

# ARTIFICIAL INTELLIGENCE ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

УДК 681.3

А.М. Наместников, П.И. Соснин

## МЕТОД КОНЦЕПТУАЛЬНОГО ИНДЕКСИРОВАНИЯ ПРОЕКТНЫХ ДИАГРАММ В ЭЛЕКТРОННЫХ АРХИВАХ<sup>1</sup>

**Наместников Алексей Михайлович**, доктор технических наук, доцент, окончил радиотехнический факультет Ульяновского государственного технического университета. Профессор кафедры «Информационные системы» УлГТУ. Имеет более 80 работ в области автоматизированного проектирования и интеллектуальных систем. [e-mail: nam@ulstu.ru].

**Соснин Петр Иванович**, заслуженный работник высшей школы РФ, доктор технических наук, профессор, окончил радиотехнический факультет Ульяновского политехнического института. Заведующий кафедрой «Вычислительная техника» УлГТУ. Имеет многочисленные труды в области концептуального проектирования автоматизированных систем. [e-mail: sosnin@ulstu.ru].

### Аннотация

Целью данного исследования является разработка метода концептуального индексирования слабоструктурированных проектных диаграмм. Предложенный метод концептуального индексирования основан на использовании специального вида онтологий – онтологии проектных диаграмм. Данная онтология включает в себя семантическое описание нотаций проектных диаграмм и применяемых при разработке программных систем шаблонов проектирования. Разработаны интеллектуальные системы проектирования, которые позволяют с большей эффективностью выполнять задачи структуризации крупных электронных архивов, включающих проектные диаграммы программно-аппаратных комплексов.

Ключевые слова: онтология, проектная диаграмма, UML, концептуальное индексирование.

## METHOD FOR CONCEPTUAL INDEXING OF PROJECT DIAGRAMS IN ELECTRONIC ARCHIVES

**Aleksei Mikhailovich Namestnikov**, Doctor of Science in Engineering; principal lecturer; graduated from the Faculty of Radioengineering at Ulyanovsk State Technical University; Professor of the Department of Information Systems at UISTU; an author of more than 80 scientific papers in the field of automated design and intelligent systems. e-mail: nam@ulstu.ru.

**Petr Ivanovich Sosnin**, Honored Worker of the Higher School of the Russian Federation, Doctor of Engineering, Professor; graduated from the Radioengineering Faculty of Ulyanovsk Polytechnic Institute; Head of

<sup>1</sup> Работа выполнена при финансовой поддержке грантов РФФИ и Правительства Ульяновской области, проекты № 16-47-730742 и № 16-47-732033.

the Department of Computer Science at Ulyanovsk State Technical University; an author of numerous works in the field of conceptual design of computer-aided systems. e-mail: sosnin@ulstu.ru.

## Abstract

The objective of the research is the development of a method for conceptual indexing of semistructured project diagrams. The proposed method of conceptual indexing is based on the use of the special type of ontologies – the ontology of project diagrams. This ontology includes semantic description of project diagram notations and design patterns used in the software system development. Intelligent design systems that allow to perform more efficiently tasks of structuring the large electronic archives including project diagrams of software and hardware systems have been developed.

Key words: ontology, project diagram, UML, conceptual indexing.

## ВВЕДЕНИЕ

В последнее время у исследователей в области программного инжиниринга наблюдается устойчивый интерес к интеллектуальным системам, в основе которых лежит онтологический принцип представления специализированных знаний [1–3]. Онтологическое представление артефактов разработки программного обеспечения (модели, исходные тексты программных модулей) позволяет выполнять автоматизированный анализ программных систем или систем, которые интенсивно используют программное обеспечение (software intensive systems).

Реализация онтологического подхода предполагает построение модели онтологии, которая учитывает аспекты слабоформализованных нотаций моделирования программных систем и особенности архитектурных решений или паттернов проектирования. Контекст предметной области может быть извлечен из текстов комментариев, сопровождающих исходный код.

Фактор принципиальной неполноты проектных диаграмм и комментариев к программному коду на естественном языке влечет необходимость применять для формализации соответствующие математические модели. В данной работе предлагается для этих целей использовать формализм нечетких гиперграфов.

