системы, гетерогенные системы, динамическая адаптация, метаданные, метамоделирование.*

***ACM Classification Keywords**: C. Computer Systems Organization – C.2 Computer-Communication Networks: C.2.4 Distributed Systems – Distributed applications, Distributed databases; H. Information Systems – H.2 Database Management: H.2.4 Systems – Distributed databases; H.2.5 Heterogeneous Databases – Data translation.*

Введение

Распределенная информационная система (ИС) в общем случае предполагает как *распределенное хранение*, так и *распределенную обработку* данных. Данные, используемые при выполнении типовых операций, автоматизируемых с помощью программных средств ИС, должны быть приближены к местам их обработки. Таким образом, при создании крупных распределенных информационных систем, включающих территориально удаленные подсистемы, работающие с собственными копиями данных, размещенными в локальных базах данных (БД) этих подсистем, на локальных серверах, необходимо обеспечить возможность *взаимодействия* ИС, их *интеграции*. При этомприходится решать проблемы, связанные с *фрагментацией* данных, необходимостью их *тиражирования*, передачи из одной подсистемы в другую и *синхронизации* локальных копий. Асинхронный подход к тиражированию данных приемлем для многих ИС, так как зачастую в них не требуется обеспечивать идентичность локальных БД различных подсистем в любое время. Однако при передаче данных из одной подсистемы в другую должна решаться задача их согласования.

Вопросам интеграции распределенных источников данных уделяется большее внимание. Большинство современных промышленных СУБД имеют встроенные средства реплицирования данных несколькими способами [1-4], но при интеграции *разнородных информационных систем* использовать возможности, обеспечиваемые различными СУБД, зачастую не представляется возможным, поэтому задача тиражирования решается и при создании прикладных систем [5]. Кроме того, во многих случаях уже после внедрения ИС возникает необходимость *изменения модели предметной области, реструктуризации данных, расширения функциональности ИС*. Информационные системы, допускающие возможность динамической настройки на меняющиеся условия и потребности пользователей в ходе эксплуатации, обеспечивают максимальную гибкость при установке и оперативную адаптацию к происходящим изменениям.Однако эти возможности ИС должны быть обеспечены соответствующими средствами.

В частности, необходимо предусмотреть возможность динамического изменения модели системы, структуры данных, настройки пользовательского интерфейса на внесенные в модель изменения и т.п. Кроме того, могут модифицироваться шаблоны документов (отчетов), генерируемых системой, перечень операций, выполняемых над данными, размещенными в БД и пр. Динамически адаптируемая ИС должна обеспечивать возможность *оперативного тиражирования всех изменений, выполненных в одной подсистеме, в другие подсистемы*. Наличие настраиваемого на изменения компонента тиражирования не только данных, но и моделей ИС является обязательным для современных информационных систем, отвечающих требованию адаптируемости [6].Если требуется организовать информационный обмен между системами, реализованными на разных платформах, использование средств, предоставляемых в распоряжение разработчиков системами управления БД, становится невозможным. Таким образом, разрабатываемые средства тиражирования должны быть *платформенно-независимы*.

В данной статье описывается подход к решению задачи тиражирования данных в динамически настраиваемых ИС, основанных на метамоделировании. Работа выполнена в рамках созданияCASE-системыMETAS, предназначенной для разработки динамически настраиваемых гетерогенных ИС.

Задача тиражирования данных в CASE-системе METAS

Предметная область системы, построенной на основе технологииMETAS, описывается на *логическомуровне* и представляется в виде объектов различных типов и связей между ними [6]. Модель, созданная разработчиком, представляется с помощью многоуровневых *метаданных*. Метаданные содержат также информацию о пользовательском интерфейсе системы (*презентационный* уровень), включают описания запросов к БД и шаблоны документов для формирования отчетов и пр. Возможно расширение функциональности системы за счет создания новых типов данных, разработки иподключения новых операций над объектами ИС. Кроме того, возможности пользовательского интерфейса могут быть расширены подключением нестандартных элементов управления. Операции и элементы управления также описывается метаданными ИС. Возможность динамической адаптации обеспечивается функционированием системы в режиме интерпретации построенных моделей.

В данной статье рассматриваются средства тиражирования данных в гетерогенных ИС, созданных на основе технологии METAS, их передачи из одной подсистемы в другую с обеспечением синхронизации, согласования данных с учетом изменений, которые могли произойти в модели системы [7].

