УДК 004.415:004.89

Сагайда П. И., Зори А. А.

РАЗРАБОТКА МОДЕЛИ И МЕТОДА ИНТЕРПРЕТАЦИИ ОНТОЛОГИЙ И ЗАПРОСОВ К БАЗАМ ЗНАНИЙ С ИСПОЛЬЗОВАНИЕМ РЕЛЯЦИОННОЙ МОДЕЛИ ХРАНЕНИЯ ДАННЫХ

В [1, 2] впервые предложена и детально разработана методология проектирования хранилищ данных и знаний (ХДиЗ) для решения задач обработки и анализа данных на основе категориально-онтологических моделей, объединяющая, в отличие от существующих методик, проектирование с помощью информационных и даталогических моделей с использованием различных диаграммных методик и языков моделирования, что позволило устранить недостатки и дополнить достоинства различных подходов к проектированию и получить рациональную структуру ХДиЗ. Однако программные компоненты, разработанные в настоящее время для реализации задач обработки формализованных знаний в виде онтологических моделей, обрабатываемых в соответствующих форматах хранения, требуют использования мощных вычислительных систем [3] и обладают низкой оперативностью выполнения запросов к базам знаний.

Целью данной работы является разработка модели и методики интерпретации онтологий и запросов к базам знаний, отличие которых состоит в использовании, на основе такой интерпретации, реляционной модели хранения данных и формулировки запросов, что позволило реализовать компоненты для интеллектуальной обработки данных (ИОД) с высокой производительностью, в том числе на базе систем с ограниченными вычислительными возможностями.

Рассмотрим переход от модели Unified Modeling Language (UML) к онтологической модели, представленной средствами Ontology Web Language (OWL) с аксиомами в виде высказываний Descriptive Logic (DL), с представлением ограничений на участие экземпляров классов в свойствах (Property), выводимых на основе кардинальности отношений между классами. Для обоснования возможности и корректности такого перехода была разработана и исследована мета-модель представления знаний о предметной области (ПрО) функционирования компьютерных систем (КС) для ИОД средствами OWL DL, в виде категориально-онтологической модели [4] результатов инженерии знаний на OWL DL. Данная категориально-онтологическая (КО) модель представлена на рис. 1. В КО модели ее объекты и морфизмы отображают основные особенности представления знаний о ПрО в виде онтологической модели на языке OWL DL – использование для представления свойств (отношений) ПрО объектов «Class» и «Object Property» для случая, когда свойством ПрО является экземпляр сущности, и «Data Type» и «Data Property», для случая, когда в качестве свойства рассматривается атрибут сущности, имеющий для ее экземпляра конкретное значение («Literal»). Возможность отображения морфизмов КО модели ПрО на «Property» модели на языке OWL DL представлена различными видами «Object Property», соответствующими видам морфизмов в КО модели для диаграммы теории категорий.

Объект equalizer («EQ») ТК, введенный в КО модель на рис. 1, обосновывает следующие важные факты модели на языке OWL DL: для каждого индивидуала (экземпляра класса) в такой модели должен быть введен экземпляр «Object Property» с соответствующим значением кодомена (Range); такой индивидуал может участвовать, через ограничение принадлежности к классу, в экземпляре «Data Property», с учетом ограничений на назначенный данному свойству в качестве кодомена тип данных. Однако все факты участия индивидуалов в экземплярах «Object Property» должны подчиняться экземплярам «Participation Constraint», в общем случае выражаемых с помощью DL для классов и свойств модели OWL [3].

