

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

УДК 004.652.8

DOI: 10.17223/19988605/51/11

А.М. Бабанов, А.В. Петров

ДЕКЛАРАТИВНОЕ ОПРЕДЕЛЕНИЕ УСЛОВИЙ НЕПРОТИВОРЕЧИВОСТИ В ERM-МЕТАСХЕМЕ

Продолжено описание реализации ERM-репозитория и представлен альтернативный способ создания механизма, обеспечивающего непротиворечивость схем. Он заключается в расширении метасхемы ERM-модели за счет включения в нее условий непротиворечивости элементов схем. При таком декларативном подходе (в противоположность преимущественно процедурному подходу, реализованному в текущем прототипе репозитория) появление новых условий не будет сопровождаться программированием соответствующих проверок элементов репозитория, так как верификацию будет выполнять универсальная программа интерпретации метасхемы.

Ключевые слова: ERM-модель; непротиворечивость репозитория; схема данных; метасхема модели данных; разработка, управляемая моделями.

Современные системные методики проектирования схем баз данных (БД) предполагают в качестве языка первоначальной формализации описания предметной области (ПрО) ту или иную семантическую модель данных. Наряду с удобным и выразительным языком модели программные CASE-инструменты проектировщика (Computer Aided System Engineering – CASE) предоставляют также возможность автоматической генерации SQL-скрипта для создания БД в той или иной системе управления базами данных (СУБД).

Для информационного обеспечения этих задач такое CASE-средство использует репозиторий – хранилище всех аналитических и проектных артефактов разработки. Важной задачей, стоящей перед разработчиками репозитория, является обеспечение непротиворечивости всех хранящихся в нем схем.

Характерной чертой большинства информационных систем является преимущественно процедурный, алгоритмический способ удовлетворения функциональных требований. Более изящным и, главное, гибким является решение, основанное на принципах разработки, управляемой моделями (Model Driven Development – MDD) [1–5]. При таком подходе строится декларативная модель, описывающая требуемую функциональность, а универсальная программа на ее основе осуществляет необходимые действия. Изменение требований влечет за собой лишь коррекцию модели, в переписывании универсальных программ, реализующих модель, нет необходимости.

Первая реализация исследовательского прототипа репозитория ERM-модели [6, 7] выполнялась в основном традиционным способом. Исключение составляли, пожалуй, проверки, которые осуществляются реляционной СУБД на основании ограничений целостности репозитория, декларированных в его схеме.

Выразительная мощьность ERM-модели [8] позволяет представить в метасхеме репозитория все условия непротиворечивости элементов ERM-схем. Это побудило исследователей ERM-модели попытаться реализовать механизм проверок схем на непротиворечивость в соответствии с MDD.

Такой подход обеспечит гибкость разработки и минимизацию программного кода и его изменения, что является немаловажным фактором эффективности в условиях постоянного совершенствования правил структуризации и задания ограничений целостности ERM-модели. Первым шагом на этом пути является расширение ERM-метасхемы соответствующими описаниями условий непротиворечивости. Вторым шагом будет реализация универсальной программы верификации.

1. Понятие метасхемы модели данных

Р. Велке [9–10] так определяет основные понятия, связанные с моделями репозитория:

«Метамодель – набор примитивов, каждый для описания отдельного типа предметов, которые могут быть представлены в любом репозитории, созданном в терминах этой метамодели. Например, если предположить использование в качестве метамодели ER-модели, такими примитивами будут “Сущность”, “Связь” и “Атрибут”. Другими словами, выбор метамодели диктует выбор категорий, в терминах которых будет описываться вся аналитическая и проектная информация о системе.

Каждая метамодель как набор примитивов или категорий ссылается на них, как на метатипы. Так, среди метатипов ER-метамодели есть такой, как “Сущность”. Метасхема утверждает, какие экземпляры метатипов и их взаимосвязи допустимы в конкретном репозитории с точки зрения метамодели. Таким образом, метамодель определяет язык метасхемы. Экземпляр метасхемы (где, например, определен такой экземпляр метатипа, как “Человек”) соответствует схеме обычных данных» [9. С. 3–4].