Отличительной особенностью предлагаемого подхода к тиражированию является отсутствие требования постоянного оперативного (on-line) соединения между узлами. Это может оказаться необходимым в условиях ненадежных каналов связи и позволяет передавать пакет тиражирования любым удобным способом (на носителе, по электронной почте и т.д.). Возможность тиражирования данных ИС в *асинхронном режиме*– одна из основных задач при разработке программных средств.

Разрабатываемые средства должны обеспечить возможность *тиражирования взаимосвязанных данных в соответствии со схемами, создаваемыми пользователями* в зависимости от их потребностей в обмене информацией. Реализация компонентов тиражирования данных об объектах ИСв рамках CASE-технологии позволяет оперировать с метаданными системы напрямую, что обеспечивает возможность *выбора данных для разработкисхемы реплицированияв терминах предметной области* ИС и делает возможным *автоматическое включение в пакет тиражирования данных о связанных между собой объектах*.

При тиражировании данных возможна реализация различных *моделей*: *тиражирование данных* об объектах системы;*тиражирование транзакций* (передача и повторение операций, выполненных в одной подсистеме с момента последней репликации, в другую подсистему). Каждая из моделей обладает

определенными преимуществами. Тиражирование транзакций основывается на ведении*журнала произведенных транзакций* в системе.Пакет тиражирования формируется на основе этого журнала и включает в себя все произведенные в системе изменения с момента последней репликации в конкретную подсистему. Тиражирование транзакций уменьшает сетевой трафик. Минусом технологии является неустойчивость к потерям пакетов репликации. Тиражирование данных об объектах– *тиражирование «снимка» БД*, например, таблиц или горизонтальных и/или вертикальных«вырезок» из них. Метод тиражирования снимка хорош для организации процесса начального наполнения баз данных.К недостаткам технологии можно отнести то, что пакет тиражирования имеет больший размер, чем при тиражировании транзакций. Кроме того, невозможно удаление объектов на другом узле.

При более детальном анализе задачи тиражирования выявляется ряд проблем, которые препятствуют созданию универсального механизма реплицирования в динамически адаптируемых системах:

Объем передаваемых данных. При тиражировании имеет смысл передавать не всю базу данных, а только те данные, которые были «заказаны» и изменились со времени передачи последней их реплики. Кроме того, при тиражировании «в обратном направлении» не следует переносить в базу-приёмник записи, полученные из нее же.

Проблема первичных ключей. В разных подсистемах распределенных ИС в БД вносится информация об объектах (записи, каждой из которых назначается некоторый первичный ключ), которая впоследствии тиражируется в другие БД, где, возможно, уже есть записи с такими же значениями ключей, но соответствующие другим объектам предметной области.

Однозначная идентификация объектов. Чтобы алгоритм тиражирования действовал эффективно, необходимо обеспечить однозначную идентификацию объектов, данные о которых получаются из разных источников. В общем случаерешается задача однозначной идентификации объектов по совокупности значений различных атрибутов.

Неполнота и/или ошибочность данных. При работе с БД важную роль играет человеческий фактор. При занесении информации в БД люди вносят в нее массу ошибок, причем как случайно, так и преднамеренно,поэтому возникают проблемы в процессе идентификации объектов ИС, определения, какие из значений атрибутов ошибочны в случае их несовпадения.

Поддержание целостности данных. Возможна ситуация, когда при восстановлении реплики в системе-приёмнике выполняется попыткавключить данные в таблицы, связанные отношениями с другими таблицами, в которые еще не внесены соответствующие записи.

Различная структура данных в базах, участвующих в процессе тиражирования. В этом случае сложно автоматически установить соответствие между атрибутами одного и того же объекта. Проблему можно решать двумя способами:первый метод состоит в приведении структур данных к одному виду, второй предполагает«ручную» (при участии человека) установку соответствия.

Различные форматы хранения данных. Различные СУБД в гетерогенных распределенных информационных системах могут по-разному хранить данные одного и того же типа.

Удаленные записи. При тиражировании данных необходимо передавать сведения об удаленных записях. Это связано с тем, что в базе-приёмнике нужно удалять не все записи, которые отсутствуют в базе-источнике, а только те, что действительно были удалены именно из нее и их копии не могли быть получены из других источников.

