

## КВАЗИПРОИЗВОЛЬНЫЕ ЗАПРОСЫ К БАЗАМ ДАННЫХ И ИНФОРМАЦИОННАЯ МОДЕЛЬ ИХ РЕАЛИЗАЦИИ

---

**Abstract.** *The different types of queries by user-analyst to the thematic database are discussed. A technology of creation system for constructing of quasi-arbitrary user queries is proposed. It is constructed the information model of domain as a set of three components: relational model, frame-based model and business rules model that provides the possibility of establishing a system for constructing of quasi-arbitrary queries.*

**Key words:** *arbitrary user requests, information domain model, frame-based domain model, business rules model.*

**Анотація.** *Розглядаються різні типи запитів користувача-аналітика до тематичної бази даних. Пропонується технологія створення системи побудови квазідовільних запитів користувача. Побудована інформаційна модель предметної області у вигляді сукупності трьох складових: реляційної моделі, фреймової моделі та моделі бізнес-правил, що забезпечує можливість створення системи побудови квазідовільних запитів.*

**Ключові слова:** *довільні запити користувача, інформаційна модель предметної області, фреймова модель предметної області, модель бізнес-правил.*

**Аннотация.** *Рассматриваются различные типы запросов пользователя-аналитика к тематической базе данных. Предлагается технология создания системы построения квазипроизвольных запросов пользователя. Построена информационная модель предметной области в виде совокупности трех составляющих: реляционной модели, фреймовой модели и модели бизнес-правил, обеспечивающая возможность создания системы построения квазипроизвольных запросов.*

**Ключевые слова:** *произвольные запросы пользователя, информационная модель предметной области, фреймовая модель предметной области, модель бизнес-правил.*

### 1. Введение

Одно из направлений повышения технологической эффективности процесса подготовки и принятия решений в СППР заключается в повышении оперативности доступа к информационным ресурсам со стороны пользователя-аналитика (ПА). В частности, типовыми ресурсами являются тематические базы данных (ТБД), содержащие информацию по определенной предметной области.

В зависимости от ожидаемой частоты обращений к тем или иным наборам данных базы данных (БД) выделяют следующие общие типы запросов [1]:

1. «Кнопочные» регламентированные запросы (КЗ), созданные заранее. Их описание и SQL-код сохранены в системе реализации запросов. Функциональность (разнообразии возможных вариантов искомых наборов данных) этих запросов минимальна, а оперативность доступа – максимальна. От ПА требуется только найти нужный запрос в предложенном перечне и дать команду на выполнение.

2. Параметрические регламентированные запросы (ПМЗ) тоже создаются заранее, но дают возможность пользователю специфицировать значения отдельных параметров, что расширяет функциональность при незначительном ухудшении оперативности доступа.

3. Произвольные нерегламентированные запросы (ПЗ). Эти запросы характеризуются максимальной функциональностью, поскольку открывают доступ ПА к любым наборам данных, но минимальной оперативностью, так как требуют применения специальных программных средств, а, зачастую, также и участия программистов.

Существует ряд стандартных программ различных фирм для построения произвольных запросов к базам данных (Query Builder фирмы Borland, Microsoft Query фирмы Microsoft и другие). Эти программы предоставляют графический интерфейс для создания произвольных запросов на

языке SQL и предназначены для работы программистов, хорошо разбирающихся в реляционных базах данных.

Средства создания нерегламентированных запросов, предоставляемые готовыми информационно-аналитическими системами (например, Oracle Discoverer) [2], сложны, нуждаются в дополнительном администрировании и специальной подготовке ПА.

В этой связи представляет практический интерес технология реализации запросов, занимающих промежуточное положение между ПМЗ и ПЗ в смысле функциональности, с одной стороны, и оперативности доступа к данным, с другой. Такая технология должна позволять формулировать и реализовывать on-line-запросы ПА, знакомого с предметной областью, без привлечения специально подготовленных лиц. Назовем такие запросы квазипроизвольными.

## 2. Технология создания КПЗ

Предлагаемое технологическое решение направлено на реализацию значительной части произвольных запросов, которые могут быть сформулированы ПА в терминах предметной области. Фактически речь идет о расширенном типе параметрических запросов, в которых пользователь специфицирует не только значение заданных параметров и условий, но и сами параметры.