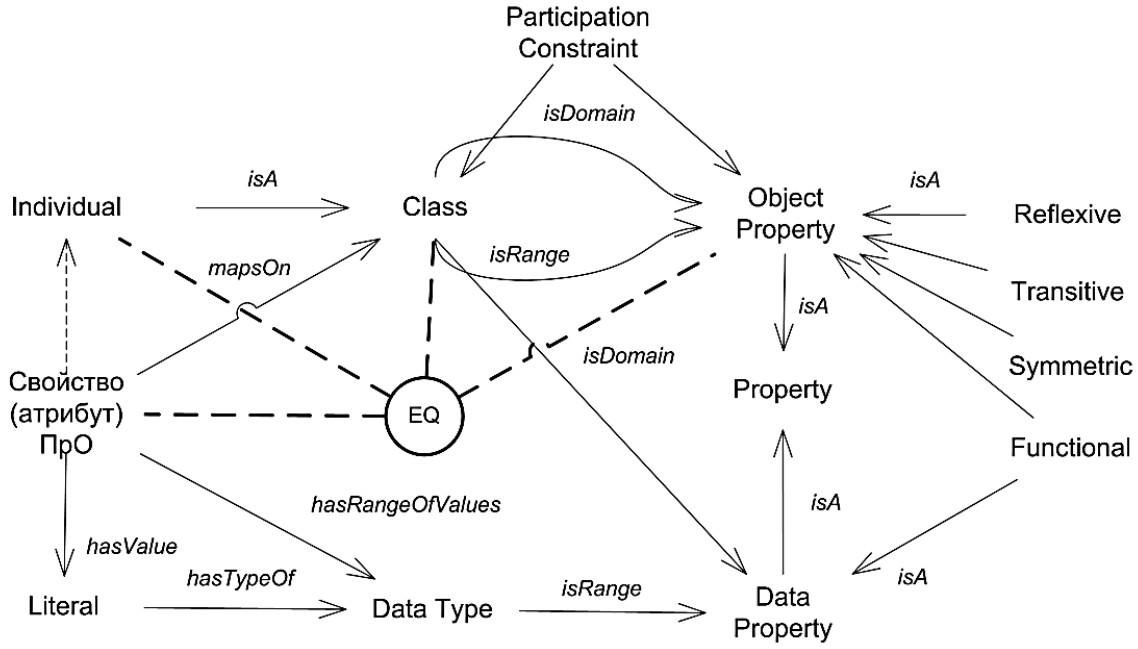

Рис. 1. Категориально-онтологическая модель результатов инженерии знаний средствами языка OWL DL

Процесс отображения концептуальной модели и содержимого хранилища знаний о ПрО в формате OWL DL, дополненного правилами на языке Semantic Web Rule Language (SWRL) [1, 3] в реляционное представление базы знаний (с использованием реляционной модели хранения данных) в виде КО модели, представлен на рис. 2.

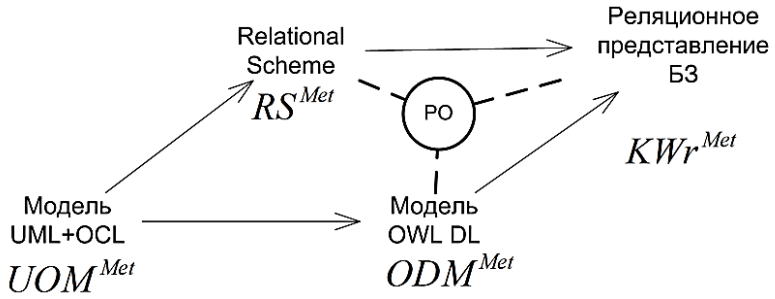

Рис. 2. Категориально-онтологическая модель отображения концептуальной модели и содержимого хранилища знаний о ПрО в формате OWL DL + SWRL в реляционное представление базы знаний

Объект pushout ТК, примененный в КО модели на рис. 2, формализует вычисление реляционного представления базы знаний (БЗ) KWr^{Met} из модели ODM^{Met} в формате OWL DL и реляционной схемы RS^{Met} хранилища данных ПрО с учетом архитектуры и ограничений модели UOM^{Met} на UML + OCL:

$$KW^{Met} = ODM^{Met} \amalg_{UOM^{Met}} RS^{Met}. \quad (1)$$

Рассмотрим объекты ODM^{Met} и KWr^{Met} как категории $\mathbf{ODM}^{Met}$ и $\mathbf{KWr}^{Met}$ с внутренней структурой в виде объектов и морфизмов, а морфизм

$\mu_{ODM,KWr} : ODM^{Met} \to KWr^{Met}$, приведенный на рис. 2 как функтор $Func_{KWr}^{ODM} : \mathbf{ODM}^{Met} \to \mathbf{KWr}^{Met}$, отображающий объекты этих категорий и их морфизмы друг в друга. Тогда КО модель для такого отображения является формальным представлением методики отображения, позволяющей построить реляционное представление базы знаний на основе его представления в формате OWL DL. На рис. 3 приведена КО модель для такого отображения. При построении данной КО модели использованы результаты моделирования, представленные в работах [1, 2, 4].

