Math-Net.Ru

Общероссийский математический портал

А. А. Филиппов, В. С. Мошкин, Г. Ю. Гуськов, Н. Г. Ярушкина, Применение нечеткой базы знаний проблемной области в задаче поиска архитектурно подобных программных проектов, *Нечеткие системы и мягкие вычисления*, 2017, том 12, выпуск 2, 107–120

DOI: <https://doi.org/10.26456/fssc27>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением

<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 54.81.137.203

26 апреля 2023 г., 12:20:10

ПРИМЕНЕНИЕ НЕЧЕТКОЙ БАЗЫ ЗНАНИЙ ПРОБЛЕМНОЙ ОБЛАСТИ В ЗАДАЧЕ ПОИСКА АРХИТЕКТУРНО ПОДОБНЫХ ПРОГРАММНЫХ ПРОЕКТОВ

Филиппов А.А., Мошкин В.С., Гуськов Г.Ю., Ярушкина Н.Г.
Ульяновский государственный технический университет, г. Ульяновск

Поступила 08.10.2017, после переработки 14.12.2017.

В работе описывается процесс разработки нечеткой базы знаний, содержимое которой формируется в результате анализа контекстов проблемной области. Для хранения содержимого базы знаний в виде прикладной онтологии используется графовая база данных. Под контекстом понимается некоторая интерпретация предметной области и ее особенности. В статье описывается разработанный подход к генерации интерфейсов, который позволяет упростить работу пользователя с базой знаний. Представлена попытка реализовать механизм логического вывода по содержимому графовой базы данных. Приведен пример применения онтологии, построенной на нечеткой базе знаний в задаче поиска архитектурно схожих решений.

Ключевые слова: база знаний, проблемная область, прикладная онтология, логический вывод, контекст.

Нечеткие системы и мягкие вычисления. 2017. Т. 12, № 2. С. 107–120.
<https://doi.org/10.26456/fssc27>

Введение

Современное постиндустриальное общество оперирует крупными массивами информации, как в профессиональной деятельности, так и в повседневной. Принятие оптимального решения при таком объеме входной информации является сложной задачей [1, 2]. Для решения задачи обработки больших объемов данных применяются разнообразные программные средства автоматизации деятельности человека. Однако, для эффективной работы таких средств необходима их адаптация к особенностям конкретной проблемной области и вариантам ее интерпретации [3–9]. Программы, позволяющие автоматизировать принятие решений, требуют настройки, будь то время работы эксперта по уточнению коэффициентов или составление обучающей выборки для нейронной сети.

В профессиональной деятельности очень остро стоит вопрос сохранения и передачи знаний между сотрудниками. Предприятия с долгой историей и большим количеством завершенных проектов в разных предметных областях нуждаются в инструменте, который бы позволил отбирать релевантные реализованные проекты на стадии проектирования нового программного проекта. Коллектив сотрудников, обладающий списком уже реализованных проектов, схожих по структуре

и предметной области с новым проектом, выполнит задачи по созданию архитектуры гораздо быстрее и избежит многих проблем, с которыми сталкивались его предшественники.

В рамках данной работы предпринята попытка построения нечеткой базы знаний (БЗ), содержимое которой представлено в виде прикладной онтологии. Основные требования к рассматриваемой базе знаний (Рис. 1):

- адаптация к особенностям предметной области с учетом контекстов;
- надежность и быстроедействие хранилища онтологии;
- наличие механизма логического вывода;
- наличие средств упрощения работы с базой знаний для неподготовленных пользователей;
- наличие механизмов импорта данных из внешних информационных ресурсов;
- наличие механизмов извлечения знаний из открытых репозиторий, например, github;
- наличие механизмов извлечения знаний из uml-диаграмм, сохраненных в виде .xmi файлов;
- наличие онтологии, позволяющей сохранять архитектуру программных продуктов;
- наличие модуля поиска среди программных продуктов со схожей структурой в релевантной предметной области.