Основой такой технологии является описание семантики ТБД на естественном языке (ЕЯ) пользователя, т.е. представление сущностей и атрибутов ТБД в терминах области ее применения, где сущность описывает семантический смысл таблицы ТБД, а атрибуты – это составляющие сущности, ее параметры (такое описание может быть сделано, например, с помощью case-средства PowerDesigner 9 фирмы Sybase). Для каждого атрибута существует его характеристика (текст, число, дата). Характеристика определяет метод получения данных по каждому атрибуту БД (правило).

Проанализировав любой пользовательский запрос, можно сделать вывод, что пользователь выбирает из базы данных одну или несколько сущностей с их атрибутами и накладывает определенные правила их нахождения, например, задавая диапазоны значений, временные характеристики и т.д.

Имея описание семантики сущностей и атрибутов всей базы данных, а также правил их нахождения, можно построить систему создания запросов, с помощью которой пользователь мог бы самостоятельно создавать и обрабатывать нужные ему запросы.

Предположим, что  $S_i$  – сущность,  $S_i.A_{ij}$  – атрибуты сущности  $S_i$ ,  $R(A_{ij})$  – правило нахождения атрибута  $A_{ij}$ .

Тогда пример БД может быть представлен следующим образом:

$$S_i.A_{ij}, R(A_{ij}) \leftarrow \dots \rightarrow S_o.A_{oi}, R(A_{oi}) \dots \rightarrow S_i.A_{ij}, R(A_{ij}) \dots \rightarrow S_k.A_{ijk}, R(A_{ijk})$$

$$|$$

$$S_i.A_{ij}, R(A_{ij})$$

Правило нахождения  $R$  для атрибута  $A_{ij}$  типа «число» может состоять из набора выражений

$$R(A_{ij}) = \{>, <, =, <>, <=, >=, \text{содержит}, \text{не содержит}, \%, f(A_{ij})\}.$$

Для атрибута  $A_i$  типа «Дата» получаем

$$R(A_{ij}) = \{>, <, =, <>, <=, >=, \text{диапазон}, f(A_{ij})\}.$$

Для атрибута  $A_{ij}$  типа «Текст»

$R(A_{ij}) = \{=, <>, \text{усечение слева}, \text{усечение справа}, \text{усечение слева и справа}, f(A_{ij})\}$ ,

где  $f(A_{ij})$  – функция, которую можно применить к определенному атрибуту, например,  $\text{sum}$ ,  $\text{max}$ ,  $\text{min}$ ,  $\text{avg}$  и т.д.

Исходя из вышеизложенного, можно привести следующий пример SQL-запроса:

`Select So.Aoi, Si.Aij, Sum(Si.Aik) from So, Si where So.Aoi = K and Si.Aij=T and (So.Aoj Between T1 and T2) group by So.Aoi, Si.Aij,`

где  $K$ ,  $T$ ,  $T1$ ,  $T2$  – константы, удовлетворяющие правилам  $R$ .

Таким образом, предлагаемая технология создания КПЗ на ЕЯ заключается в:

- 1) представлении сущностей и их атрибутов ТБД на ЕЯ в терминах предметной области с помощью case-средства (например, PowerDesigner 9);
- 2) определении правил нахождения различных атрибутов и функций, которые можно применить к ним, и в описании их на ЕЯ с помощью case-средства;
- 3) разработке интерфейса, в котором ПА предлагаются на выбор знакомые ему термины предметной области и правила их нахождения.

Основываясь на описанной технологии и разработав соответствующий интерфейс, можно создать систему построения квазипроизвольных запросов (СПКПЗ) ПА к ТБД.

Общий алгоритм функционирования СПКПЗ приведен на рис. 1.