Модель, представленная на рис. 3, содержит объекты и морфизмы для категории, построенной для баз знаний в виде онтологических моделей с использованием выражений и аксиом в формате OWL DL, а также объекты и морфизмы для категории, построенной для баз данных, использующих реляционную модель представления. На рис. 3 в виде прерывистых линий показаны отображения только объектов двух категорий в ходе реализации функтора $Func_{KWr}^{ODM}$, хотя, в соответствии с теории категорий, при его реализации происходит и отображение морфизмов двух категорий. Этот аспект в данном представлении, для наглядности, опущен.

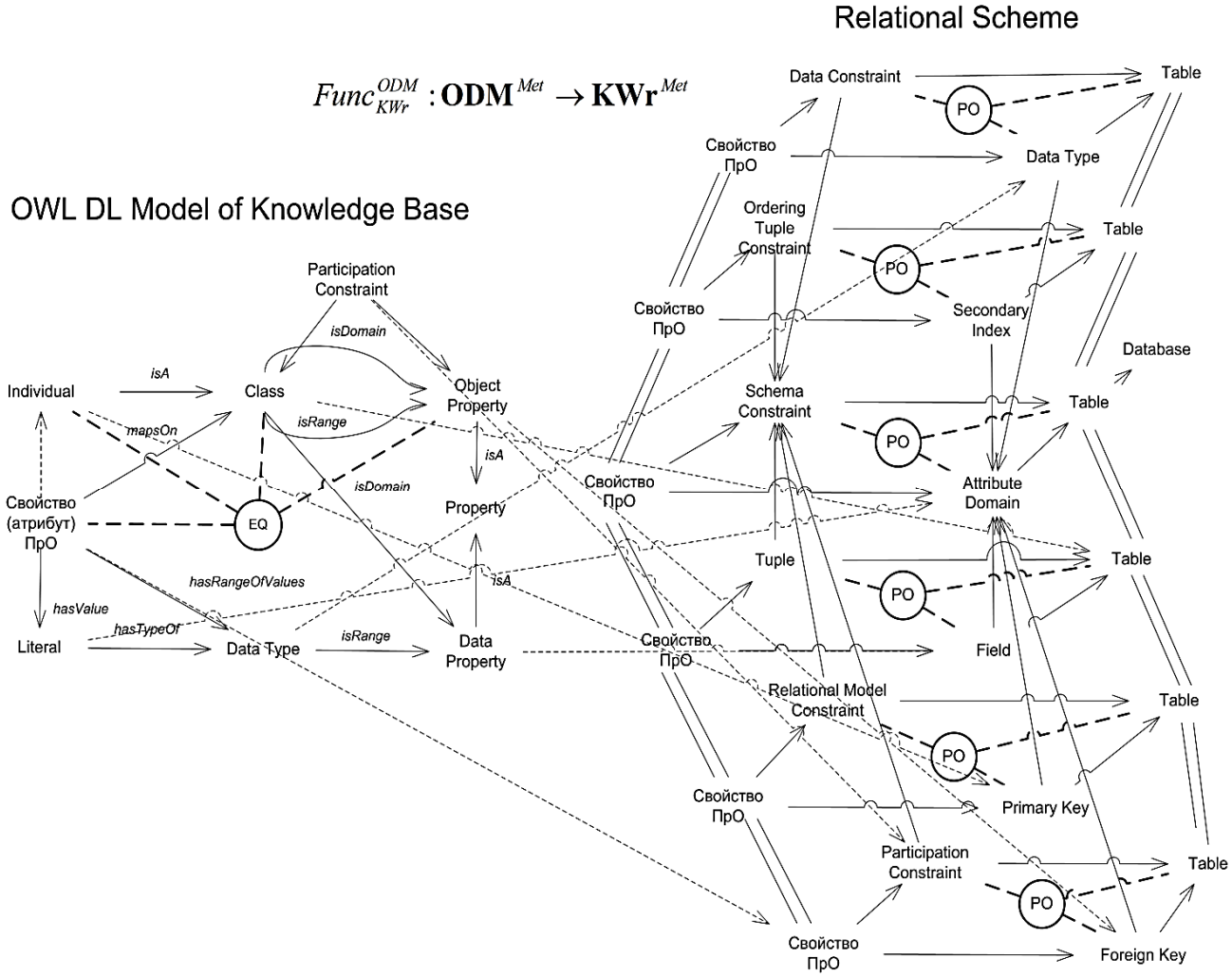

Рис. 3. Категориально-онтологическая модель отображения категорий $\mathbf{ODM}^{Met}$ и $\mathbf{KWr}^{Met}$ с внутренней структурой в виде их объектов и морфизмов

На рис. 4 приведена упрощенная версия результатов, представленных на рис. 3. В КО модели на рис. 4 рассмотрено только отображение объектов соответствующих категорий в процессе реализации функтора $Func_{KWr}^{ODM} : \mathbf{ODM}^{Met} \rightarrow \mathbf{KWr}^{Met}$, учета которых достаточно для алгоритмизации разрабатываемой методики.

