**Инженерный вестник Дона**, №3 (2020)
ivdon.ru/ru/magazine/archive/n3y2020/6343

# Анализ особенностей реализации объектно-ориентированной парадигмы в проектах с открытым исходным кодом

*Е.С. Голубцов[1], С.В. Клименков[2], Е.А. Цопа[2], А.Е. Харитонова[2], В.В. Николаев[2], А.В. Гаврилов[2]*

*[1]ООО «Ру Квад Код», Санкт-Петербург*
*[2]Национальный исследовательский университет ИТМО, Санкт-Петербург*

**Аннотация:** В статье проводится анализ реализации базовых принципов объектно-ориентированного моделирования в открытых проектах, содержащих контактную информацию относительно эталонной модели OASIS UBL Party. Используя формальные критерии оценки соответствия моделей проектов эталонной модели выполнен анализ программных продуктов с открытым исходным кодом при помощи количественных характеристик, трансформации концептуальных графов и кластерного анализа. На основании результатов анализа моделей предлагаются способы уменьшения смыслового несоответствия между доменными моделями программных продуктов и эталонными моделями.

**Ключевые слова:** объектно-ориентированный анализ и проектирование, эталонная модель, OASIS UBL.

## Введение

В основе любого программного продукта, созданного на объектно-ориентированном языке программирования, лежит модель, которая отображает предметную область, используемую данным приложением. Большинство бизнес-процессов, реализуемых приложениями, действуют в ограниченном множестве предметных областей, для которых уже построены эталонные модели, универсальные решения общих задач, наиболее полно использующие средства объектно-ориентированного проектирования [1].

Однако, во многих случаях реальная модель приложения сильно отличается от эталонной, а возможности объектно-ориентированной парадигмы при отображении предметной области и реализации бизнес-процессов используются не полностью или не используются вовсе. Одним из способов повысить качество программного обеспечения является проверка корректности еще на этапе разработки модели, где используются методы верификации [2]. Кроме исследования качества, ведется работа над методами

сравнения структурных и функциональных информационных моделей. При этом довольно часто используется трансформация моделей в аналитическую [3], логическую [4] или в документальную форму [5], либо отображение моделей в виде функциональных процессов [6].

Вышеперечисленные методы в основном предназначены для попарного сравнения произвольных моделей. При этом, в настоящее время активно используются эталонные модели, которые построены с учетом мнений ведущих экспертов в области программной инженерии. Хотя эталонные модели все равно могут различаться между собой, воплощая различные принципы архитектурных решений, они могут быть использованы в качестве образца для оценки качества реальных моделей. Исследователями разработаны методы сравнения и оценки эталонных моделей между собой на основе формальных критериев, например, основанных на конгруэнтности их целевых, контекстных и дизайнерских характеристик [7]. Однако, при этом недостаточно проработаны количественные и качественные показатели оценки структурного подобия иерархии классов реальных моделей приложений с открытым исходным кодом и эталонных моделей, которые являются наиболее полной реализацией концепции взаимодействия объектов реального мира.

Для разработки такой системы показателей необходимо исходить из позиции, что вся разработка объектно-ориентированных приложений представляет собой моделирование реального мира при помощи отражения связей предметной области в абстракции модели. Следовательно, программные продукты должны не только содержать ту же информацию, что содержится в моделируемых сущностях, но и обладать набором связей, существующих в реальном мире. Это особенно важно с точки зрения повторного использования кода. Если эталонная модель близко соответствует реальным сущностям, то использование элементов этой

модели сократит трудоемкость разработки нового программного обеспечения в рамках одной и той же предметной области [8].

Одним из направлений, позволяющих учитывать взаимосвязи между понятиями реального мира в разработке программных систем, является использование семантической информации в качестве основного или вспомогательного средства. Существуют методы сравнения программных архитектур с использованием формальной семантики и трансформации UML-диаграмм [9], и логического моделирования и анализа сложных систем [10], однако на данный момент в подобных методах недостаточно используются именно количественные показатели..

В данной работе предлагается подход к формальной количественной оценке соответствия объектно-ориентированных моделей реальных приложений и стандартных эталонных доменных моделей.