## 1 ФОРМАЛЬНАЯ МОДЕЛЬ ОНТОЛОГИИ ПРОЕКТНЫХ ДИАГРАММ

Для решения задачи интеллектуального анализа проектных диаграмм, входящих в состав проектной документации, необходимо обладать знаниями в области построения формализованных диаграмм (использования нотаций).

На рисунке 1 представлена структура фрагмента онтологии проектных диаграмм, в частности, диаграммы классов языка UML (Unified Modeling Language). Такие знания позволяют выполнять идентификацию применяемых в различных проектах шаблонов проектирования и, следовательно, находить проекты со схожими архитектурными решениями и подходами к реализации программных подсистем автоматизированных си-

стем (АС). На рисунке 1 в качестве примера приведен шаблон проектирования, который называется «Делегирование» [4].

Формально, онтологию проектных диаграмм представим как множество:

$$O^{prj} = \left\{ O_{dc}^{prj}, O_{imp_1}^{prj}, O_{imp_2}^{prj}, \dots, R^{prj}, F^{prj} \right\}, \quad (1)$$

где  $O_{dc}^{prj}$  – онтология диаграммы UML (в работе применяется диаграмма классов),

$O_{imp_i}^{prj}$  – онтологическое представление  $i$ -го шаблона проектирования программных систем.

Рассмотрим основные компоненты онтологии диаграммы UML:

$$O_{dc}^{prj} = \left\{ C^{prj}, R^{prj}, F^{prj} \right\},$$

где  $C^{prj} = \left\{ c_1^{prj}, c_2^{prj}, \dots, c_l^{prj} \right\}$  – множество понятий, определяющих основные элементы диаграмм классов языка моделирования UML (например, «Thing», «Class», «Object», «Interface», «Relationship» и другие);

$R^{prj}$  – множество отношений между понятиями, позволяющее формировать онтологические представления проектных диаграмм при соблюдении соответствующих нотаций (например, определение наименования элемента диаграммы строкового типа «hasAName»; отношение наследования «isA»; отношение, связывающее класс с объектом этого класса «isObjectOf», и другие);

$F^{prj}$  – множество функций интерпретации, определенных на отношениях  $R^{prj}$ .

Онтологическое представление  $i$ -го шаблона проектирования программных систем:

$$O_{imp_i}^{prj} = \left\{ inst(c_1^{prj}), \dots, inst(r_1^{prj}), \dots, r_{sameAs} \right\}.$$

Фактически онтологическое представление отдельного взятого шаблона проектирования представляет собой множество экземпляров понятий и отношений из онтологии проектных диаграмм с добавлением отношения  $r_{sameAs}$ , которое представляет собой встроенное в язык описания онтологий OWL отношение «owl:sameAs». Если указанное отношение связывает

два экземпляра онтологического представления шаблона проектирования, то эти экземпляры считаются одним и тем же объектом.

На рисунке 1 отношением «owl:sameAs» связываются *attribute01* и *object01*. Данный факт означает, что указанный объект некоторого класса (*class2*) является атрибутом другого класса (*class1*).

## 2 КОНЦЕПТУАЛЬНОЕ ИНДЕКСИРОВАНИЕ ПРОЕКТНЫХ ДИАГРАММ

Целью концептуального индексирования проектных диаграмм является формирование индекса, структура которого представлена на рисунке 2. Для решения этой задачи необходимо для индексированной диаграммы определить экземпляры классов онтологии проектных диаграмм и вычислить степень соответствия проектной диаграмме шаблонов проектирования, определенных в онтологии.

Исходными данными будем считать:

- $\{\langle cs_1, dc_1 \rangle, \langle cs_2, dc_2 \rangle, \dots, \langle cs_n, dc_n \rangle\}$  – множество анализируемых проектов электронного архива, каждый из которых включает исходный код  $cs_i$  и диаграмму классов  $dc_i$ ,  $i$  – номер проекта;

- онтологию

$O^{prj} = \{ \langle C^{prj}, R^{prj} \rangle, \{tmp_1, tmp_2, \dots, tmp_m\} \}$ , включающую в качестве элементов множество концептов нотации проектных диаграмм  $C^{prj}$  (диаграммы классов языка UML), множество отношений между классами  $R^{prj}$  и множество шаблонов проектирования  $\{tmp_1, tmp_2, \dots, tmp_m\}$ ;