Как видно из Рис. 1, БЗ состоит из следующих подсистем:

1. Хранилище онтологии: Neo4j, модуль для управления содержимым БЗ, модуль для импорта онтологий в формате RDF или OWL.
2. Подсистема логического вывода: модуль для организации логического вывода, модуль для динамического построения экранных форм пользовательского интерфейса.
3. Подсистема взаимодействия с пользователями: модуль для динамического построения экранных форм пользовательского интерфейса.
4. Подсистема для импорта данных из внешних информационных ресурсов: модуль для импорта данных из внешних wiki-ресурсов, модуль для наполнения внешних wiki-ресурсов.

1. Организация хранилища онтологий нечеткой БЗ

Онтология — модель представления предметной области в виде семантической паутины (графа) [10,11]. В качестве основы хранилища онтологий рассматриваемой БЗ используется графо-ориентированная система управления базами данных (ГСУБД) Neo4j. Данная ГСУБД в настоящее время является одной из самых популярных графовых баз данных и обладает следующими преимуществами:

Рис. 1: Архитектурная схема нечеткой базы знаний

1. Наличие свободной и бесплатной community версии.
2. Нативный формат для хранения данных.
3. Один экземпляр Neo4j может работать с графами, содержащими миллиарды узлов и отношений.
4. Наличие графо-ориентированного языка запросов Cypher.
5. Поддержка транзакций.

Также ГСУБД Neo4j была выбрана для хранения описания Про в виде прикладной онтологии, так как онтология фактически является графом. При этом необходимо лишь ограничить набор узлов и отношений графа, в которые будут транслироваться онтологии, описанные с помощью языков RDF и OWL.

Для определения сущностей, в которые будут транслироваться онтологии в процессе импорта, было решено использовать результат операции пересечения сущностей языков RDF и OWL.

Контекст онтологии — некоторое состояние онтологии, полученное в процессе версионирования, либо формирования онтологии с использованием различных точек зрения.

Формально онтологию нечеткой базы знаний можно представить в виде следующего выражения:

$$O = \langle T, C^{T_i}, I^{T_i}, P^{T_i}, S^{T_i}, F^{T_i}, R^{T_i} \rangle, i = \overline{1-t}, \quad (1)$$

где

- t — количество контекстов онтологии нечеткой базы знаний;
- $T = \{T_1, T_2, \dots, T_t\}$ — множество контекстов онтологии нечеткой базы знаний;
- $C^{T_i} = \{C_1^{T_i}, C_2^{T_i}, \dots, C_n^{T_i}\}$ — множество классов онтологии в рамках i -ого контекста;
- $I^{T_i} = \{I_1^{T_i}, I_2^{T_i}, \dots, I_n^{T_i}\}$ — множество объектов онтологии в рамках i -ого контекста;
- $P^{T_i} = \{P_1^{T_i}, P_2^{T_i}, \dots, P_n^{T_i}\}$ — множество свойств онтологии в рамках i -ого контекста;
- $S^{T_i} = \{S_1^{T_i}, S_2^{T_i}, \dots, S_n^{T_i}\}$ — множество состояний онтологии в рамках i -ого контекста;
- $F^{T_i} = \{F_1^{T_i}, F_2^{T_i}, \dots, F_n^{T_i}\}$ — множество логических правил онтологии в рамках i -ого контекста;
- R^{T_i} — множество отношений онтологии в рамках i -ого контекста следующего вида:

$$R^{T_i} = \langle R_C^{T_i}, R_I^{T_i}, R_P^{T_i}, R_S^{T_i}, R_F^{T_i} \rangle, \quad (2)$$

где

- $R_C^{T_i}$ — множество отношений, формирующих иерархию классов онтологии в рамках i -ого контекста;
- $R_I^{T_i}$ — множество отношений, определяющих связь вида «класс-объект» онтологии в рамках i -ого контекста;
- $R_P^{T_i}$ — множество отношений, определяющих связь вида «класс-свойство» онтологии в рамках i -ого контекста;
- $R_S^{T_i}$ — множество отношений, определяющих связь вида «объект-состояние» онтологии в рамках i -ого контекста;
- $R_F^{T_i}$ — множество отношений, сгенерированных на основе логических правил онтологии в рамках i -ого контекста.