Для проведения оценки было отобрано 10 проектов с открытым исходным кодом. Для этих проектов построены доменные модели (диаграммы классов аналитического уровня) для выбранной предметной области, широко используемой как в бизнес-приложениях, так и в пользовательский приложениях - «адресная книга». В качестве эталонной для данной предметной области была выбрана эталонная модель (Party Model), описанная согласно спецификации OASIS Universal Business Language Version 2.2. Блок-схема модели приведена на рис. 1.

## Количественный анализ

Количественный анализ позволяет оценить совпадение элементов доменной модели исследуемого проекта с элементами эталонной. При проведении количественного анализа использовалась адаптированная оценка моделей по критериям, предложенным для сравнения моделей жизненного цикла управления программными проектами [11].

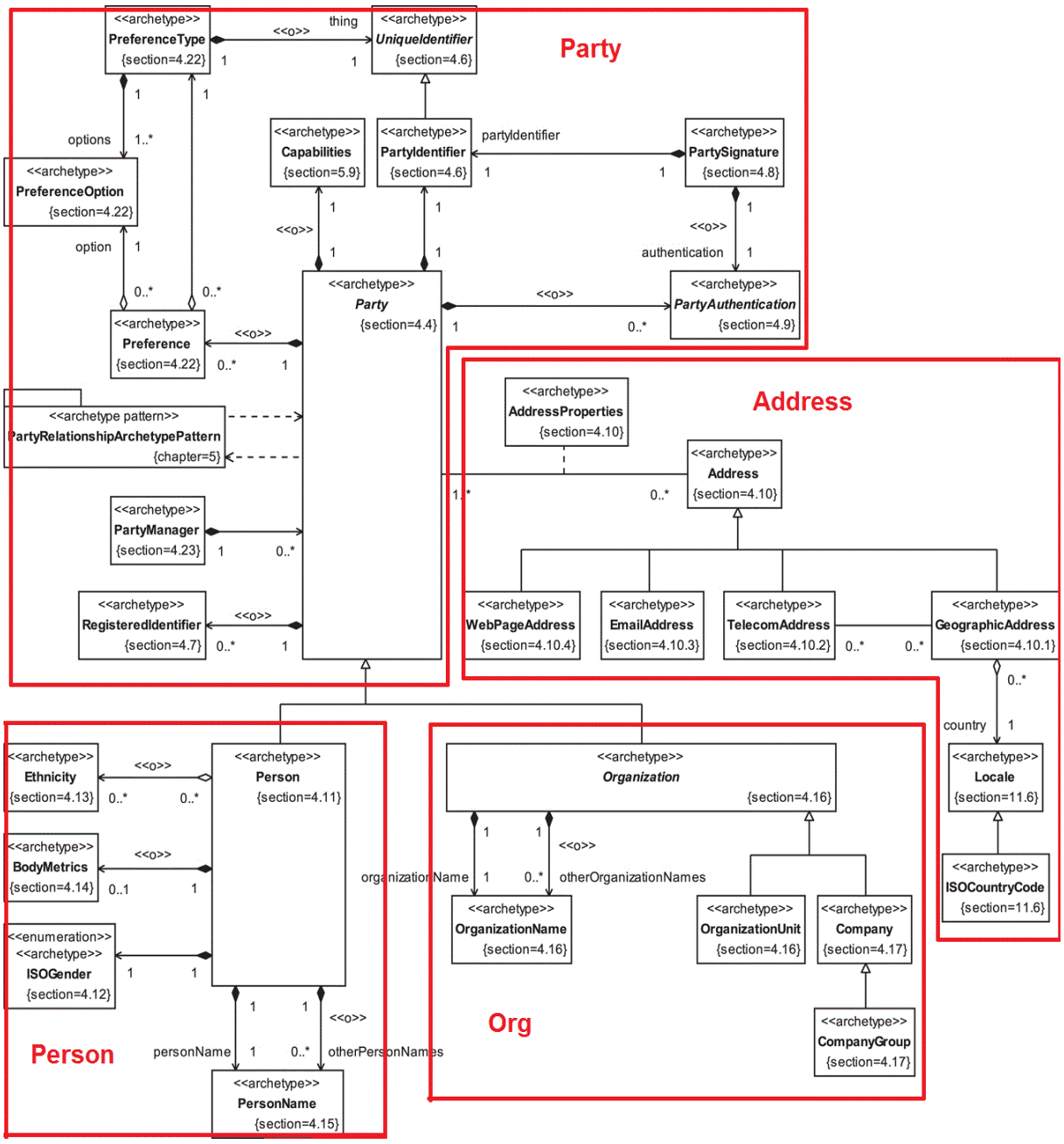


Рисунок 1. Эталонная модель UBL Party

Первый критерий — контекст покрытия *Sc*, оценивает количество семантически схожих полей исследуемой модели по отношению к эталонной модели в процентном соотношении.

$$Sc = \frac{\sum_{i=1}^{n} \varphi c(i)}{n\varphi(max)}$$

где $n$ – количество компонент (полей) в эталонной модели;

$\varphi c(i)$ – соответствие отображаемой компоненты (поля) в исследуемой модели, принимает значение [0...1], где 0 — полное отсутствие компоненты в отображаемой модели, 1 — полное совпадение компоненты в исследуемой и эталонной моделях ;

$\varphi(max)$ – максимально возможное совпадение для компоненты.

Следующий критерий — фактор разработки $Ef$, показывает сходство в доменах в исследуемой модели.

$$Ef = \frac{MPd}{MRd}$$

где $MRd$ – количество доменов в эталонной модели;

$MPd$ – количество доменов исследуемой модели со схожим для эталонной модели семантическим значением.

И последний критерий — структурная связность $StrC$ характеризует связь между доменами в моделях, чем меньше данная величина, тем лучше (.

$$StrC = \frac{n}{N(N-1)}$$

где $N$ – число доменов в исследуемой модели;