- $Tz^p = \{ \langle t_1^p, f_1^p \rangle, \langle t_2^p, f_2^p \rangle, \dots, \langle t_l^p, f_l^p \rangle \}$  – техническое задание на новый проект АС  $p$ , прошедший предобработку и представленный в виде набора терминов  $t_1^p, \dots, t_l^p$  с соответствующими частотами  $f_1^p, \dots, f_l^p$ ;

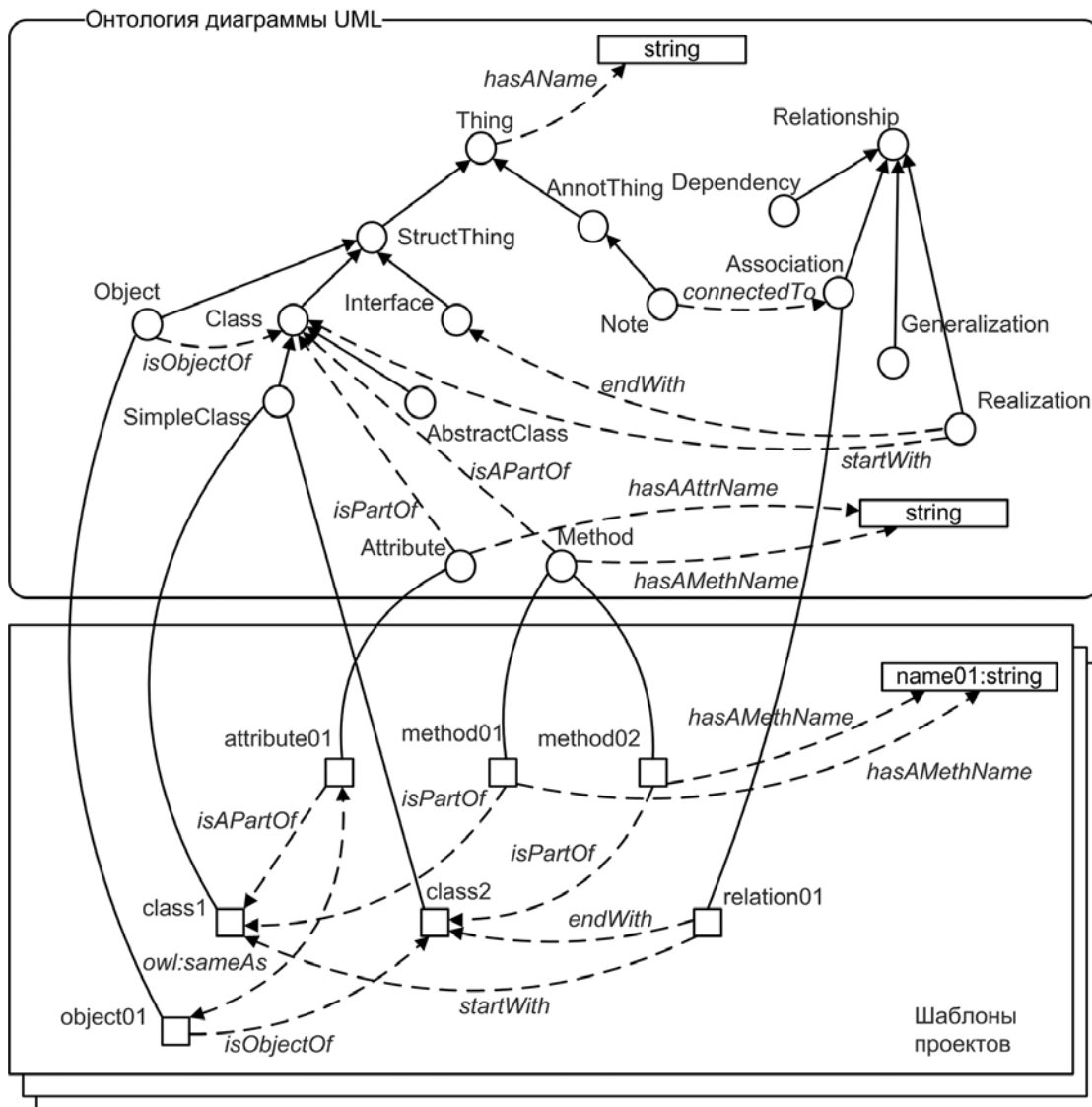


Рис. 1. Структура онтологии проектных диаграмм (включая пример шаблона проектирования)