Реализация подсистемы тиражирования данных в рамках технологии METAS требует решения следующих задач: построение моделиИС и компонентов тиражирования;разработка метода тиражирования данных об объектах, алгоритмов создания копий данных и их восстановления,

основанных на построенной модели;разработка метода тиражирования транзакций, обеспечивающего независимость от используемой в системе СУБД;реализацияпрограммных компонентов тиражирования в рамках CASE-технологии METAS. Эти задачи решаются с учетом перечисленных выше проблем, различных подходов к их решению.

Базовая модель системы, построенной на основе CASE-технологии METAS

Системы, созданные с помощью CASE-инструментария METAS, функционируют в режиме интерпретации многоуровневых моделей, описывающих систему с различных точек зрения, на разных уровнях детализации. В *основе системы – графовые модели*, представляющие модель (схему) данных ИС, её пользовательский интерфейс и пр.

Математическая модель данных*логического уровня*– граф $G_l(V_l, E_l)$, где $V_l = (e_1,...,e_p), p \in N$ – множество вершин, представляющих объекты предметной области(сущности); $E_l = (r_1,...,r_q), q \in N$ – множество дуг, представляющих связи между ними ($r_i = (e_j, e_k), i = 1..q;\ e_j, e_k \in V_l$), направление дуги определяется типом связи между сущностями. С каждой вершиной графа $e \in V_l$ свяжем множество, $Attr(e) = \{a_{e,0}, a_{e,1},..., a_{e,tn}\}$, представляющее атрибуты сущности e, каждый атрибут представляется парой <*name, ref*>, где *name* – имя атрибута, а *ref* указывает таблицу-справочник, если значение атрибута выбирается из справочника: $a_{ref} \in V_{ph}$, в противном случае – пустое значение $a_{ref} = \varnothing$.

Физический уровень модели системы (уровень представления данных в реляционной БД) описывается графом $G_{ph}(V_{ph}, E_{ph})$, в котором $V_{ph} = \{t_1, t_2,..., t_{pn}\}, pn \in N$ – множество таблиц в БД; $E_{ph} = \{r_1, r_2,..., r_{pm}\}, pm \in N$ – множество связей между ними ($r_i = (t_j, t_k), i = 1..pm;\ t_j, t_k \in V_{ph}$). Направление дуги от t_j(начальной вершины)к t_k(конечной вершине) определяется типом связи между таблицами. Дуги графа могут быть только однонаправленными. С каждой вершиной графа $t \in V_{ph}$ связывается множество *Fields(t)*, обозначающее поля таблицы *t*: $Fields(t) = \{f_{t,0}, f_{t,1},..., f_{t,n_t}\}$, где каждый элемент представляет собой четверку <*name, type, req, pe*> (*name* – имя поля таблицы; *type* – тип поля; *req* – признак, является ли значение обязательным; *pe* указывает родительскую таблицу ($f_{pe} \in V_{ph}$), если поле содержит внешний ключ (первичной ключ родительской таблицы), иначе *pe* имеет пустое значение ($f_{pe} = \varnothing$). Будем считать, что элемент $f_{t,0}$ представляет ключевое поле таблицы *t*. Таким образом, каждая вершина $t \in V_{ph}$ представляет структуру таблицы в БД.

Экземпляр данных (строка) *o* таблицы *t* включает множество значений полей. Обозначим значение ключевого поля (идентификатора) строки *o* $Id(o) = Val(f_0, o) \in N$.

Зададим местоположение данных (строк таблицы) $O(t, place), t \in V_{ph}, place \in Place$ (*place* задает местоположение объектов – значение из множества $Place = \{db, pk\}$ – база данных ИС или пакет тиражирования соответственно). Будем обозначать $O(t, db) = O_{db}(t)$ и $O(t, pk) = O_{pk}(t)$. Обозначим через $CAV(r, o), r = (t_1, t_2) \in E_{ph}, o \in O(t_2, place)$ значение дочернего поля связи указанного объекта.

Пусть $MT(e) = t, e \in V_l, t \in V_{ph}$ – главная таблица сущности *e*; $S(e), e \in V_l$ – множество справочников сущности *e*.