В основе онтологии нечеткой БЗ лежат принципы, схожие с парадигмой объектно-ориентированного программирования:

- классы онтологии представляют собой концепты предметной области;

- классы могут иметь свойства, класс-потомок наследует свойства класса-родителя;
- объекты онтологии описывают экземпляры концептов предметной области;
- конкретные значения для свойств объектов, унаследованных от родительского класса, определяются состояниями;
- логические правила используются для реализации функций логического вывода по содержимому нечеткой базы знаний.

2. Организация логического вывода по содержимому нечеткой базы знаний

Логический вывод — это автоматизированный процесс получения новых фактов, исходя из наборов фактов и правил, заданных ранее. Так же с помощью автоматизированного логического вывода осуществляется автоматический вывод большого количества математических теорем. Для осуществления функции логического вывода используется специализированный класс программных средств — ризонеры (reasoner). Ризонеры формируют логические следствия на основе множества утверждений, фактов и аксиом [12, 13].

В настоящий момент существует несколько различных ризонеров, но ни один из них не поддерживает работу с ГСУБД Neo4j. Таким образом, возникает необходимость в разработке механизма логического вывода по содержимому нечеткой БЗ.

Наиболее распространенным языком для записи логических правил является SWRL. Формально логическое правило онтологии нечеткой БЗ может быть представлено в следующем виде:

$$F^{T_i} = \langle A_C^{Tree}, A_I^{SWRL}, A^{Cypher} \rangle, \quad (3)$$

где T_i — i -й контекст онтологии нечеткой базы знаний; A^{Tree} — SWRL-представление логического правила F^{T_i} ; A^{Cypher} — Cypher-представление логического правила F^{T_i} .

Древовидное представление A^{Tree} логического правила F^{T_i} имеет вид:

$$A^{Tree} = \langle Ant, Cons \rangle, \quad (4)$$

где $Ant = Ant_1 \Theta Ant_2 \Theta \dots Ant_n$ — антецедент (условие) логического правила F^{T_i} ; $\Theta \in \{AND, OR\}$ — множество допустимых логических операций между атомами антецедента; $Cons$ — консеквент (следствие) логического правила F^{T_i} .

3. Применение нечеткой базы знаний для поиска архитектурно подобных проектов из одной предметной области

Для решения прикладной задачи необходимо было составить онтологию, позволяющую хранить информацию о программных продуктах [14]. Онтология, хранящая информацию о программных продуктах, должна быть построена на принципах архитектуры программного обеспечения.

3.1 Структура онтологии программного обеспечения

Как правило, архитектура программного обеспечения выражается с помощью UML-диаграмм. UML-диаграммы строятся на основе хорошо документированной и описанной иерархической мета-модели. Мета-модель языка UML легла в основу онтологии программных продуктов. Граф онтологии представлен на Рис. 2.

Рис. 2: Структура онтологии, основанной на мета-схеме языка UML

3.2 Шаблоны проектирования в онтологии программного обеспечения

Мета-модель языка UML хорошо подходит для описания базовых конструкторов онтологии программного обеспечения, но слабо передает семантику архитектуры программного обеспечения. Следует разделять семантику предметной области и процессов, отраженную в программном обеспечении, и семантику архитектуры программного обеспечения. Для выражения семантики архитектуры программного обеспечения следует использовать шаблоны проектирования и учитывать степень их выраженности в программных проектах.

Удобно переносить шаблоны проектирования в онтологию из UML-диаграмм. UML-диаграмма классов шаблона проектирования представлена на Рис. 3.

Рис. 3: UML-диаграмма классов шаблона проектирования Builder

3.3 Алгоритм поиска архитектурно подобных проектов

Основной сценарий использования онтологии программного обеспечения:

1. Составление UML-диаграмм нового программного обеспечения.
2. Выделение кластера проектов из предметной области нового проекта.
3. Сохранение архитектурных решений в формате онтологии программного обеспечения.
4. Расчет степени выраженности шаблонов проектирования в проектах, как новом, так и уже реализованных.
5. Расчет степени архитектурной подобности, опираясь на меру выраженности.