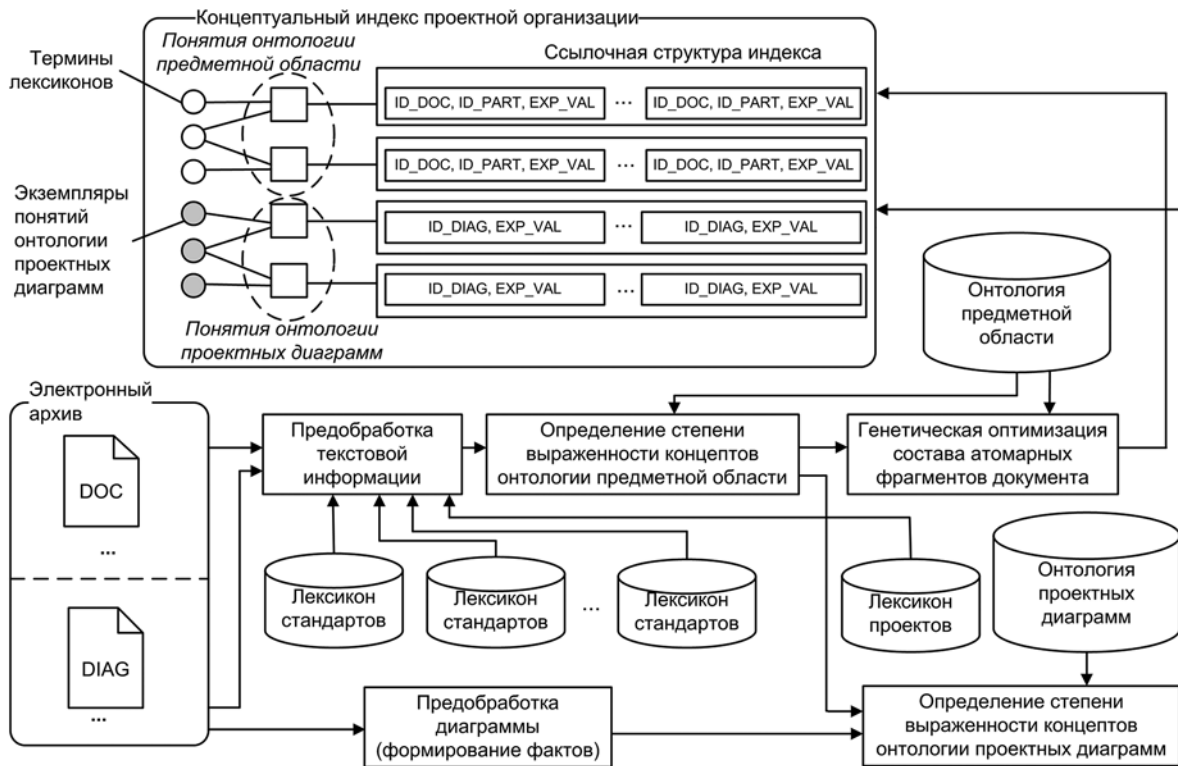


Рис. 2. Формирование концептуального индекса проектной организации

•  $dc^p$  – проектная диаграмма как составная часть нового проекта АС, соответствующая техническому заданию  $Tz^p$ .

Согласно схеме формирования концептуального индекса проектной организации (рис. 2), концептуальное индексирование проектных диаграмм выполняется посредством следующих шагов:

1. Определение контекста проекта.
2. Определение подмножества диаграмм электронного архива, соответствующих контексту проекта.
3. Определение степени соответствия шаблонов проектирования из онтологии  $O^{pj}$  проектной диаграмме  $dc^p$  относительно каждого элемента найденного подмножества диаграмм электронного архива.

Определение контекста проекта выполняется на основе метода концептуального индексирования текстовых информационных ресурсов [5–8]:

$$oV_{tz} = F_{oV}(Tz, O^{dom}, O^{tz}).$$

На вход функции концептуального индексирования текстовой информации  $F_{oV}$  поступает техническое задание  $Tz$ , прошедшее предобработку, онтология предметной области  $O^{dom}$  и тезаурус  $O^{tz}$ .