$n$ – число однонаправленных связей между доменами.

Результаты проведенного анализа представлены в табл. 1.

## Сравнение моделей с помощью концептуального графа

Помимо количественных оценок, важно провести сравнение структур исследуемых моделей. Одним из успешно применяющихся методов для сравнения сущностей, заданных в виде структуры, является их предварительная трансформация в более удобные для этого формы, например, в концептуальные графы. Среди существующих способов применения концептуальных графов можно отметить их использование для моделирования характеристик программных продуктов [12], а также знаний о предметной области [13]. Таким образом, для сравнения структур моделей

предметной области был использован метод, заключающийся в преобразовании доменных моделей в концептуальные графы с целью их последующего анализа.

**Таблица 1. Результаты количественного анализа**

| Показатели<br>Модели | Sc, % | Ef, % | StrC, % |
|---|---|---|---|
| aProject | 42 | 9 | 50 |
| Centraview | 33 | 55 | 21 |
| GanttProject | 15 | 27 | 25 |
| jAddressBook | 68 | 36 | 16 |
| Liferay | 34 | 45 | 13 |
| Liferay Google | 16 | 18 | 25 |
| Openbravo | 21 | 36 | 30 |
| Prj1 | 33 | 45 | 25 |
| Tscclient | 21 | 27 | 15 |
| Wf1 | 16 | 9 | 30 |
| OASIS | 100 | 100 | 10 |

Данный метод выявляет семантически схожие домены, которые будут являться вершинами, на основе выявленных вершин строятся графы, а затем проверяется сходство структур моделей.

Результаты проведенного сравнения показаны на рис. 2. При сравнении моделей Openbravo (M1) и Wf1 (M2) видно, что M1 является подграфом M2. При сравнении моделей Prj1 (M3) и Centraview (M4) можно сказать лишь о частичном сходстве C1-C2-C3. По результатам видно, что наиболее часто встречается именно частичное сходство графов, как во втором примере, несмотря на то, что все модели описывают одну и ту же предметную область.

Это вызвано тем, что во одних моделях одна логическая сущность может быть описана несколькими классами, а в других, наоборот, один класс может содержать информацию о нескольких сущностях.