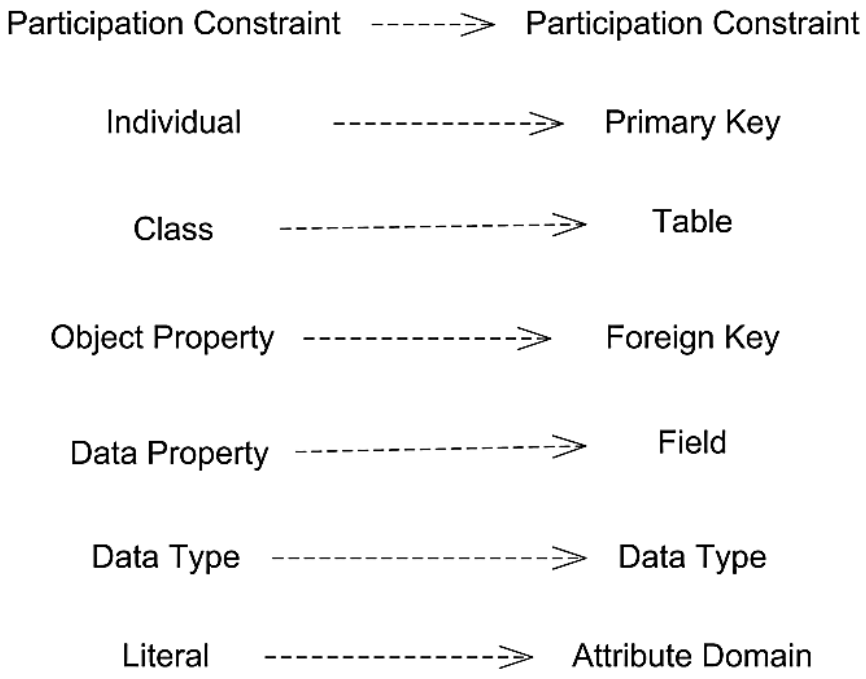

Рис. 4. Схема отображения объектов рассмотренных категорий в процессе реализации функтора $Func_{KWr}^{ODM} : \mathbf{ODM}^{Met} \rightarrow \mathbf{KWr}^{Met}$

В ходе теоретических и практических исследований получены результаты сравнительного анализа выразительных возможностей языков Object Constraint Language (OCL) [5], SWRL, Stored Procedures and Triggers Language (SPL) [1, 2] и информационных и даталогических моделей, на которых базируется синтаксис и семантика данных языков. Разработаны запросы к ХДиЗ на языках OCL, SQWRL [3], SQL, показаны возможности использования реляционной схемы для хранения онтологических моделей и использования языка SQL для запросов к таким моделям, что позволяет реализовать базы знаний без использования специализированных средств для редактирования онтологий и запросов к ним, в том числе в компьютерных системах с ограниченными вычислительными возможностями (в частности, встроенных системах).