Результатом концептуального индексирования является множество:

$$oV_{tz} = \left\{ \mu(c_1^{tz})/c_1^{tz}, \mu(c_2^{tz})/c_2^{tz}, \dots, \dots, \mu(c_k^{tz})/c_k^{tz} \right\} = \mu_{oV}(c^{tz}),$$

включающее понятие  $c_i^{tz} \in C^{dom}$  с соответствующим значением функции принадлежности  $i$ -го понятия  $\mu(c_i^{tz})$  техническому заданию  $Tz$  (степень выраженности понятия в тексте технического задания). Полученное множество  $oV_{tz}$  будем считать контекстом реализуемого проекта.

Аналогичным образом концептуальное индексирование выполняется для множества проектов электронного архива  $\{ \langle Sc, Dc \rangle \}$ . Сначала для каждого текста исходного кода  $sc_i \in Sc$  извлекается текст комментария:  $\forall i : tc_i = F_{extcomm}(sc_i)$ , где  $tc_i$  – текстовое представление комментариев к программному модулю  $sc_i$ , прошедшее предобработку:

$$tc_i = \left\{ \langle t_1^{sc_i}, f_1^{sc_i} \rangle, \langle t_2^{sc_i}, f_2^{sc_i} \rangle, \dots, \langle t_s^{sc_i}, f_s^{sc_i} \rangle \right\}.$$

В результате выполнения функции концептуального индексирования получаем онтологическое представление комментариев к исходному коду для каждого программного модуля [9]:

$$oV_{sc_i} = F_{oV}(tc_i, C^{dom}, C^{tz}),$$

$$oV_{sc_i} = \left\{ \mu(c_1^{sc_i})/c_1^{sc_i}, \mu(c_2^{sc_i})/c_2^{sc_i}, \dots, \dots, \mu(c_l^{sc_i})/c_l^{sc_i} \right\} = \mu_{oV}(c^{sc_i}).$$

Множество проектных диаграмм электронного архива проектной организации, соответствующее контексту  $oV_{tz}$ , будем определять следующим образом:

$$Dc|_{oV_{tz}} = \left\{ dc_i : \& \left( \mu_{oV} (c^{sc_i}) \leftrightarrow \mu_{oV} (c^{tz}) \right) \geq 0,5 \right\},$$

где « $\leftrightarrow$ » – операция эквивалентности нечетких множеств, а « $\&$ » – операция конъюнкции по всем  $c^{sc_i}$ ,  $c^{tz} \in C^{dom}$ .

Рассмотрим процесс определения степени выраженности шаблонов онтологии проектных диаграмм, позволяющий сформировать концептуальный индекс проектной организации по представленным в электронном архиве проектным диаграммам (на языке UML). В основе концептуального индекса проектных диаграмм лежит понятие нечеткой меры степени соответствия элементов проектной диаграммы шаблону онтологии (1).

Будем обозначать через  $\mu_{tmp_i}(dc_i)$  степень принадлежности проектной диаграммы  $dc_i$  шаблону  $tmp_i$ .

Аналитически  $\mu_{tmp_i}(dc_i)$  будем определять по следующей формуле:

$$\mu_{tmp_j}(dc_i) = \frac{N(ABox_{dc_i}^{prj})}{N(ABox_{tmp_j}^{prj})},$$

где  $N(ABox_{dc_i}^{prj})$  – количество фактов, которые являются истинными при условии истинности терминологии  $TBox^{prj}$  и соответствуют базе фактов  $ABox_{tmp_j}^{prj}$ ;  $N(ABox_{tmp_j}^{prj})$  – количество фактов шаблона  $tmp_j$ .

Если количество фактов  $N(ABox_{tmp_j}^{prj})$  некоторого шаблона  $tmp_j$  определяется достаточно просто (сум-

мированием количества фактов  $j$ -го шаблона проектирования), то для вычисления  $N(ABox_{dc_i}^{prj})$  необходимо использовать следующий разработанный алгоритм:

*Шаг 1.* Преобразование проектной диаграммы  $dc_i$

электронного архива в набор фактов  $ABox_{dc_i}^{prj}$  вида:

$elem_k^{dc_i} : Concept$ ,

$\langle elem_k^{dc_i}, elem_s^{dc_i} \rangle : Role$ ,

где *Concept* – понятие, определенное в  $TBox^{prj}$ ;

*Role* – роль, определенная в  $TBox^{prj}$ ;

$elem_k^{dc_i}, elem_s^{dc_i}$  – экземпляры понятий, извлеченные из проектной диаграммы  $dc_i$ .