Репозиторий Oracle Designer (OD), как и любой другой репозиторий, имеет свою метасхему. Все структуры данных, находящиеся в репозитории, можно разделить на три основных метатипа, составляющих метамодель OD [6]:

- элементные типы,
- ассоциативные типы,
- текстовые типы.

Изначально в репозитории OD представлены метасхемы таких моделей, как реляционная модель данных, ER-модель данных в нотации Баркера, функциональные модели потоков данных и иерархий функций и некоторые другие. В ходе реализации прототипа ERM-репозитория на базе репозитория OD метасхема последнего была расширена метатипами структурных понятий ERM-модели [6].

2. ERM-метасхема на языке расширенной ER-модели

Как отмечалось в [8], ERM-модель отличается следующими особенностями: богатые выразительные способности, синонимия элементов схемы, возможность наличия в схеме одновременно нескольких представлений одних и тех же явлений предметной области. Богатство языка ERM-модели обеспечивают ее базовые понятия – классы и отображения (типы); объекты и экземпляры отображений (знаки). Однако для удобства проектирования в модель добавлены в качестве производных традиционные структурные понятия – сущность, связь, атрибут [Ibid.]. На рис. 1 представлена каркасная диаграмма ERM-метасхемы на языке расширенной ER-модели.

Данный вид метасхемы принято называть каркасным, потому что на нем представлены структурные понятия ERM-модели и их взаимосвязи. Также уже обозначены элементы метасхемы, с помощью которых будут определяться условия непротиворечивости схем данных – отношения между классами, отношения между отображениями, операции над классами и отображениями.