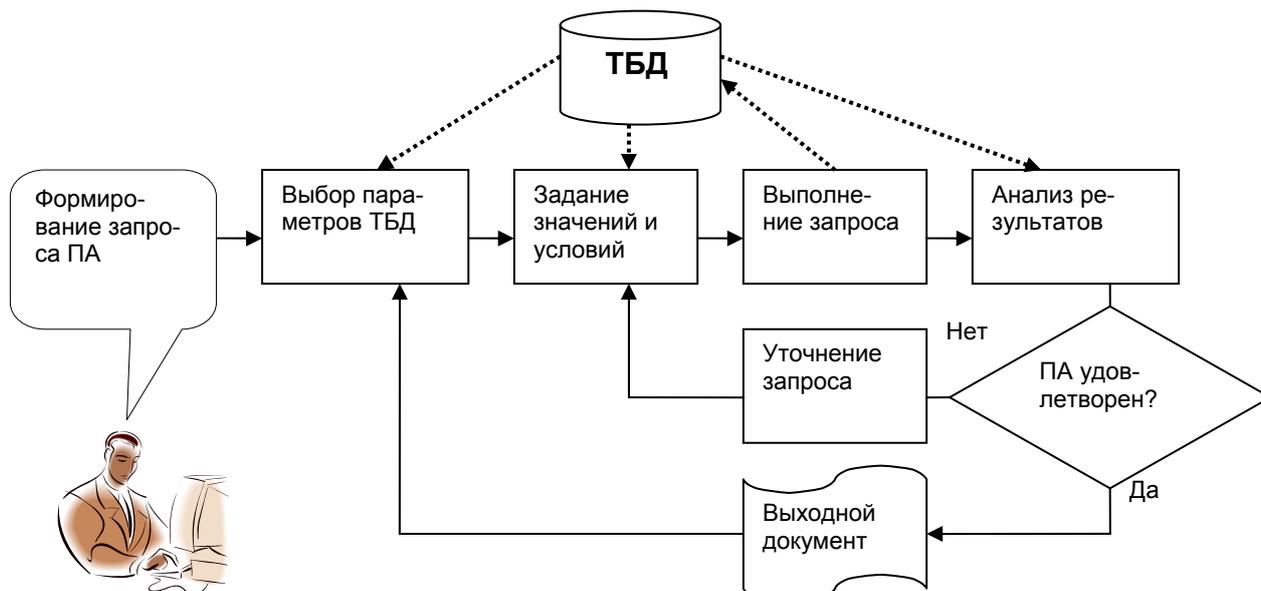


Рис. 1. Общий алгоритм СПКПЗ

### 3. Построение информационной модели КПЗ

В основе информационной модели СПКПЗ лежит информационная модель предметной области (ПрО).

Под ПрО будем понимать совокупность объектов некоторой ограниченной части внешнего мира, о которых ПА может задавать вопросы. ПА не может получить в ИС ответ на вопрос о том, что ей неизвестно.

Информационную модель ИнфМод ПрО, описывающую семантику предметной области, представим в виде совокупности 3-х составляющих (рис. 2):

- реляционной модели РелМод данных ПрО;
- фреймовой модели ФрМод ПрО;
- модели бизнес-правил БПрМод ПрО.

ИнфМод={РелМод, ФрМод, БпрМод}.

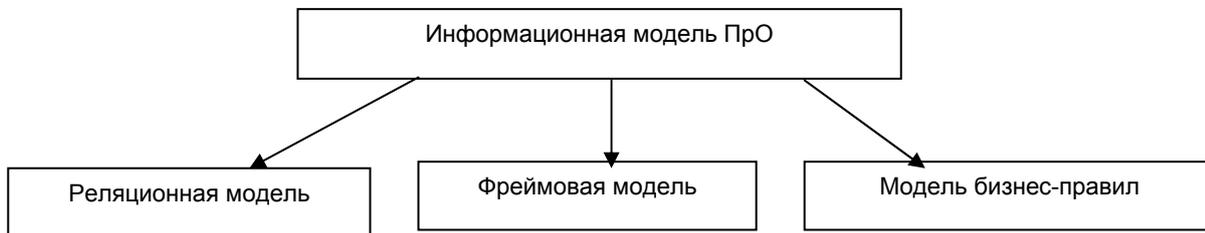


Рис. 2. Декомпозиционная схема информационной модели

РелМод данных ПрО включает в себя сущности РелСущ, их атрибуты РелАтр и отношения между сущностями Отн(сущн-сущн).

РелМод={РелСущ, РелАтр, Отн(сущн-сущн)}.

РелМод отражает структуру хранения данных в БД. Она, как правило, состоит из множества таблиц и отношений между ними. Эта модель сложна и не предназначена для работы ПА. Для того, чтобы ПА мог самостоятельно формировать запросы, нужна более простая и знакомая ПА структура понятий ПрО, в качестве которой может быть предложена фреймовая модель ПрО.