*Шаг 2.* Определение набора базовых классов из  $ABox_{dc_i}^{prj}$  относительно шаблона  $tmp_j$ .

Базовым классом будем называть такой экземпляр  $elem_k^{dc_i}$  понятия «Class» (или его дочернего понятия «Subclass») из  $ABox_{dc_i}^{prj}$ , который соответствует некоторому экземпляру  $cls_i^{tmp_j} \in Class$  из  $ABox_{tmp_j}^{prj}$  и для которого в шаблоне  $tmp_j$  имеет место максимальное количество фактов вида:

$elem_k^{dc_i} : Concept$ ,

$\langle elem_k^{dc_i}, * \rangle : Role$ ,  $\langle *, elem_k^{dc_i} \rangle : Role$ .

Определенный набор базовых классов проектной диаграммы  $dc_i$  относительно шаблона  $tmp_j$  запишем в виде множества:

$$\left\{ \langle elem_1^{dc_i}, cls_1^{tmp_j} \rangle, \langle elem_2^{dc_i}, cls_2^{tmp_j} \rangle, \dots \right\},$$

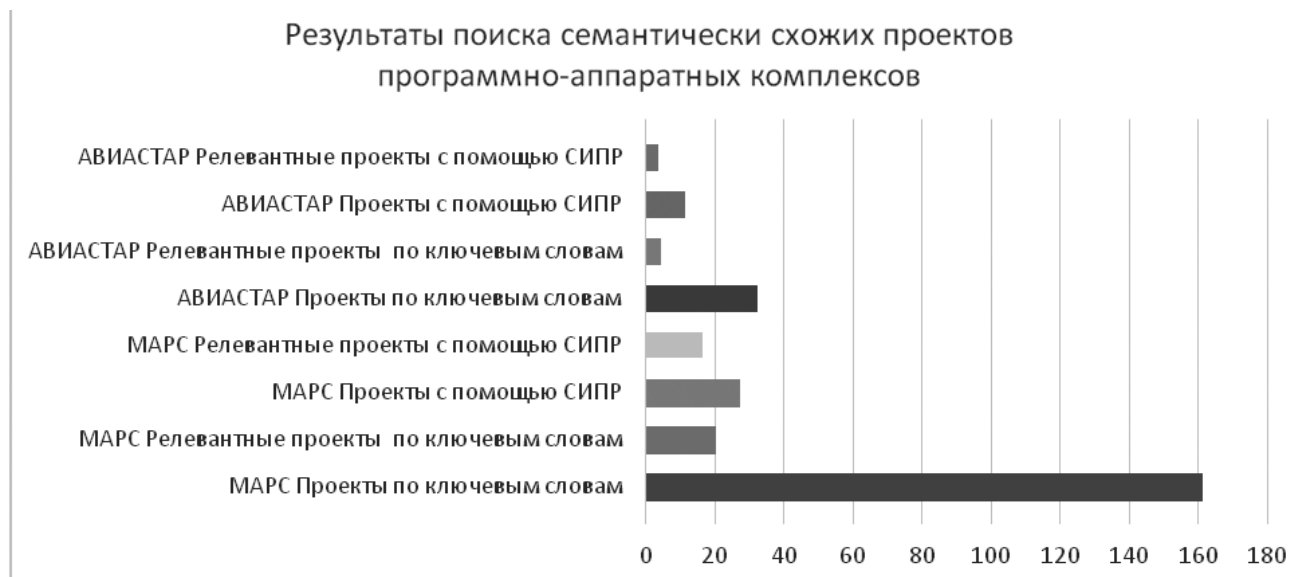


Рис. 3. Результаты экспериментов по использованию СИПР

где кортеж  $\langle elem_k^{dc_i}, cls_k^{tmp_j} \rangle$  означает, что экземпляр понятия проектной диаграммы  $elem_k^{dc_i}$  эквивалентен экземпляру класса проектного шаблона  $cls_k^{tmp_j}$ .