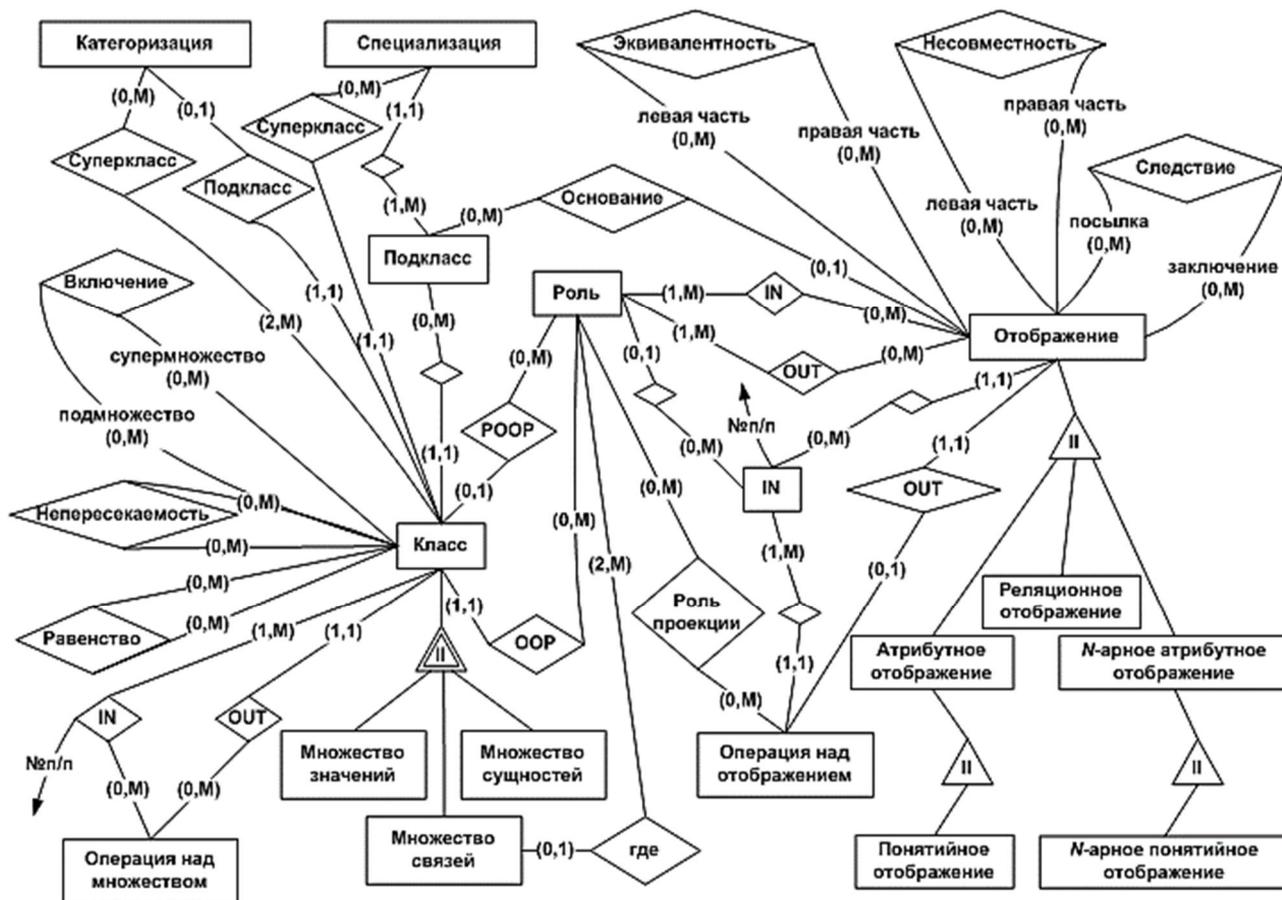


Рис. 1. Каркасная диаграмма метасхемы ERM-модели
 Fig. 1. Skeleton diagram of the ERM model metaschema

Есть на каркасной диаграмме метасхемы и условия непротиворечивости, заданные в виде ограничений на максимальные и минимальные кардинальные числа отображений (пометки на дугах). К сожалению, язык ER-модели может передать только эти ограничения. Для определения всех остальных условий непротиворечивости необходимо переходить на более мощный язык отображений, т.е. использовать в качестве метамодели саму ERM-модель.

3. Уточнение используемого понятийного базиса

В технологии БД сложилась следующая терминологическая основа. Состояние ПрО описывают данные. В табличном представлении – это тела таблиц. Строки соответствуют объектам или связям, которые определяются значениями характеристик. Для образования из данных осмысленной информации используются их интерпретации – имена таблиц и столбцов. Эти интерпретации определяют структуру и семантику (смысл) данных. Вместе с ограничениями целостности, реализующими бизнес-правила ПрО, они составляют схему данных, которая первоначально (до ввода самих данных) передается СУБД.

Таким образом, конечная цель технологии БД – актуальные структурированные данные, описывающие объекты и связи между ними. Схема БД на уровне типов описывает представляющие интерес понятия о предметах и их кортежах, а также представляющие интерес признаки этих предметов. Для достижения указанной цели каждая СУБД предоставляет пользователям конкретную модель, которая состоит из правил построения схем (структур и ограничений целостности), а также операций над данными.

В репозиториях CASE-систем предметную область составляют сами схемы данных. Такие системы, как правило, создаются по технологии БД и представляют собой не что иное, как специали-

зированные системы БД. Элементы схем данных «обычных» ПрО (типы) превращаются в их БД (репозитории) в данные (знаки). Например, множество сущностей ЧЕЛОВЕК описывается строкой в таблице МНОЖЕСТВА СУЩНОСТЕЙ. Атрибут множества сущностей ЧЕЛОВЕК с именем Дата Рождения превращается в строку таблицы АТТРИБУТЫ.

В зависимости от используемой метамодели (в примере фигурировали структурные понятия ER-модели Чена) схему репозитория (хранилища схем) будут составлять соответствующие понятиям именно этой модели структуры.