#### 4. Фреймовая модель предметной области

Фреймовая модель представляет ПрО в виде жесткой иерархической структуры фреймов, в качестве которых выступают тематические разделы ПрО:

ФрМод={ФР, СЛ, ОТН(фр-фр)},

где ФР – фреймы (разделы ПрО, состоящие из одного или нескольких понятий ПрО);

СЛ – слоты (термины разделов ПрО, которые могут принимать определенные значения);

ОТН(фр-фр) – связь между фреймами.

Фреймы и слоты в данной модели располагаются в строгом порядке в соответствии с их показателями важности (приоритета)  $w_i$ , определенными по шкале  $1,2,3,\dots,n$ . В начале стоит корневой фрейм  $\Phi P_1$  с  $w_1 = 1$ , затем следуют остальные фреймы в порядке возрастания их  $w_i$  ( $w_i > w_{i-1}$ ). Аналогичным образом располагаются слоты внутри каждого фрейма, так что каждый слот имеет свое место внутри своего фрейма. Полученная структура образует дерево важности разделов (ДВР) ПрО и их терминов с главным разделом в корне.

Открывая окно интерфейса СПКПЗ, ПА видит на экране ДВР, из которого он может выбрать любую комбинацию разделов и терминов ПрО, задать условия (значения определенных заданных им терминов) и получить результат (значения искомых терминов).

Задача создания ФР с точки зрения базы данных может быть решена с помощью виртуальных таблиц – тематических представлений (ТемаView). Каждое ТемаView представляет один из

фреймов ФрМод, а его атрибуты – это слоты ФрМод. Содержимое TemaView динамически вычисляется на основании данных, находящихся в таблицах РелМод:

TemaView создаются по определенным правилам, включающим:

- правило наименований (кодировки);
- правило нумерации;
- правило соединения.

Правило наименований позволяет идентифицировать TemaView на уровне объектов БД, на которых основывается программное приложение.

Правило нумерации TemaView дает возможность построить ДВР в интерфейсе пользователя.

Правило соединения TemaView используется для определения объема информации, получаемой ПА по запросу. Поскольку соединение TemaView производится автоматически (программно), то TemaView должны иметь общий ключ (состоящий из одного или нескольких атрибутов). Таким ключом (ID) может служить первичный ключ таблицы фактов РелМод.

Рассмотрим две схемы соединения TemaView: последовательную и иерархическую.

Последовательная схема соединения TemaView (TV) (рис. 3) – это цепь, которая предполагает такое соединение TemaView по ключу ID, где каждое последующее TemaView соединяется с предыдущим.

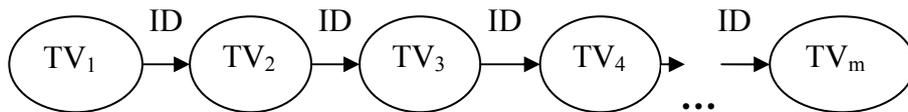


Рис. 3. Последовательная схема соединения

В результате соединения всех TemaView получается отношение  $Q$ , представленное в дизъюнктивной нормальной форме:

$$Q = TV_1 \times TV_2 \cup TV_2 \times TV_3 \cup TV_3 \times TV_4 \cup \dots \cup TV_{m-1} \times TV_m = \bigcup_2^m TV_{m-1} \times TV_m, \quad (1)$$

где конъюнкция  $TV_{m-1} \times TV_m$  – это отношение, содержащее кортежи из декартова произведения отношений  $TV_{m-1}$  и  $TV_m$ , удовлетворяющие условию равенства по общему ключу ID.

Дизъюнкция получается в результате применения операции левого внешнего соединения (left outer join), включающего в результирующее отношение  $Q$  кортежи с незадаваемыми атрибутами правых частей конъюнкций  $TV_{m-1} \times TV_m$  (с NULL-значениями  $TV_m$ ).

Выражению (1) соответствует следующий оператор Select (по стандарту ANSI SQL 92):

Select TV1.A1, TV2.A2, TV3.A3, ..., TVm.Am.

From TV1 left outer join TV2 on TV1.ID = TV2.ID left outer join TV3 on TV2.ID = TV3.ID left outer join TV4 on TV3.ID = TV4.ID ... left outer join TVm on TVm-1.ID = TVm.ID.

Результат соединения  $Q_1$  представляет собой множество кортежей декартова произведения атрибутов TemaView, описывающих ФР и их термины, выбранные ПА.