**Шаг 3.** Вычисление количества истинных фактов при условии выполнения попарной замены экземпляров классов  $j$ -го шаблона  $tmp_j$  и  $i$ -й проектной диаграммы  $dc_i$ :

$$dc_i: \forall k : cls_k^{tmp_j} \leftrightarrow elem_k^{dc_i}.$$

Указанные шаги алгоритма концептуального индексирования проектной диаграммы  $dc_i$  выполняются для каждого шаблона проектирования, доступного в онтологии проектных диаграмм. В результате, онтологическое представление проектной диаграммы электронного архива имеет следующий вид:

$$oV_{dc_i} = \{ \mu_{tmp_1}(dc_i)/tmp_1, \mu_{tmp_2}(dc_i)/tmp_2, \dots, \dots, \mu_{tmp_s}(dc_i)/tmp_s \}. \quad (2)$$

Фактически выражение (2) представляет собой нечеткое множество, построенное на множестве шаблонов проектирования онтологии проектных диаграмм, где  $\mu_{tmp_i}(dc_i)$  есть степень принадлежности проектной диаграммы  $dc_i$  шаблону проектирования  $tmp_j$ .

### 3 ВЫЧИСЛИТЕЛЬНЫЕ ЭКСПЕРИМЕНТЫ

Выполняемые в ходе исследования вычислительные эксперименты заключались в сравнении времени, затрачиваемого на анализ проектов, с помощью только ключевых слов и с помощью применения системы интеллектуального проектирования (СИПР) к результатам поиска по ключевым словам.

Рассматриваемые проекты были загружены из открытого репозитория *github.com*. Соответствие проектов, отобранных для анализа с помощью традиционного и предлагаемого подходов, устанавливалось с помощью экспертов. Задача отбора и анализа проектов на стадии разработки нового программно-аппаратного комплекса является нетривиальной и творческой. Как правило, оценки качества решения подобных задач даются экспертами, в качестве которых выступали специалисты: разработчики, проектировщики и менеджеры проектов, входящие в состав научной группы кафедры «Информационные системы» УлГТУ, и специалисты предприятий ФНПЦ АО «НПО «Марс» и АО «Авиастар-СП». Результаты экспериментов приведены на рисунке 3.

Исходя из полученных результатов можно сделать вывод о существенном сокращении объема проектных работ и, как следствие, временных затрат на этапах анализа существующих решений и проектирования нового программно-аппаратного комплекса.

Для проекта, разрабатываемого совместно с ФНПЦ АО «НПО «Марс», количество проектов сократилось в 5,96 раза, а для проекта, разрабатываемого совместно

с АО «Авиастар-СП», – в 2,9 раза. Качество работы системы определяется количеством релевантных проектов, входящих в выборку.

Для проекта, разрабатываемого совместно с ФНПЦ АО «НПО «Марс», качество выборки по ключевым словам составило 12,4 %, а с помощью СИПР – 59,2 %, что привело к повышению точности в 4,77 раза.

Для проекта, разрабатываемого совместно с АО «Авиастар-СП», качество выборки по ключевым словам составило 12,5 %, а с помощью СИПР – 27 %, что привело к повышению точности в 2,18 раза.

Полученные результаты вычислительных экспериментов позволяют сделать вывод о том, что подсистему поддержки интеллектуального проектирования СИПР можно рекомендовать к применению в случае большого размера выборки с относительно малым количеством релевантных проектов.

### ЗАКЛЮЧЕНИЕ

В работе представлена формальная модель онтологии проектных диаграмм, позволяющая на семантическом уровне описывать нотации диаграмм и используемые в организации шаблоны проектирования. Разработанный метод концептуального индексирования слабоформализованных проектных диаграмм на языке UML позволяет свести задачу анализа информационных ресурсов к операциям на гиперграфах. Выполнены вычислительные эксперименты на основе разработанных интеллектуальных программных систем структуризации содержимого электронных архивов крупных проектных организаций.