Что же собой представляют ограничения целостности этой схемы? Например, такое, как обязательность наличия ссылки на одно и только одно множество сущностей для каждой роли любого множества связей. С точки зрения данных репозитория – это действительно ограничение целостности. И если репозиторий реализован как реляционная БД, то это ограничение целостности декларативно определяется как обязательность (NOT NULL) значения внешнего ключа (FOREIGN KEY).

Для таких понятий предпочтительней использовать термин «условие непротиворечивости», потому как с точки зрения схемы ПрО это не что иное, как необходимое условие синтаксической корректности. И звучит оно, как грамматическое правило модели данных – «каждую роль множества связей может и должно играть одно и только одно множество сущностей».

4. Пополнение метасхемы ERM-модели дополнительными условиями непротиворечивости

Что касается метасхемы модели, в ней в идеале должны быть указаны все ограничения на допустимые схемы, которые метамодель может отразить. Тот факт, что ERM-модель позиционируется как одна из наиболее богатых по описательным возможностям, позволяет предполагать, что использование ее в качестве метамодели позволит отразить все дополнительные условия непротиворечивости схем.

Полную метасхему ERM-модели невозможно представить в виде одной диаграммы, это колоссально усложнило бы процесс ее восприятия. Взаимосвязи основных структурных понятий представлены на каркасной диаграмме (см. рис. 1). Для каждого условия непротиворечивости используется отдельная диаграмма. На ней показаны структурные понятия каркасной диаграммы, составляющие контекст условия, и те особые взаимосвязи, которые, собственно, и отражают его суть.

Для иллюстрации декларативных принципов задания условий непротиворечивости приведем несколько «простейших» таких диаграмм.

4.1. Проверка наличия у множества связей не менее двух ролей, которые играют исключительно сущности

На каркасной диаграмме метасхемы зафиксирована обязательность наличия у множества связей не менее двух ролей (минимальное кардинальное число, равное двум, на ребре между множеством сущностей «Роль» и множеством связей «где»). Однако то, к каким именно типам классов они должны относиться, не уточняется. Между тем под множеством связей в ERM-модели понимается такая же структура, как и в ER-модели. И вступать в связи могут только сущности. Следовательно, играть роли множеств связей могут только элементы множеств сущностей. Данное условие задается схемой, показанной на рис. 2.

Реляционное отображение «роль множества связей» позволяет перейти к ролям, связанным с множеством связей. Реляционное отображение «класс роли» позволяет получить класс роли в связи «ООР». Соответственно, результатом композиции этих двух отображений будет отображение, позволяющее перейти от множества связей непосредственно к классам, элементы которых играют роли этого множества связей.

Отображение «тип класса» – частичное функциональное (так как класс может быть неспециализирован). Отсюда следует, что композиция отображения, являющегося результатом объединения отображений «операнд операции» и «результат операции», и отображения «тип класса» также должна быть частичной функциональной.

4.3. Проверка совпадения типов классов суперклассов и подкласса категоризаций

Данная проверка с точки зрения реализации схожа с предыдущей проверкой. Композиция отображения, являющегося результатом объединения отображений «суперкласс категоризации» и «подкласс категоризации», и отображения «тип класса» также должна быть частичной функциональной (рис. 4).