Недостатком этой схемы является тот факт, что отсутствие данных в каком-либо из TemaView приводит к разрыву цепи и недоступности оставшейся правой части, то есть данные из всех TemaView, стоящих правее, не будут выбраны.

Поскольку ПА должен иметь возможность выбора любых (всех) кортежей из ТБД, целесообразно использовать иерархическую схему соединения TemaView, в которой все TemaView соединены с одним главным TemaView по ключу ID. Эта схема представляет собой ориентированный граф  $G$  – веер, состоящий из главной корневой вершины степени  $m-1$  и  $m-1$  висячих вершин (листьев) степени 1 (где  $m$  – количество TemaView) (рис. 4).

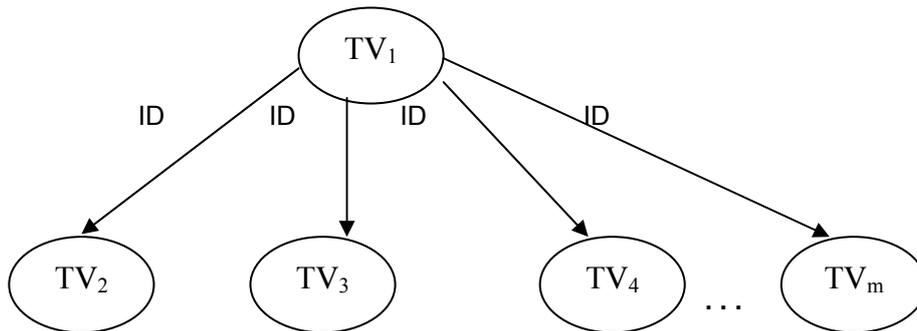


Рис. 4. Иерархическая схема соединения

$G$  – это двудольный граф, моделирующий отношения между главным TemaView –  $TV_1$  (корневым фреймом, общим разделом ПрО) и остальными TemaView (фреймами, разделами). Если ТБД содержит данные, то  $TV_1$  также будет непустым. Поскольку вершинное покрытие (минимальное число вершин, соединенных со всеми другими вершинами) такого графа равно 1, то можно утверждать, что он представляет наилучший вариант представления структуры ПрО для СПКПЗ, так как дает возможность относительно легко получать данные из всех TemaView, соединенных с общей вершиной  $TV_1$ . Ребрами графа  $G$  является общий ключ ID, с помощью которого соединяются все TemaView.

Добавление нового ФР или удаление существующего ФР в интерфейсе пользователя – это добавление новой вершины в граф  $G$  или удаление существующей. Эта операция выполняется администратором путем создания или удаления соответствующего TemaView и не влияет на работу ПА.

Результат соединения – отношение  $Q$  в дизъюнктивной нормальной форме имеет вид

$$Q = TV_1 \times TV_2 \cup TV_1 \times TV_3 \cup TV_1 \times TV_4 \cup \dots \cup TV_1 \times TV_m = \bigcup_2^m TV_1 \times TV_m . \quad (2)$$

Дизъюнкция получается в результате применения операции левого внешнего соединения (left outer join), включающего в результирующее отношение  $Q$  кортежи с незадавленными атрибутами правых частей конъюнкций  $TV_1 \times TV_m$  (NULL-значения  $TV_m$ ).

Выражению (2) соответствует следующий оператор Select (по стандарту ANSI SQL 92):

Select TV1.A1, TV2.A2, TV3.A3, ..., TVm.Am.

From TV1 left outer join TV2 on TV1.ID=TV2.ID left outer join TV3 on TV1.ID=TV3.ID ... left outer join TVm on TV1.ID=TVm.ID.

Применение операции левого внешнего соединения left outer join позволяет создать конструкцию логического «ИЛИ», при этом отсутствие данных в любом из TV не приводит к недоступности оставшейся части, как в последовательной схеме, и не ведет к искажению результирующего отношения (подразумевается, что в непустой БД главное TemaView – TV1 всегда содержит информацию). Иерархическая схема соединения TemaView позволяет получить результирующее отношение  $Q$  в виде множества кортежей декартова произведения отношений  $TV_1$  и  $TV_m$ , удовлетворяющих условию равенства по общему ключу ID без потери информации и без избыточности. В этом заключается преимущество иерархической схемы по сравнению с последовательной.