Мера выраженности шаблона проектирования в проекте может быть рассчитана по следующей формуле:

$$\mu_{dc_{\gamma}, dc_{\delta}} = \bigvee_{tmp \in (dc_{\gamma} \cap dc_{\delta})} \mu_{dc_{\gamma} \cap dc_{\delta}}(tmp), \quad (5)$$

где dc_{γ} и dc_{δ} — UML-диаграммы классов, представленные как множество утверждений онтологии программного обеспечения, $\mu_{dc_{\gamma}, dc_{\delta}}(tmp)$ — мера выраженности шаблона проектирования в диаграмме классов программного проекта.

Приведенная формула (5) не является абсолютно справедливой для всех наборов проектов и смещает вес на шаблон проектирования, максимально выраженный в обоих программных продуктах.

Пример описания шаблона проектирования Builder в онтологии программного обеспечения представлен на Рис. 4.

Рис. 4: Шаблон проектирования Builder, сохраненный в виде утверждений онтологии программного обеспечения

4. Построение графического интерфейса пользователя на основе содержимого нечеткой базы знаний

Для упрощения работы с базой знаний неподготовленных пользователей и реализации функции контроля введенных пользователем данных используется механизм динамического построения графического интерфейса пользователя [15].

Для решения задачи построения графического интерфейса пользователя на основе содержимого нечеткой базы знаний необходимо сопоставить сущности онтологии нечеткой базы знаний с элементами построения графического интерфейса пользователя.

Формальную модель ГИП можно представить в виде следующего выражения:

$$UI = \langle L, C, I, P, S \rangle, \quad (6)$$

где

- $L = L_1, L_2, \dots, L_n$ — множество графических компонентов графического интерфейса пользователя (например, ListBox, TextBox, ComboBox и т. п.);
- $C = C_1, C_2, \dots, C_n$ — множество классов онтологии нечеткой базы знаний;
- $I = I_1, I_2, \dots, I_n$ — множество объектов онтологии нечеткой базы знаний;
- $P = P_1, P_2, \dots, P_n$ — множество свойств классов онтологии нечеткой базы знаний;
- $S = S_1, S_2, \dots, S_n$ — множество состояний объектов онтологии нечеткой базы знаний.

Для построения ГИП на основе содержимого нечеткой БЗ используется следующая функция:

$$F(O) = \{C^{T_i}, I^{T_i}, P^{T_i}, S^{T_i}, F^{T_i}, R^{T_i}\} \rightarrow \{L, C, I, P, S\}, \quad (7)$$

где $\{C^{T_i}, I^{T_i}, P^{T_i}, S^{T_i}, F^{T_i}, R^{T_i}\}$ — множество сущностей онтологии нечеткой базы знаний, представленной выражением (1), в рамках i -го контекста; $\{L, C, I, P, S\}$ — множество сущностей графического интерфейса пользователя нечеткой базы знаний, представленного выражением (7).

Таким образом, содержимое нечеткой базы знаний отображается на множество компонентов графического пользовательского интерфейса, что позволяет упростить работу с базой знаний для пользователя, не обладающего навыками в области онтологического анализа и инженерии знаний, а также контролировать логическую целостность вводимых пользователем данных, что приводит к снижению количества потенциальных ошибок ввода.

Заключение

Таким образом, применение нечетких баз знаний, хранимых в ГСУБД, в процессе поддержки принятия решений предполагает наличие определенного набора механизмов. Представленные механизмы позволяют максимально автоматизировать процесс обучения взаимодействия с базой знаний и упростить работу специалистов:

- организация логического вывода по содержимому нечеткой базы знаний посредством трансляции SWRL-правил в Cypher-структуры;
- построение графического интерфейса пользователя на основе содержимого нечеткой базы знаний;
- автоматизированный импорт знаний из внутрикорпоративных и внешних wiki-ресурсов.

Применение контекстного подхода к хранению знаний повышает эффективность использования предметных онтологий, позволяя адаптировать базы знаний не только под особенности описываемой проблемной области, но и под требования специалистов, предоставляя им удобный в использовании программный инструментарий, динамически изменяемый в зависимости от содержимого базы знаний. Полученное решение может быть включено в систему онтологий для решения сложных задач, как было продемонстрировано на примере программных продуктов.