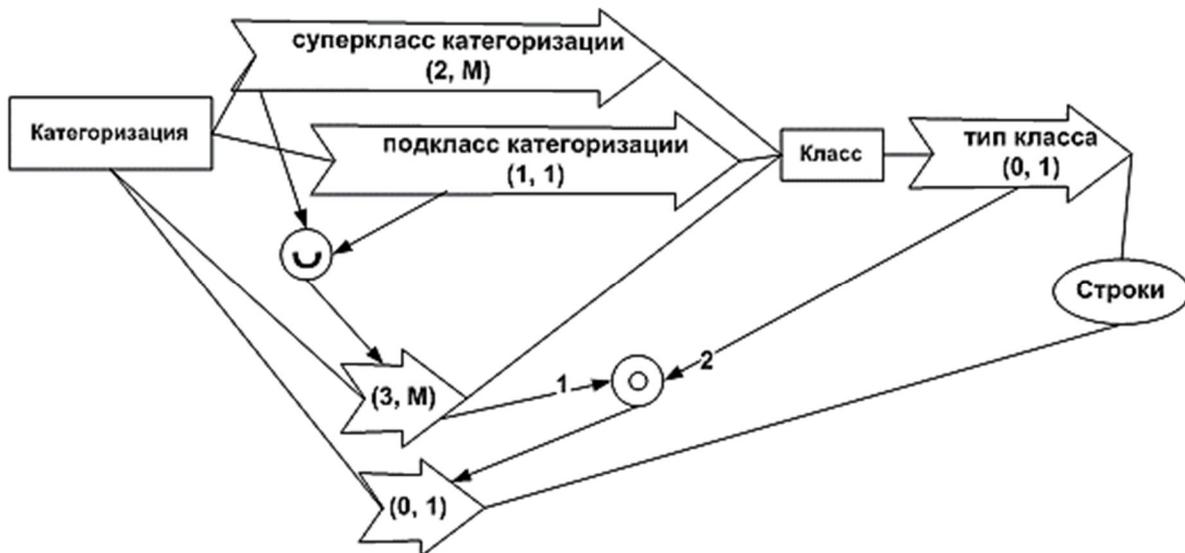


Рис. 4. Диаграмма условия непротиворечивости
Fig. 4. Diagram of the consistency condition

4.4. Проверка наличия связи с ролью проекции только у операций проекции

У операции проекции в качестве дополнительных параметров выступают роли, которые в схеме соединяются с кругом, обозначающим операцию, наряду с отображением-аргументом и отображением-результатом. На дугах указываются пометки In или Out. Пометка In означает, что данная роль является ролью прообразов, Out – образов. Пример проекции на роль приведен на рис. 5.

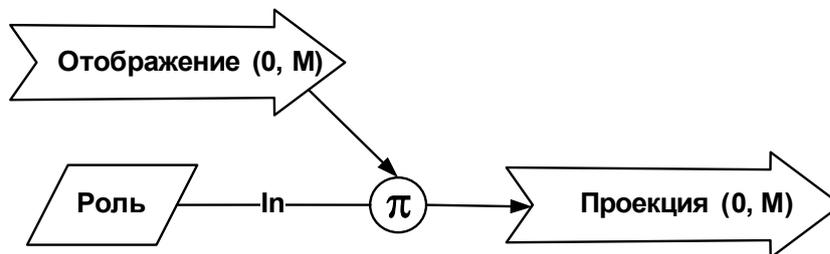


Рис. 5. Операция проекции на роль
Fig. 5. Operation of projection on the role

Каркасный вариант метасхемы ERM-модели не задает ограничения на тип операции, который может использоваться в рамках множества связей «Роль проекции», однако очевидно, что в этих связях могут участвовать только операции-проекции. Данное ограничение описывается схемой, показанной на рис. 6.