## 5. Модель бизнес-правил КПЗ

Отношение  $Q$ , полученное в результате соединения TemaView (2), представляет собой исходное отношение, с которым будет работать ПА, выбирая атрибуты (слоты отдельных фреймов) и накладывая ограничения или применяя предопределенные функции, заданные моделью бизнес-правил [4].

Модель бизнес-правил БПрМод описывает правила, которым подчиняются значения слотов (терминов) ПрО, соединяя каждый слот ФрМод с определенным набором правил. Она включает

$$\text{БПрМод} = \{\text{БПр}, \text{СЛ}\},$$

где БПр = {>, <, =, in, not in, like, %, функция, ...} – множество бизнес-правил;

СЛ – множество слотов (терминов разделов ПрО).

Открывая окно интерфейса, где отображена фреймовая модель ПрО, ПА выполняет следующие операции:

1. Операция выбора слотов. Этой операции соответствует реляционная операция проекции  $\pi$  [3]:

$$Q_\pi = \pi_{A_1, A_2, \dots, A_m}(Q),$$

где  $Q_\pi$  – отношение, полученное в результате применения операции проекции к исходному отношению  $Q$ ,  $A_m$  – атрибуты отношения  $Q$  (слоты ФР).

2. Операция наложения условий. Этой операции соответствует реляционная операция селекции  $\sigma$ :

$$Q_\sigma = \sigma F(Q_\pi),$$

где  $Q_\sigma$  – отношение, полученное в результате применения операции селекции к отношению  $Q_\pi$ ;

$F$  – критерий выбора на множестве атрибутов в форме логического выражения, образованного с помощью определенных операндов (констант, имен атрибутов, арифметических операций сравнения, логических операций, аналитических функций), которые представляют собой множество бизнес-правил БПр, разработанных и связанных с каждым слотом таким образом, что каждый слот

имеет свой набор бизнес-правил. В результате применения операций проекции и селекции результирующее отношение  $Q_{\sigma}$  будет содержать множество кортежей, удовлетворяющих критерию  $F$ , что дает возможность полностью автоматизировать работу ПА в интерфейсе СПКПЗ.

## 6. Выводы

Предложенная технология создания СПКПЗ может быть использована для построения интеллектуальных пользовательских интерфейсов в СППР, позволяющих расширить возможности ПА по реализации произвольных запросов без необходимости знания структуры ТБД и языка SQL или привлечения администратора.

Из изложенного выше следует, что КПЗ обладают большей функциональностью по сравнению с ПМЗ за счет того, что:

- они позволяют ПА самому создавать запросы к ТБД (а не предлагают перечень заранее созданных запросов);

- имеют большую оперативность по сравнению с ПЗ за счет того, что ПА может работать в рамках фреймовой модели ПрО, где фреймы и слоты (тематические разделы и их термины) систематизированы и расположены в привычном для ПА порядке и для каждого слота имеется свой набор бизнес-правил. То есть часть работы уже выполнена.

Однако нужно отметить, что КПЗ обладают тем недостатком, что ПА не может применять правила выбора слотов и задавать функции, которые не предусмотрены заранее в программе. В этом состоят ограничения КПЗ по сравнению с ПЗ.

## СПИСОК ЛИТЕРАТУРЫ

1. Оксанич И.Н. К вопросу реализации произвольных запросов пользователя к тематической базе данных / И.Н. Оксанич // Системи підтримки прийняття рішень. Теорія і практика: зб. доповідей наук.-практ. конф. з міжнар. участю (Київ, 8 червня 2009 р.). – Київ, 2009. – С. 78 – 81.
2. Печенкин И.А. Информационные технологии на службе разведки. Обзор современных программных средств обеспечения принятия управленческих решений [Электронный ресурс] / И.А. Печенкин. – Режим доступа: <http://www.it2b.ru/files/pechenkin.pdf>.
3. Райордан Р. Основы реляционных баз данных / Р. Райордан. – М.: Издательско-торговый дом «Русская редакция», 2001. – С. 95 – 101.
4. Muehlen M. Business Process and Business Rule Modeling: A Representational Analysis [Электронный ресурс] / M. Muehlen, M. Indulska, G. Kamp. – Режим доступа: <http://www.bpm-research.com/download/papers/MIZU.MAIN.GEKA-VORTE2007-Final.pdf>.

*Стаття надійшла до редакції 13.10.2009*