По результатам анализа полученных результатов, и на основе теоретического обоснования применения в составе ХДиЗ ограничений OCL и правил SWRL/SQWRL, а также с учетом преемственности КО моделей, моделей UML, Entity-Relationship and Functional Dependencies (ER+FDs) [1, 6] и OWL DL в рамках разработанной методологии [1, 2], представим отображение правил SWRL/SQWRL в SQL/SPL в процессе проектирования ХДиЗ в виде следующего алгоритма, приведенного на рис. 5. Отличие разработанной модели и методики отображения хранилища знаний в формате OWL DL + SWRL в реляционное представление базы знаний, состоит в использовании в ходе такой интерпретации категориально-онтологических моделей хранилищ знаний и данных в соответствующих форматах хранения.

Вход: // Схема реляционной БД, содержащая набор схем отношений
DatabaseScheme $= \langle RelationSchemes, Constraints \rangle$;
$Constraints =$ collectionof $Constraint :: Trigger \mid StoredProc \mid View$**;**
$StoredProc \mid View :: < project\_part >, < join\_part >, < select\_part >,$
$< groupBy\_part >, < nestedStoredProc\_part >;$

// Результат формализации БЗ **KnowledgeBase** в виде аксиом **OWL_DL** и правил (запросов) **SWRL**
KnowledgeBase $= \langle$ **OWL_DL, SWRL** $\rangle$; **OWL_DL** $= \langle Classes, Properties, Axioms \rangle$;
$Axioms =$ collectionof $Axiom :: ClassAssertion \mid PropertyAssertion \mid Restriction$;
SWRL $= \langle Rules \rangle$; $Rules =$ collection of $Rule :: Antecedent = \langle Atoms \rangle, Consequent = \langle Atoms \rangle$;
Выход: // Схема реляционной БД **DatabaseScheme**, содержащая набор схем отношений и набор ограничений, сгенерированных на основе аксиом и правил **KnowledgeBase**
Begin
$Constraints = \{\}$;
//Отображение **OWL_DL** аксиом БЗ на триггеры и хранимые процедуры БД для реализации правил ПрО
for all $A \in Axioms$ ***in*** **OWL_DL** ***do begin***
if isClassAssertion(A) and not exists($A \cong C \in Constraints$ ***in*** **DatabaseScheme**) ***then begin***
$R =$ createRelation(A) as Relation; $RelationSchemes = RelationSchemes \cup R$; ***end;***
if isPropertyAssertion(A) and not exists($A \cong C \in Constraints$ ***in*** **DatabaseScheme**) ***then begin*** $C =$ createTrigger(A) as Constraint; $Constraints = Constraints \cup C$; ***end;***
if isRestriction(A) and not exists($A \cong C \in Constraints$ ***in*** **DatabaseScheme**) ***then begin***
$C =$ createTrigger(A) as Constraint; $Constraints = Constraints \cup C$; ***end;***
end for;
//Отображение **SWRL** правил БЗ на триггеры и хранимые процедуры БД для реализации правил ПрО
for all $Rule \in Rules$ ***in*** **SWRL** ***do begin***
$C =$ createStoredProc($Rule$) as StoredProc;
for all $Atom \in Rule.Antecedent$ ***in*** $Rule$ ***do begin***
if isClassExpression($Atom$) ***then***
C.project_part = C.project_part $\cup$ createProjection($Atom$, $RelationSchemes$);
if isIndividualPropertyExpression($Atom$) ***then***
C.join_part = C.join_part $\cup$ createJoin($Atom$, $RelationSchemes$);
if isIndividualRangeRestriction($Atom$) ***then***
C.groupBy_part = C. groupBy_part $\cup$
createGroupByRestriction ($Atom$, $RelationSchemes$);
if isDataValuedPropertyExpression($Atom$) ***then***
C.project_part = C.project_part $\cup$ createProjection($Atom$, $RelationSchemes$);
if isDataRangeRestriction($Atom$) ***then***
C.select_part = C.project_part $\cup$ createSelection($Atom$, $RelationSchemes$);
end for;
for all $Atom \in RuleConsequen$ ***in*** $Rule$ ***do begin***
if isSQWRLExpression($Atom$) ***then*** **continue;**
if isClassExpression($Atom$) ***then begin***
$R_C =$ createRelation($Atom$) as Relation;
if not exists(R_C ***in*** $RelationSchemes$ ***then***
$RelationSchemes = RelationSchemes \cup R_C$;
insertRecords(R_C, C);
end;
if isObjectPropertyExpression($Atom$) ***then begin***
$R_P =$ createRelation($Atom$) as Relation;
if not exists(R_P ***in*** $RelationSchemes$ ***then***
$RelationSchemes = RelationSchemes \cup R_C$;
C.nestedStoredProc_part = C.nestedStoredProc_part $\cup$ createInsertRecords(R_P, C);
end;
if isDataValuedPropertyExpression($Atom$) ***then begin***
$R_D =$ createStoredProc($Atom$) as StoredProc;
C.nestedStoredProc_part = C.nestedStoredProc_part $\cup$ createUpdateRecords(R_D, C);
end;
end for;
end for;
return DatabaseScheme;
end;