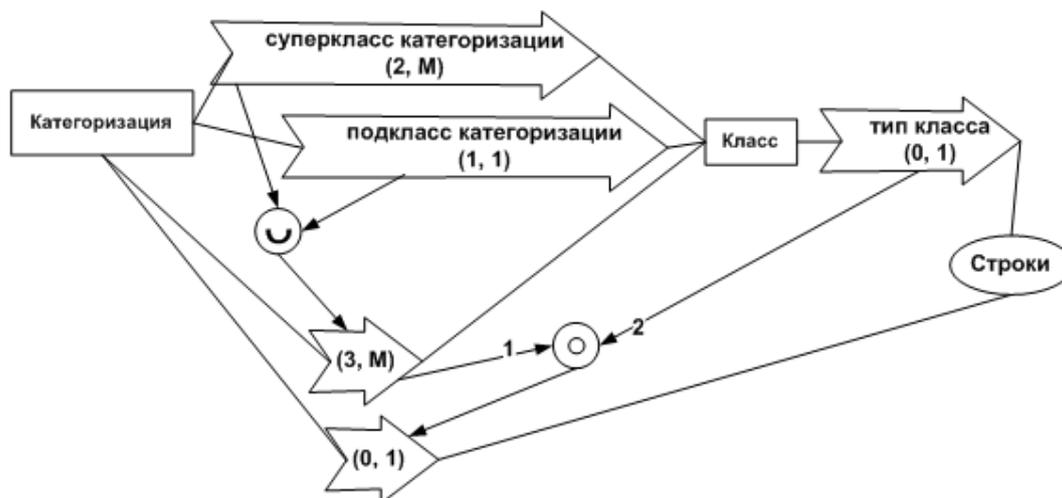


Рис. 6. Диаграмма условия непротиворечивости
 Fig. 6. Diagram of the consistency condition

Идея заключается в том, чтобы показать, что все операции, участвующие в связи «Роль проекции», имеют тип «Проекция», и наоборот. Для того чтобы это показать, необходимо построить отображение из ролей в операции над отображением в контексте множества связей «Роль проекции», с другой стороны, отображение строки «Проекция» в операции этого типа. ОЗО обоих производных отображений должны совпадать, что и определяет требуемое условие.

Заключение

Приведенные в статье примеры представления условий непротиворечивости в метасхеме ERM-модели демонстрируют возможность ее использования в качестве метамодели. Это, в свою очередь, создает необходимые условия для применения MDD-подхода к реализации механизма верификации схем в ERM-репозитории.

При таком подходе в репозитории строится декларативная метасхема ERM-модели, описывающая условия непротиворечивости схем, а универсальная программа на ее основе осуществляет необходимые проверки схем. Изменения, вносимые в правила структуризации и задания ограничений целостности ERM-модели, влекут за собой лишь коррекцию метасхемы, в переписывании универсальных программ, реализующих проверки, нет необходимости.

ЛИТЕРАТУРА

1. Brambilla M., Cabot J., Wimmer M. Model-Driven Software Engineering in Practice. 2nd ed. Morgan & Claypool Publishers, 2017.
2. Stahl T., Völter M. Model-Driven Software Development: Technology, Engineering, Management. John Wiley & Sons Ltd., 2006.
3. Beydeda S., Book M., Gruhn V. Model-Driven Software Development. Springer, 2005.
4. Pastor O., Molina J. Model-Driven Architecture in Practice: a Software Production Environment Based on Conceptual Modeling. Springer, 2007.
5. Whittle J. Model-Driven Development: a Practical Approach. Chapman and Hall/CRC, 2012. (Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series).
6. Бабанов А.М., Петров А.В. Реализация репозитория ERM-модели в CASE-системе Oracle Designer // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2017. № 41. С. 47–54.
7. Бабанов А.М., Петров А.В. Проблема непротиворечивости схем данных и ее решение в ERM-репозитории // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2018. № 43. С. 72–80.
8. Бабанов А.М. Семантическая модель «Сущность–Связь–Отображение» // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2007. № 1. С. 77–91.
9. Welke R. The case repository: more than another database application. Meta Systems, Ltd., 1988.
10. Welke R. The case repository: more than another database application // Challenges and strategies for research in systems development. New York : John Wiley & Sons Inc., 1992. P. 181–218.

Поступила в редакцию 29 августа 2019 г.

Babanov A.M., Petrov A.V. (2020) DECLARATIVE DEFINITION OF CONSISTENCY CONDITIONS IN ERM METASHEMA. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie vychislitel'naya tekhnika i informatika* [Tomsk State University Journal of Control and Computer Science]. 51. pp. 94–101

DOI: 10.17223/19988605/51/11

A characteristic feature of most information systems is mainly a procedural, algorithmic way of satisfying the functional requirements of the customer. Solution based on the principles of Model Driven Development (MDD) is more elegant, and, most importantly, flexible. In this case, a declarative model is constructed that describes the required functionality, and a universal program carries out the necessary actions based on this model. Changing the requirements entails only a correction of the model; there is no need to rewrite universal programs that implement the model.