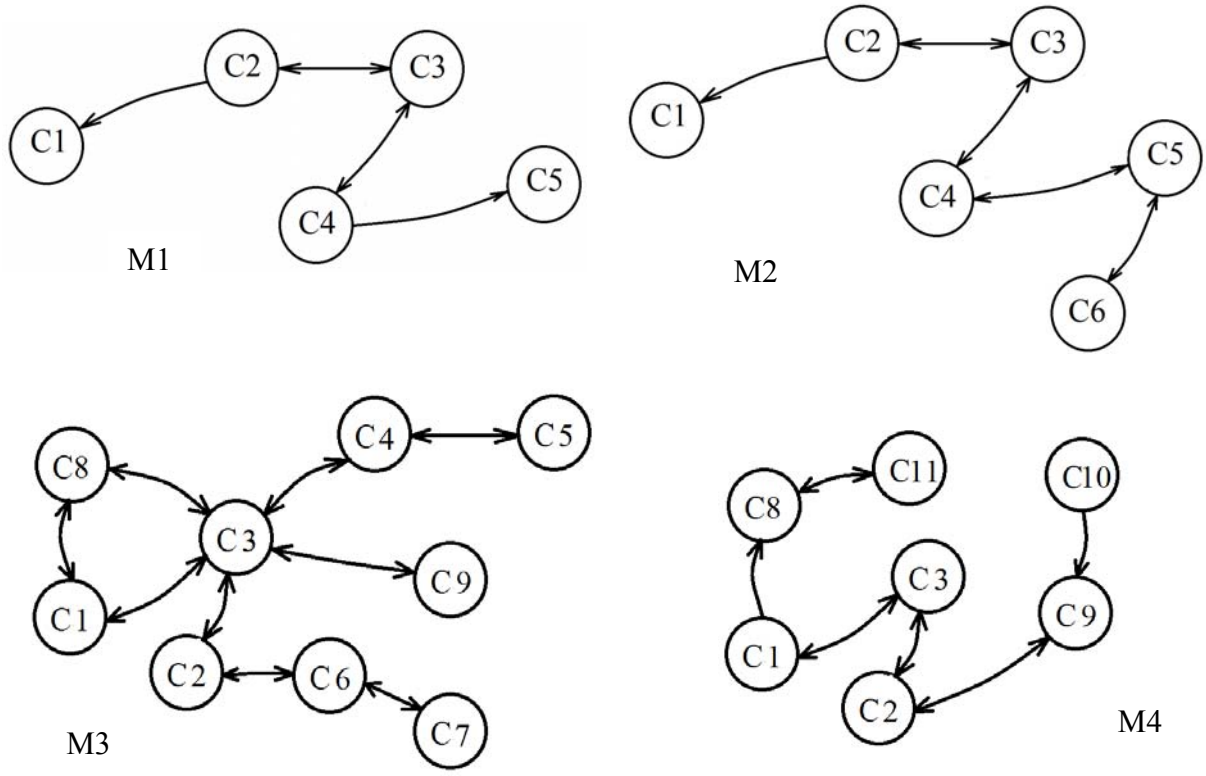


Рис. 1. Сравнение графов моделей проектов

## Кластерный анализ

Таким образом, оценку структурного подобия иерархии классов приложений необходимо осуществлять внутри смысловых и структурных групп классов приложения. Для этого выделим кластеры в эталонной модели, их получилось четыре: *Party, Person, Contact, Org* (рис. 1), и определим схожие кластеры в исследуемых моделях. В моделях могут присутствовать не все кластеры, имеющиеся в эталонной модели. Кластеры, входящие в состав исследуемых моделей, показаны в таблице 2.

**Таблица 2. Кластеры сущностей в моделях**

| Модели | Кластеры | | | |
|---|---|---|---|---|
| | Org | Party | Person | Contact |
| aProject | | + | + | + |
| Centraview | | + | + | + |
| GanttProject | + | + | + | + |
| jAddressBook | | | + | + |
| Liferay | + | + | + | + |
| Liferay Google | + | + | + | + |
| Openbravo | + | + | + | + |
| Prj 1 | + | + | + | + |
| Tscclient | | + | | + |
| Wf1 | + | + | + | + |