Отношение «M:M» между сущностями реализуется в системе при помощи вспомогательной таблицы. Ей будет соответствовать функция:

$$Mid(r=(e_1,e_2))=\begin{bmatrix} t_m,(e_2,e_1)\in E_l \\ \varnothing,(e_2,e_1)\notin E_l \end{bmatrix}.$$

Опишем соотношения вершин и дуг графа логической модели и элементов графа физической модели.

Любой вершине (сущности) графа логической модели соответствует множество таблиц на физическом уровне, состоящее из главной таблицы и таблиц-справочников:

$$(\forall e\in V_l)(\exists T_e\subset V_{ph}):T_e=MT(e)\cup S(e).$$

Любой связи «1:M» или «M:1» на логическом уровне соответствует одна связь на физическом.

Связь «M:M» реализуется через промежуточную таблицу, при этом главные таблицы сущностей, состоящих в связи, имеют отношение «1:M» с промежуточной таблицей:

$$(\forall r=(e_1,e_2)\in E_l,e_1,e_2\in V_l)(\exists E_r\subset E_{ph}):$$
$$E_r=\{(t_1,t_2)\in E_{ph}\mid t_1,t_2\in V_{ph},t_1=MT(e_1),t_2=MT(e_2),Mid(r)=\varnothing\}\vee$$
$$\vee\{(t_1,t_m)\in E_{ph},(t_2,t_m)\in E_{ph}\mid t_1,t_2,t_m\in V_{ph},t_1=MT(e_1),t_2=MT(e_2),t_m=Mid(r)\neq\varnothing\}.$$

Все множество вершин на физическом уровне можно представить следующим образом:

$$V_{ph}=\left\{\bigcup_{e\in V_l}MT(e)\cup S(e)\cup\left(\bigcup_{\substack{d\in V_l,\\(e,d)\in E_l}}Mid(e,d)\right)\right\}.$$

Это объединение главных таблиц и справочников всех сущностей, а также всевозможных вспомогательных таблиц, реализующих связь «M:M».

Между атрибутами сущности и полями главной таблицы можно установить взаимно однозначное соответствие.

Понятие схемы тиражирования

При тиражировании создаётся*копия данных*, представляющих информацию о некоторых объектах предметной области ИС, размещенных в локальной БД ее подсистемы. Средства реплицирования позволяют извлекать (восстанавливать) данные из этой копии в базы данных других подсистем ИС.

В динамически настраиваемой системе, допускающей реструктуризацию данных в БД, меняется и структура данных, передаваемых из одной подсистемы в другую. Поэтому необходимо обеспечить пользователя средствами, позволяющими не только определить перечень передаваемой информации, но и структуру пакета тиражирования. При создании копии пользователь должен выбрать экземпляр сущности (*начальный*, или *корневой*, объект), информацию о котором требуется передать на другие узлы, а также сформировать список *дочерних сущностей*, связанных с корневой, информация о которых также должна быть передана в другую БД. Для каждой сущности из этого списка должны быть переданы в пакет только те экземпляры, которые имеют отношение к выбранному объекту. При этом начальная сущность может иметь обязательные связи (ссылки) на *родительские сущности*. Для создания корректной копии при включении сущности в список вместе с ней должны быть добавлены все ее родительские сущности.

Вводится понятие *схемы тиражирования*, которая включает:

список сущностей, выбранных пользователем $V_{sc}\subset V_l$;

список атрибутов каждой выбранной сущности, необходимых для сопоставления экземпляров (записей) при восстановлении данных $\forall e \in V_{sc} : \exists IdentAt(e) \subset Attr(e)$, причем список включает все обязательные атрибуты сущности плюс, возможно, некоторые другие:

$IdentAt\ (e) = req\ (e) \cup nodefa\ (e), nodefa\ (e) \subset Attr\ (e)$;

- для каждого атрибута задается *точность сопоставления* $\forall a \in IdentAt\ (e) : \exists EqAc\ (a) \subset Accuracy$ (здесь множество $Accuracy$ задает всевозможные наборы параметров сравнения значений (полное, с учетом регистра, с учетом пробела и др.); факт того, что значения атрибутов (или полей) совпадают с точностью ea, будем записывать следующим образом: $Val(f, o) \overset{ea}{\cong} Val(f, p),\ o \in O(t, pl1), p \in O(t, pl2), f \in Fields(t), t \in V_{sc}$;
- *правила сопоставления объектов*: требуется, чтобы выполнялось равенство значений всех атрибутов или хотя бы одного; правило задается так: $EqType(e) \in EqTypes = \{all, one\}, e \in V_{sc}$;

корневую сущность $e_{start} \in V_{sc}$ и корневой объект $o_{start} \in O(e_{start})$.