Рис. 5. Обобщенный алгоритм интерпретации онтологий и запросов к базам знаний с использованием реляционной модели хранения данных

Также использование КО моделей позволило разработать и реализовать алгоритм преобразования запросов на Semantic Query Web Rule Language, к хранилищу знаний в формате OWL DL + SWRL, в запросы на SQL/SPL, к реляционному представлению БЗ ПрО. Отображение хранилищ знаний в реляционную модель хранения данных позволило реализовать компоненты для ИОД с высокой производительностью обработки данных и знаний, в том числе на базе встроенных систем. Так, за счет использования предложенной методики и соответствующих структурно-алгоритмических решений при организации и функционировании КС для ИОД на ряде предприятий скорость выполнения интерпретированных на основе разработанных моделей и предложенным методом запросов к онтологическим моделям выросла на 35–40 %, что подтверждено актами внедрения.

ВЫВОДЫ

На основании разработанной ранее методологии проектирования хранилищ данных и знаний для решения задач интеллектуальной обработки данных на основе категориально-онтологического подхода, разработана модель и метод отображения хранилища знаний в формате онтологической модели, представленной средствами Ontology Web Language с аксиомами в виде высказываний Descriptive Logic, вместе с правилами, выраженными на Semantic Web Rule Language, в реляционное представление базы знаний. Отличие разработанной модели, метода и методики на их основе, а также предложенных алгоритмов разработки и отображения концептов и правил, состоит в использовании в ходе такой интерпретации категориально-онтологических моделей хранилищ знаний и данных в различных форматах представления. Реализация такого отображения позволила реализовать программные компоненты компьютерных систем с высокой производительностью обработки данных и знаний, в том числе на базе встроенных систем. Полученные в данном исследовании результаты позволили реализовать ряд программных компонентов компьютерных систем для производственных предприятий. Оценка системного эффекта от внедрения предложенной методики показала, что скорость выполнения интерпретированных на основе реляционной модели хранения запросов к онтологическим моделям существенно возросла.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Sahaida P. Development of methodology for data and knowledge warehouse design in computer systems for intellectual data processing / P. Sahaida // Technology audit and production reserves. Information and Control Systems. – 2018. – Vol 1. – No 2(39). – P. 10–15.

2. Сагайда П. І. Методологія проектування сховищ даних і знань на основі категоріально-онтологічних моделей / П. І. Сагайда // Сучасні проблеми математичного моделювання, обчислювальних методів та інформаційних технологій: Матеріали міжнародної наукової конференції. – Рівне : РДГУ, 2018. – С. 105–106.

3. Сагайда П. И. Категориально-онтологическое моделирование интеллектуальной обработки данных для математического обоснования результатов инженерии знаний / П.И. Сагайда // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2017. – № 4. – С. 149–158.

4. Allemang D. Semantic Web for the working ontologist: effective modeling in RDFS and OWL / D. Allemang, J. Hendler. – Waltham, USA : Morgan Kaufmann, 2011. – 384 p.

5. Larman C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development / C. Larman. – Addison Wesley Professional, 2004. – 736 p.

6. Johnson M. Entity-relationship-attribute designs and sketches/ M. Johnson, R. Rosebrugh, R. J. Wood // Theory and Applications of Categories. – 2002. – Vol. 10. – No. 3. – P. 94–112.

Статья поступила в редакцию 12.03.2018 г.