The first implementation of the research prototype of the ERM model repository was carried out mainly in the traditional way. The exception was, perhaps, the checks that are performed by the relational DBMS based on the integrity constraints of the repository declared in its schema.

The expressive power of the ERM model allows representing all the conditions for the consistency of elements of ERM schemas in the repository metaschema. This prompted researchers of the ERM model to try to implement a mechanism for checking consistency schemas in accordance with MDD.

The first step in this direction is to expand the ERM metaschema with appropriate descriptions of the consistency conditions. The second step will be the implementation of a universal verification program.

As for the model's metaschema, ideally, it should contain all the constraints on the permissible schemas that the metamodel can reflect. The fact that the ERM model is positioned as one of the richest in descriptive capabilities suggests that using of it as a metamodel will allow reflecting all the additional conditions for the consistency of schemas.

It is impossible to present the complete metaschema of the ERM model in the form of a single diagram; this would greatly complicate the process of its perception. The interconnections of the basic structural concepts are presented in a skeleton diagram. A separate diagram is used for each consistency condition. It shows the structural concepts of the skeleton diagram that make up the context of the condition, and those special relationships that, in fact, reflect its essence.

To illustrate the declarative principles of defining the consistency conditions, several “simple” such diagrams are given.

The examples of the presentation of the consistency conditions in the ERM model metaschema presented in the article demonstrate the possibility of its use as a metamodel. This creates the necessary conditions for applying the MDD approach to the implementation of the schema verification mechanism in the ERM repository. It remains to create a program for the interpretation of the metaschema elements reflecting these conditions.

Keywords: ERM model; repository consistency; data schema; data model metaschema; model driven development.

BABANOV Alexey Mikhailovich (Candidate of Technical Sciences, Associate Professor, National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: babanov2000@mail.ru

PETROV Alexander Viktorovich (National Research Tomsk State University, Tomsk, Russian Federation).

E-mail: flamingcode@mail.ru

REFERENCES

1. Brambilla, M., Cabot, J. & Wimmer, M. (2017) *Model-driven software engineering in practice*. 2nd ed. Morgan & Claypool.
2. Stahl, T. & Völter, M. (2006) *Model-driven software development: technology, engineering, management*. John Wiley & Sons Ltd.
3. Beydeda, S., Book, M. & Gruhn, V. (2005) *Model-Driven Software Development*. Springer.
4. Pastor, O. & Molina, J. (2007) *Model-driven architecture in practice: a software production environment based on conceptual modeling*. Springer.
5. Whittle, J. (2012) *Model-Driven Development: A Practical Approach (Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series)*. Chapman and Hall/CRC.
6. Babanov, A.M. & Petrov, A.V. (2017) Implementation of the ERM-model repository in CASE-system Oracle Designer. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 41. pp. 47–54. DOI: 10.17223/19988605/41/6
7. Babanov, A.M. & Petrov, A.V. (2018) Problem of data scheme consistency and its solution in the ERM repository. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 43. pp. 72–80. DOI: 10.17223/19988605/43/9
8. Babanov, A.M. (2007) Semantic model “Entity-Relationship-Mapping”. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitel'naya tekhnika i informatika – Tomsk State University Journal of Control and Computer Science*. 1. pp. 77–91.
9. Welke, R. (1988) *The case repository: more than another database application*. Meta Systems, Ltd.
10. Welke, R. (1992) The case repository: more than another database application. In: Cotterman, W.W. & Senn, J.A. (eds) *Challenges and strategies for research in systems development*. New York: John Wiley & Sons Inc. pp. 181–218.