**Таблица 3. Результаты совокупного анализа.**

| | Org | | Party | | Person | | Contact | | StrC, % |
|---|---|---|---|---|---|---|---|---|---|
| Модели | Sc,% | Ef,% | Sc,% | Ef,% | Sc,% | Ef,% | Sc,% | Ef,% | |
| aProject | 0 | 0 | 7 | 20 | 100 | 100 | 41 | 25 | 50 |
| Centraview | 0 | 0 | 27 | 40 | 80 | 100 | 38 | 75 | 150 |
| GanttProject | 25 | 50 | 20 | 20 | 40 | 100 | 8 | 25 | 17 |
| jAddressBook | 0 | 0 | 0 | 0 | 100 | 100 | 76 | 100 | 300 |
| Liferay | 50 | 100 | 8 | 49 | 50 | 100 | 80 | 49 | 217 |
| Liferay Google | 100 | 50 | 0 | 5 | 100 | 50 | 0 | 5 | 83 |
| Openbravo | 75 | 100 | 20 | 0 | 75 | 100 | 20 | 0 | 50 |
| Prj 1 | 1 | 1 | 0 | 27 | 1 | 1 | 0 | 27 | 133 |
| Tscclient | 0 | 0 | 0 | 14 | 0 | 0 | 0 | 14 | 350 |
| Wf1 | 75 | 50 | 0 | 19 | 75 | 50 | 0 | 19 | 100 |
| Oasis | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 42 |

## Заключение

Доменные модели приложений, разработанных на объектно-ориентированных языках программирования, в значительной части случаев сильно отличаются в плане структуры и набора связей от описываемых ими объектов реального мира и соответствующих объектам эталонных моделей. Это негативно сказывается на удобстве поддержки и возможности повторного использования кода приложений — каждый раз для описания одних и тех же объектов реального мира приходится заново проектировать доменную модель. Данное отличие можно объяснить, в основном, двумя причинами. Во-первых, во многих приложениях не используется многоуровневая архитектура, т. е., уровень представления не отделен от уровня бизнес-логики и доменной модели. Используемые разработчиками технологии и фреймворки предполагают свои подходы к проектированию объектной модели приложения, часто вступающие в противоречие с принципами описания объектов реального мира в доменной модели этапа анализа предметной области. Во-вторых, использованные в этих проектах языки и технологии программирования никак не способствуют созданию доменных моделей, базирующихся на эталонных, и не содержат эффективных инструментов для их описания.

Существует несколько подходов, которые могут уменьшить отличие реальных проектных моделей от эталонных. К ним относится, например, разработка языка описания объектов реального мира с последующим использованием его в процессе проектирования программного обеспечения, разработка модификаций существующих объектно-ориентированных языков программирования для прозрачного описания программистом сущностей реального мира без конфликта в фазе проектирования с требованиями конкретных фреймворков и технологий, а также реализация описания объектов реального мира в отдельный

программный слой (по аналогии с конфигурацией объектно-реляционного преобразования). Кроме того, учитывая проведенные исследования авторов в области применения семантических сетей для прикладных проектов [14], перспективным способом может стать использование семантической сети с заданной онтологией в качестве основы для эталонной модели предметной области.

## Литература

1. Universal Business Language Version 2.2. Edited by G. Ken Holman. OASIS Standard. URL: docs.oasis-open.org/ubl/UBL-2.2.html
2. Бурякова Н.А., Чернов А.В. Классификация частично формализованных и формальных моделей и методов верификации программного обеспечения // Инженерный вестник Дона, 2010, №4. URL: ivdon.ru/ru/magazine/archive/n4y2010/259
3. Leitner-Fischer F., Leue, S. Quantitative analysis of UML models. // Proceedings of Modellbasierte Entwicklung eingebetteter Systeme (MBEES 2011). Dagstuhl, Germany, pp. 91–100
4. Сергиевский М.В., Конкин А.Ю. Использование дескрипционной логики для оптимизации диаграмм классов UML // Cloud of science. 2017. №3. С. 465-479.
5. Su J., Bao J. Measuring UML model similarity // Proceedings of the 7th International Conference on Software Paradigm Trends, 2012. Pp. 319--323.
6. Jarraya, Y., Debbabi, M. Quantitative and qualitative analysis of SysML activity diagrams. International Journal on Software Tools for Technology Transfer, 2004, 16(4), pp. 399–419.
7. Angelov S., Grefen P.W.P.J., Greefhorst, D. A framework for analysis and design of software reference architectures. Information & Software Technology. 2012. Vol 54, pp. 417-431.