Список литературы

- [1] Бова В.В., Курейчик В.В., Нужнов Е.В. Проблемы представления знаний в интегрированных системах поддержки управленческих решений // Известия ЮФУ. Технические науки. 2010. № 7(108). С. 107–113.
- [2] Черняховская Л.Р., Федорова Н.И., Низамутдинова Р.И. Интеллектуальная поддержка принятия решений в оперативном управлении деловыми процессами предприятия // Вестник Уфимского государственного авиационного технического университета. 2011. Т. 15, № 2(42). С. 172–176.
- [3] Вагин В.Н., Михайлов И.С. Разработка метода интеграции информационных систем на основе метамоделирования и онтологии предметной области // Программные продукты и системы. 2008. № 1. С. 22–26.
- [4] Гаврилова Т.А. Онтологический подход к управлению знаниями при разработке корпоративных информационных систем // Новости искусственного интеллекта. 2003. № 2. С. 24–30.
- [5] Загоруйко Ю.А. Построение порталов научных знаний на основе онтологии // Вычислительные технологии. 2007. Т. 12, № S2. С. 169–177.
- [6] Карабач А.Е. Системы интеграции информации на основе семантических технологий // Наука, техника и образование. 2014. № 2(2). С. 58–62.
- [7] Смирнов С.В. Онтологическое моделирование в ситуационном управлении // Онтология проектирования. 2012. № 2. С. 16–24.
- [8] Тузовский А.Ф. Разработка систем управления знаниями на основе единой онтологической базы знаний // Известия Томского политехнического университета. Инжиниринг георесурсов. 2007. Т. 310, № 2. С. 182–185.
- [9] Филиппов А.А., Мошкин В.С., Шалаев Д.О., Ярушкина Н.Г. Единая онтологическая платформа интеллектуального анализа данных // Материалы шестой международной научно-технической конференции «OSTIS-2016». Минск, Республика Беларусь, 2016. С. 77–82.
- [10] Мошкин В.С., Ярушкина Н.Г. Методики построения нечетких онтологий сложных предметных областей // Материалы пятой международной научно-технической конференции «OSTIS-2015». Минск, 2015. С. 401–406.
- [11] Тарасов В.Б., Калущая А.П., Святкина М.Н. Гранулярные, нечеткие и лингвистические онтологии для обеспечения взаимопонимания между когнитивными агентами // Материалы второй международной научно-технической конференции «OSTIS-2012». Минск, 2012. С. 267–278.
- [12] Махортов С.Д. Об алгебраической модели распределенной продукционной системы // Материалы пятнадцатой национальной конференции по ИИ с международным участием «КИИ-2016». Т. 1. Смоленск, 2016. С. 64–72.
- [13] Мошкин В.С., Ярушкина Н.Г. Логический вывод на основе нечетких онтологий // Интегрированные модели и мягкие вычисления в искусственном интеллекте: сборник научных трудов VIII-й Международной научно-практической конференции. Т. 1. М.: Физматлит, 2015. С. 259–267.

- [14] Гуськов Г.Ю., Наместников А.М. Ontological mapping for conceptual models of software system // Материалы шестой международной научно-технической конференции «OSTIS-2017». Минск, Республика Беларусь, 2017. С. 111–116.
- [15] Грибова В.В., Клещев А.С. Управление проектированием и реализацией пользовательского интерфейса на основе онтологий // Проблемы управления. 2006. № 2. С. 58–62.

Образец цитирования

Филиппов А.А., Мошкин В.С., Гуськов Г.Ю., Ярушкина Н.Г. Применение нечеткой базы знаний проблемной области в задаче поиска архитектурно подобранных программных проектов // Нечеткие системы и мягкие вычисления. 2017. Т. 12, № 2. С. 107–120. <https://doi.org/10.26456/fssc27>

Сведения об авторах

1. Филиппов Алексей Александрович

доцент кафедры «Информационные системы» Ульяновского государственного технического университета.