Модель данных, образованная списком сущностей V_{sc}, представляет собой некоторую *подсхему логической модели*. *Графом схемы тиражирования* назовём подграф графа логической модели G_l, полученный путём редукции вершин (сущностей), не включенных в схему тиражирования (из множества вершин V_lудаляются все «лишние» вершины, а из множества дуг – все дуги, инцидентные этим вершинам): $G_{sc}(V_{sc}, E_{sc})$, где $V_{sc} = (e_1, ..., e_n) \subset V_l, n \in N, n \le p$ – множество сущностей, заданных схемой; $E_{sc} = (r_1, ..., r_m) \subset E_l, m \in N, m \le q$ – множество связей между ними; $r_i = (e_j, e_k), i = 1..m;\ e_j, e_k \in V_{sc}$, $E_{sc} = E_l \setminus \{r = (e_j, e_k) \in E_l \mid e_j \in V_l \setminus V_{sc} \vee e_k \in V_l \setminus V_{sc}\}$, т.е. из множества дуг графа логической модели G_l удаляются все дуги, инцидентные сущностям, не включенным в схему тиражирования.

Алгоритм создания копии при тиражировании данных

Операция создания копии по заданной схеме – это операция построения графа $G_{pk}(V_{pk}, E_{pk})$, где $V_{pk} \subset V_{ph}, E_{pk} \subset E_{ph}$.Учитывая то, что схема задает для копирования набор сущностей Vsc, становится возможным определить множество таблиц в этом графе

$$V_{pk} = \left\{ \bigcup_{e \in V_{sc}} MT(e) \cup S(e) \cup \left(\bigcup_{\substack{d \in V_{sc}, \\ (e,d) \in E_{sc}}} Mid(e,d) \right) \right\}.$$

При этом в пакет тиражирования должны попасть объекты, имеющие отношение к начальному, -- множество $O(t, pk) \subset O(t, db)$ для каждой таблицы $t \in V_{pk}$.

Разработанный алгоритм выполняет *рекурсивный обход графа схемы тиражирования в глубину*.Перед запуском алгоритма необходимо указать корневой объект и в качестве входных параметров операции задать корневую сущность e_{start} и идентификатор корневого объекта $Id(o_{start})$. ЕслиN – число вершин в графе, аK – общее число строк в таблицах, тоодна и та же вершина может быть выбрана при обходе в худшем случае K раз,таким образом, рекурсивный вызов может быть выполнен $O(K \cdot N)$. Показано, что алгоритм создания копии (реплики) данных при тиражировании имеет сложность $O(K^3 N + K N^2)$.

Алгоритм восстановления данных при тиражировании

Операция восстановления данных из полученной копииподразумевает обновление данных, содержащихся в локальной БД подсистемы ИС, на основе реплики БД, созданнойв другой подсистеме. Пакет тиражирования содержит множество таблиц $V_{pk} \subset V_{ph}$, заполненных строками $O(t, pk)$.

Необходимо организовать просмотр и анализ всех записей. При анализе очередной строки в пакете нужно искать запись в локальной БД, которая представляет тот же самый объект. В случае успешного поиска требуется обновить значения атрибутов, иначе – добавить объект в множество $O(t, db)$.

Правила сопоставления для каждой сущности задаются схемой тиражирования, по которой была создана резервная копия. Строки считаются *похожими*, если у них совпадают значения идентифицирующих атрибутов $IdentAt(e)$с точностью $EqAc(e)$, где $e \in V_l$. При распознавании строк могут возникать *неопределенности*, связанные с неполнотой или ошибочностью данных. Их разрешение целесообразно возложить на пользователя.Если для объекта в пакете не был найден соответствующий объект в БД или их было найдено несколько, то этот экземпляр будет помещен в *промежуточный* (или *транзитный*) пакет вместе с результатами распознавания. Чтобы сохранить целостность данных, необходимо помещать в транзитный пакет и все дочерние объекты «сомнительного» экземпляра.