### СПИСОК ЛИТЕРАТУРЫ

1. Wongthongtham P., Pakdeetrakulwong U., Marzooq S. Ontology annotation for software engineering project management in multisite distributed software development environments. Cham : Springer International Publishing, 2017. – pp. 315–343.
2. Emdad A. Use of ontologies in software engineering // SEDE (Hisham Al-Mubaid and Rym Zalila-Wenkstern, eds.), ISCA. 2008. – pp. 145–150.
3. Dillon T., Chang E., Wongthongtham P. Ontology-based software engineering. – software engineering 2.0. // Australian Software Engineering Conference, IEEE Computer Society. 2008. – pp. 13–23.
4. Mark Grand. Java enterprise design patterns: Patterns in java, John Wiley&Sons, 2002.
5. Guskov G., Namestnikov A. Ontological mapping for conceptual models of software system // Seventh Conference, OSTIS 2017, Minsk, Belarus, February 18–20, 2017, Proceedings, 2017. – pp. 111–117.
6. Namestnikov A., Filippov A., Avvakumova V. An ontology based model of technical documentation fuzzy structuring // CEUR Workshop Proceedings. SCAKD 2016, Moscow, Russian Federation. 2016. – Vol. 1687. – pp. 63–74.
7. Гуськов Г.Ю., Наместников А.М. Программная система преобразования UML-диаграмм в онтологии на

языке OWL // Пятнадцатая национальная конференция по искусственному интеллекту с международным участием КИИ-2016 (3–7 октября 2016 г., г. Смоленск, Россия) : тр. конф. – Смоленск, 2016. – Т. 3. – С. 270–278.

8. Гуськов Г.Ю., Наместников А.М., Ярушкина Н.Г. Подход к поиску похожих по структуре проектов, основанный на онтологии языка UML // Радиотехника. – 2017. – № 6. – С. 122–127.

9. Наместников А.М., Субхангулов Р.А. Формирование информационных запросов к электронному архиву на основе концептуального индекса // Радиотехника. – 2014. – № 7. – С. 126–129.

## REFERENCES

1. Wongthongtham P., Pakdeetrakulwong U., Marzooq S. *Ontology Annotation for Software Engineering Project Management in Multisite Distributed Software Development Environments*. Cham. Springer International Publishing, 2017. pp. 315–343.

2. Emdad A. Use of Ontologies in Software Engineering. *SEDE (Hisham Al-Mubaid and Rym Zalila-Wenkstern, eds.)*, ISCA. 2008. pp. 145–150.

3. Dillon T., Chang E., Wongthongtham P. Ontology-Based Software Engineering- Software Engineering 2.0. *Australian Software Engineering Conference, IEEE Computer Society*, 2008. pp. 13–23.

4. Mark Grand. *Java Enterprise Design Patterns: Patterns in Java*. John Wiley&Sons, 2002.

5. Guskov G., Namestnikov A. Ontological Mapping for Conceptual Models of Software System. *Proceedings of the Seventh Conference, OSTIS 2017*. Minsk, Belarus, February 18–20, 2017, pp. 111–117.

6. Namestnikov A., Filippov A., Avvakumova V. An Ontology Based Model of Technical Documentation Fuzzy Structuring. *CEUR Workshop Proceedings. SCAKD 2016*. Moscow, Russian Federation, 2016, vol. 1687, pp. 63–74.

7. Guskov G.lu., Namestnikov A.M. Programmnaia sistema preobrazovaniia UML-diagramm v ontologii na iazyke OWL [Software System of UML-Diagrams Conversion into Ontologies in the OWL Language]. *15-aia natsionalnaia konferentsiia po iskusstvennomu itellektu s mezhdunar. uchastiem KII-2016 (3–7 oktiabria 2016)*. Tr. konf. [Proc. of the 15th National Conf. on Artificial Intelligence with Intern. Participation AIC-2016. 3-7 October]. Smolensk, 2016, vol. 3, pp. 270–278.

8. Guskov G.lu., Namestnikov A.M., Yarushkina N.G. Podkhod k poisku pokhzhikh po strukture proektov, osnovannyi na ontologii iazyka UML [Approach to the Search for Similar Software Projects Based on the UML Ontology]. *Radiotekhnika* [Journal Radioengineering], 2017, no. 6, pp. 122–127.

9. Namestnikov A.M., Subkhangulov R.A. Formirovanie informatsionnykh zaprosov k elektronnomu arkhivu na osnove kontseptualnogo indeksa [Formation of Queries to Electronic Archive on the Basis of the Conceptual Index]. *Radiotekhnika* [Journal Radioengineering], 2014, no. 7, pp. 126–129.