Россия, 432000, г. Ульяновск, ул. Радищева, д. 102, УлГТУ.

2. Мошкин Вадим Сергеевич

доцент кафедры «Информационные системы» Ульяновского государственного технического университета.

Россия, 432000, г. Ульяновск, ул. Радищева, д. 102, УлГТУ.

3. Гуськов Глеб Юрьевич

аспирант кафедры «Информационные системы» Ульяновского государственного технического университета.

Россия, 432000, г. Ульяновск, ул. Радищева, д. 102, УлГТУ.

4. Ярушкина Надежда Глебовна

зав. кафедрой «Информационные системы» Ульяновского государственного технического университета.

Россия, 432000, г. Ульяновск, ул. Радищева, д. 102, УлГТУ.

E-mail: jng@ulstu.ru

APPLICATION OF FUZZY KNOWLEDGE BASE OF THE PROBLEM AREA IN THE PROBLEM OF SEARCHING FOR ARCHITECTURALLY SIMILAR SOFTWARE PROJECTS

Filippov Alexey Alexandrovich

Associate professor at Information Systems department,
Ulyanovsk State Technical University
Russia, 432000, Ulyanovsk, Radisheva st., 102, UlSTU.

Moshkin Vadim Sergeyevich

Associate professor at Information Systems department,
Ulyanovsk State Technical University
Russia, 432000, Ulyanovsk, Radisheva st., 102, UlSTU.

Guskov Gleb Yuryevich

PhD student at Information Systems department,
Ulyanovsk State Technical University
Russia, 432000, Ulyanovsk, Radisheva st., 102, UlSTU.

Yarushkina Nadezhda Glebovna

Head of Information Systems department,
Ulyanovsk State Technical University
Russia, 432000, Ulyanovsk, Radisheva st., 102, UlSTU. E-mail: jng@ulstu.ru

Received 08.12.2017, revised 14.12.2017.

The paper describes the process of developing a fuzzy knowledge base (KB), the content of which is formed as a result of the analysis of the subject area (SA) context. To store the content of the KB in the form of an applied ontology graph database (DB) is used. Under the context is understood some interpretation of the SA and its features. The paper describes the developed approach to interface generation, which makes it possible to simplify the user's work with the KB. An attempt were made to implement the inference engine by contents of a graph database. An example of the application of ontology based on fuzzy knowledge base in the problem of searching architecturally similar solutions is given.

Keywords: knowledge base, subject area, applied ontology, inference engine, context.

Nechetkie Sistemy i Myagkie Vychisleniya [Fuzzy Systems and Soft Computing], 2017, vol. 12(2), pp. 107–120. <https://doi.org/10.26456/fssc27>