Будем считать, что граф $G_{pk}(V_{pk}, E_{pk})$ и граф логической модели $G_l(V_l, E_l)$ на узле-приемнике изоморфны соответствующим графам на узле-источнике (подсхема модели данных, образованная схемой тиражирования, одинакова вэтих подсистемах ИС). Для этих условий сложность одной итерации алгоритмавосстановления при обходе таблиц копии – $O(K \cdot A)$, где K – общее число строк в таблицах, A – число атрибутов. Поскольку в цикле просматриваются все строки таблицы, общая сложность процедуры восстановления равна$O(K^2 \cdot A)$.

Если требуется вручную устанавливать соответствие объектов из промежуточного пакета с объектамиБД подсистемы-приемника, то при добавлении нового экземпляра объекта в БД по решению пользователя запускается специальный алгоритм восстановления данных из транзитного пакета.

Заключение

В работе предложен подход к тиражированию данных в системах, управляемых метаданными.

Современные системы управления базами данных (Oracle, MicrosoftSQLServer и др.) содержат встроенные средства тиражирования данных, однако большинство из них имеют ограничения: репликация возможна только между БД одного формата, структуры таблицы-источника и таблицы-приемника должны быть одинаковы.

Предлагаемый подход снимает эти ограничения.Его можно сравнить с тиражированием моментальных снимков в Microsoft SQL Server, но использование моделей, метаданных различного уровня позволяет пользователю работать в терминах предметной области ИС, а также осуществлять гибкую настройку правил сопоставления различных экземпляров сущностей при их передаче из одной подсистемы ИС в другую.

В качестве инструментального средства разработки компонентов тиражирования CASE-системы METAS выбрана платформа Microsoft .NET, для доступа к данным в БД используется технология ADO.NET, данные в пакете тиражирования передаются в формате XML.

Благодарности

Статья частично финансирована из проекта **ITHEA XXI** Института Информационных теорий и Приложений FOI ITHEA и консорциума FOIBulgaria (www.ithea.org, www.foibg.com)

Библиографический список

[1] Артемов Д.В. Microsoft SQL Server // Корпоративные базы данных'96: Материалы технической конференции / Центр информационных технологий. – Москва, 1996.

[2] Барон Г., Ладыженский Г. Технология тиражирования данных в распределенных системах // Открытые системы. – 1994. №2. С. 17-22.

[3] Сиколенко В.В. Поддержка распределенных систем в СУБД Oracle // Системы управления базами данных. – 1996. – № 4. – С. 27-35.

[4] Armendariz-Inigo J.E., Juarez-Rodrıguez J.R., de Mendıvil J.R.G.,Garitagoitia J.R., Munoz-Escoı F.D., Irun-Briz L. Relaxed Approaches for Correct DB-Replication with SI Replicas // Software and Data Technologies: Proceedings of the Third International Conference, ICSOFT 2008. Porto, Portugal, July22-24, 2008. – Springer-VerlagBerlinHeidelberg, 2009. P. 161-174.

[5] Карауш А.С. Модель тиражирования библиографических баз данных // EL Pub2003: VIII Международная конференция по электронным публикациям / Институт вычислительных технологий СО РАН. Новосибирск, 2003.

[6] Лядова Л.Н. Метамоделирование и многоуровневые метаданные как основа технологии создания адаптируемых информационных систем // InternationalBookSeries “Information Science&Computing”, Number 4. SupplementtotheInternationalJournal “InformationTechnologies&Knowledge” Vol. 2, 2008. P. 125-132.

[7] Стрелков М.А. Компонент тиражирования данных в системах, основанных на метаданных // Инженерный вестник / Научно-технический журнал общественного объединения «Белорусская инженерная академия» – Минск, 2006. – №1(21)/2.

Сведенияобавторах

Лядова Людмила Николаевна *– доцент кафедры информационных технологий в бизнесе Пермского филиала Государственного университета – Высшей школы экономики; 614070, Россия, г. Пермь, ул. Студенческая, 38; e-mail: LNLyadova@mail.ru.*

СтрелковМихаилАлександрович *– студент магистратуры; Пермский государственный университет; 614990, Россия, г. Пермь, ул. Букирева, 15; e-mail: strelkopf@yandex.ru.*