8. Almakadmeh K., Meridji K., Al-Sarayreh K.T. Towards a reference model of software resources quality. Journal of Computer Science. 2018. 14(2). pp. 182-198.
9. Дерюгина О.А. Семантика и семантически эквивалентные трансформации UML-диаграмм классов // Труды МФТИ. 2015. №2 (26). С. 146-155.
10. Ferreira A.L., Machado R.J., Paulk M.C., Quantitative Analysis of Best Practices Models in the Software Domain // Asia Pacific Software Engineering Conference, Sydney, NSW, 2010, pp. 433-442.
11. Ильичева О.А, Технология логического моделирования и анализа сложных систем // Инженерный вестник Дона, 2012, №4. URL: ivdon.ru/ru/magazine/archive/n4p2y2012/1234
12. Bachmeyer R.C., Delugach H.S., A Conceptual Graph Approach to Feature Modeling // Proceedings of 15th International Conference on Conceptual Structures, ICCS 2007, Sheffield, UK, pp 179-191.
13. Valatkaite I., Vasilecas O., Application Domain Knowledge Modelling Using Conceptual Graphs // Information Systems Development. Springer, Boston, MA, 2002, pp. 193-202.
14. Письмак А.Е., Харитонова А.Е., Цопа Е.А., Клименков С.В. Метод автоматического формирования семантической сети из слабоструктурированных источников // Программные продукты и системы, 2016, №3. С. 74-78.

## References

1. Universal Business Language Version 2.2. Edited by G. Ken Holman. OASIS Standard. URL: docs.oasis-open.org/ubl/UBL-2.2.html
2. Buryakova N.A., Chernov A.V. Inzenernyj vestnik Dona, 2010, №4. URL: ivdon.ru/ru/magazine/archive/n4y2010/259

3. Leitner-Fischer F., Leue, S. Quantitative analysis of UML models. Proceedings of Modellbasierte Entwicklung eingebetteter Systeme (MBEES 2011). Dagstuhl, Germany, pp. 91–100
4. Sergievskiy M.V., Konkin A.Yu. Cloud of science. 2017. №3. pp. 465-479.
5. Su J., Bao J. Measuring UML model similarity. Proceedings of the 7th International Conference on Software Paradigm Trends, 2012. Pp. 319-323.
6. Jarraya, Y., Debbabi, M. International Journal on Software Tools for Technology Transfer, 2004, 16(4), pp. 399–419.
7. Angelov S., Grefen P.W.P.J., Greefhorst, D. Information & Software Technology. 2012. Vol 54, pp. 417-431.
8. Almakadmeh K., Meridji K., Al-Sarayreh K.T., Journal of Computer Science. 2018. 14(2). pp. 182-198.
9. Deryugina O.A. Trudy Moskovskogo fiziko-tehničeskogo instituta (gosudarstvennogo universiteta). 2015. №2 (26). pp. 146-155
10. Ferreira A.L., Machado R.J., Paulk M.C., Quantitative Analysis of Best Practices Models in the Software Domain. Asia Pacific Software Engineering Conference, Sydney, NSW, 2010, pp. 433-442.
11. Il'icheva O.A, Inzenernyj vestnik Dona, 2012, №4. URL: ivdon.ru/ru/magazine/archive/n4p2y2012/1234
12. Bachmeyer R.C., Delugach H.S., A Conceptual Graph Approach to Feature Modeling. Proceedings of 15th International Conference on Conceptual Structures, ICCS 2007, Sheffield, UK, pp 179-191.
13. Valatkaite I., Vasilecas O., Application Domain Knowledge Modelling Using Conceptual Graphs. Information Systems Development. Springer, Boston, MA, 2002, pp. 193-202
14. Pis'mak A.E., Kharitonova A.E., Tsopa E.A., Klimenkov S.V. Programmnye produkty i sistemy, 2016, №3. pp. 74-78.