References

- [1] Bova V.V., Kureichik V.V., Nuzhnov E.V. Problems of representation of knowledge in integrated systems of support of management decisions. *Izvestiya YuFU. Tekhnicheskie Nauki* [Herald of SFU. Technical Sciences], 2010, no. 7(108), pp. 107–113. (in Russian)
- [2] Chernyakhovskaya L.R., Fedorova N.I., Nizamutdinova R.I. Intellectual support of decision-making in the operational management of business processes of the enterprise. *Vestnik Ufmskogo Gosudarstvennogo Aviatsionnogo Tekhnicheskogo Universiteta* [Bulletin of the Ufa State Aviation Technical University], 2011, vol. 15, no. 2(42), pp. 172–176. (in Russian)
- [3] Vagin V.N., Mikhailov I.S. Development of the method of integration of information systems based on metamodeling and ontology of the subject domain. *Programmnye Produkty i Sistemy* [Software Products and Systems], 2008, no. 1, pp. 22–26. (in Russian)
- [4] Gavrilova T.A. Ontological approach to knowledge management in the development of corporate information systems. *Novosti Iskusstvennogo Intellekta* [News of Artificial Intelligence], 2003, no. 2, pp. 24–30. (in Russian)
- [5] Zagorul'ko Yu.A. Construction of portals of scientific knowledge on the basis of ontology. *Vychislitel'nye Tekhnologii* [Computational Technologies], 2007, vol. 12(S2), pp. 169–177. (in Russian)
- [6] Karabach A.E. Information integration systems based on semantic technologies. *Nauka, Tekhnika i Obrazovanie* [Science, Technology and Education], 2014, no. 2(2), pp. 58–62. (in Russian)
- [7] Smirnov S.V. Ontological modeling in situational management. *Ontologiya Proektirovaniya* [Ontology of Design], 2012, no. 2, pp. 16–24. (in Russian)
- [8] Tuzovskii A.F. Development of knowledge management systems based on a single ontological knowledge base. *Izvestiya Tomskogo Politehnicheskogo Universiteta. Inzhiniring Georesursov* [Bulletin of the Tomsk Polytechnic University. Georesources Engineering], 2007, vol. 310(2), pp. 182–185. (in Russian)
- [9] Filippov A.A., Moshkin V.S., Shalaev D.O., Yarushkina N.G. Single ontological platform for data mining. *Materialy Shestoi Mezhdunarodnoi Nauchno-Tekhnicheskoi Konferentsii «OSTIS-2016»* [Proceedings of the Sixth International Scientific and Technical Conference «OSTIS-2016»]. Minsk, Republic of Belarus, 2016. Pp. 77–82. (in Russian)
- [10] Moshkin V.S., Yarushkina N.G. Methods of constructing fuzzy ontologies of complex subject domains. *Materialy Pyatoi Mezhdunarodnoi Nauchno-Tekhnicheskoi Konferentsii «OSTIS-2015»* [Proceedings of the Fifth International Scientific and Technical Conference «OSTIS-2015»]. Minsk, 2015. Pp. 401–406. (in Russian)
- [11] Tarasov V.B., Kalutskaya A.P., Svyatkina M.N. Granular, fuzzy and linguistic ontologies for providing mutual understanding between cognitive agents. *Materialy Vtoroi Mezhdunarodnoi Nauchno-Tekhnicheskoi Konferentsii «OSTIS-2012»* [Proceedings of the Second International Scientific and Technical Conference «OSTIS-2012»]. Minsk, 2012. Pp. 267–278. (in Russian)

- [12] Makhortov S.D. On the Algebraic Model of a Distributed Production System. *Materialy Pyatnadtsatoi Natsional'noi Konferentsii po AI s Mezhdunarodnym Uchastiem «KII-2016»* [Proceedings of the Fifteenth National Conference on AI with International Participation «KII-2016»]. Vol. 1. Smolensk, 2016. Pp. 64–72. (in Russian)
- [13] Moshkin V.S., Yarushkina N.G. The logical conclusion based on fuzzy ontologies. *Integrirovannye Modeli i Myagkie Vychisleniya v Iskusstvennom Intellekte: Sbornik Nauchnykh Trudov VIII-i Mezhdunarodnoi Nauchno-Prakticheskoi Konferentsii* [Integrated Models and Soft Calculations in Artificial Intelligence: Proceedings of the VIII International Scientific and Practical Conference]. Vol. 1. Fizmatlit Publ., Moscow, 2015. Pp. 259–267. (in Russian)
- [14] Gus'kov G.Yu., Namestnikov A.M. Ontological mapping for conceptual models of software system. *Materialy Shestoi Mezhdunarodnoi Nauchno-Tekhnicheskoi Konferentsii «OSTIS-2017»* [Materials of the Sixth International Scientific and Technical Conference «OSTIS-2017»]. Minsk, Republic of Belarus, 2017. Pp. 111–116.
- [15] Gribova V.V., Kleshchev A.S. Managing the design and implementation of the user interface on the basis of ontologies. *Problemy Upravleniya* [Control Problems], 2006, no. 2, pp. 58–62. (in Russian)

Citation

Filippov A.A., Moshkin V.S., Guskov G.Yu., Yarushkina N.G. Application of fuzzy knowledge base of the problem area in the problem of searching for architecturally similar software projects. *Nechetkie Sistemy i Myagkie Vychisleniya* [Fuzzy Systems and Soft Computing], 2017, vol. 12(2), pp. 107–120. (in Russian). <https://doi.org/10.26456/fssc27>