

## Искусственный интеллект и машинное обучение

УДК 004.89

### Унифицированное применение методов онтологического инжиниринга в задачах обработки текстовых данных

К. В. Гимашева<sup>1; a</sup>, Е. А. Гладких<sup>1; b</sup>, С. И. Чуприна<sup>1; c</sup><sup>1</sup>Пермский государственный национальный исследовательский университет; Пермь, Россия<sup>a</sup>e-mail: gimashevaksusha1@gmail.com; ORCID: 0000-0002-8144-9656<sup>b</sup>e-mail: katya.myr.23@yandex.ru; ORCID: 0000-0002-2930-3289<sup>c</sup>e-mail: chuprinas@inbox.ru; ORCID: 0000-0002-2103-3771, AuthorID: 11124

Рассматриваются вопросы унифицированного применения методов и средств онтологического инжиниринга для автоматической обработки текстовых данных при решении двух классов задач. Первый класс задач связан с автоматизацией построения приложений реляционных баз данных на основе извлечения нужных фактов непосредственно из неструктурированных коллекций текстовых документов на естественном языке. Второй класс задач связан с автоматизацией обработки текстов на искусственных языках в процессе верификации программного кода приложений баз данных. Задачи проектирования и разработки информационных систем, так же, как и задачи верификации программного кода, представляют известную сложность для непрофессионалов в области программирования. Автоматизация решения этих задач актуальна и для профессионалов, так как способна значительно сократить общее время разработки приложений БД. Предложена концепция онтологически управляемого решения указанных задач, описан унифицированный подход к их решению и реализация демоверсий соответствующих инструментальных средств.

**Ключевые слова:** онтология; онтологически управляемая разработка информационных систем; автоматизация построения БД; обработка текстов на естественном языке; верификация исходного кода программ.

Поступила в редакцию 25.07.2022, принята к опубликованию 12.08.2022

## An Ontology-Driven Methods Unified Application in Text Data Processing Problems

K. V. Gimasheva<sup>1; a</sup>, E. A. Gladkikh<sup>1; b</sup>, S. I. Chuprina<sup>1; c</sup><sup>1</sup>Perm State University; Perm, Russia<sup>a</sup>e-mail: gimashevaksusha1@gmail.com; ORCID: 0000-0002-8144-9656<sup>b</sup>e-mail: katya.myr.23@yandex.ru; ORCID: 0000-0002-2930-3289<sup>c</sup>e-mail: chuprinas@inbox.ru; ORCID: 0000-0002-2103-3771, AuthorID: 11124

Unified application issues of ontological engineering methods and tools for text data processing automation in 2 problems classes solving are considered. The first class is the relational database applications construction automation based on the necessary facts extracting directly from unstructured text documents collections in natural language. The second class is the text data processing automation in artificial languages in the program code verifying process for database applications. The information systems designing and developing problems, same as program code verifying



Эта работа © 2022 Гимашева К.В., Гладких Е.А., Чуприна С.И. лицензируется под CC BY 4.0. Чтобы просмотреть копию этой лицензии, посетите <http://creativecommons.org/licenses/by/4.0/>

problems, are complex for non-professional programmers. The text data processing problem solving automation is also relevant for professional programmers, because it significantly reduces the database applications development time. The ontologically controlled solution concept is proposed for these problems. An unified approach for problems solution and the tools demos implementation are described.

**Keywords:** *Ontology; Ontology-Driven Information System Design; Automation of Database Construction; Natural Language Processing; Source Code Verification.*

*Received 25.07.2022, accepted 12.08.2022*

DOI: 10.17072/1993-0550-2022-3-49-58

## Введение

В эпоху всеобщей цифровизации актуальна разработка программного обеспечения (ПО), которое в автоматизированном режиме могло бы унифицированным образом обрабатывать тексты как на естественном, так и на искусственном языках. При этом важнейшим требованием к разработке такого программного инструментария является его адаптируемость в смысле независимости от специфики конкретной предметной области решаемой задачи для обеспечения автоматизированного учета этой специфики без внесения изменений в исходный код ПО.

Среди различных подходов к построению адаптируемых информационных систем (ИС) в настоящее время все большее предпочтение отдается модельно-ориентированному подходу (англ., MDA, Model Driven Architecture). MDA представляет собой концепцию построения архитектуры приложений на основе некоторой формальной модели, управляющей функционированием всех основных компонент приложения [1]. Понятие формальных моделей достаточно обширно, например, в качестве формальных моделей могут быть использованы UML (Unified Modeling Language), DSL (Domain-Specific Language), модели онтологий.

В нашем подходе к решению задач автоматизации построения приложений баз данных (БД) и верификации исходного текста приложений БД используется модель прикладной онтологии [2]. Благодаря методам онтологического инжиниринга, появляется возможность унифицированным образом решать задачи в разных предметных областях посредством изменения онтологии без внесения изменений в исходный код основных компонент разрабатываемого программного обеспечения, что является важным преимуществом в задачах автоматизации построения ИС.

Предлагаемый подход реализуется в рамках современной парадигмы семантического доступа и интеграции данных, называемой доступом к данным на основе онтологий (англ., OBDA – Ontology-Based Data Access [3, 4]). В парадигме OBDA онтология определяет глобальную схему высокого уровня (обычно уже существующих) источников данных и предоставляет словарь для формирования пользовательских запросов, в т.ч. запросов на естественном языке (ЕЯ). Затем OBDA-система использует онтологии и вспомогательные информационные ресурсы для трансформации ЕЯ-запросов в запросы к источникам данных, например созданным в среде систем управления реляционными базами данных.

Новизна нашего подхода заключается в том, что мы унифицированным образом используем принципы OBDA не для семантической интеграции разнородных данных, а для решения двух разных задач: задачи автоматического построения БД непосредственно из фактов, извлеченных из ЕЯ-текстов, и задачи верификации программного кода приложений баз данных, написанных на искусственных языках.

## Онтологически управляемое решение по автоматизации построения ИС

Предлагаемая концепция ориентирована на широкую категорию пользователей, необязательно являющихся специалистами в области информационных технологий (ИТ). Зачастую такие пользователи (частные предприниматели, самозанятые и др.) в процессе своей профессиональной деятельности накапливают большие коллекции неструктурированных текстовых документов (опросов, анкет, бланков, приказов, финансовых документов, отчетов и т.п.) как в печатном, так и в электронном виде.

Для автоматизации обработки, анализа и эффективного поиска данных требуется их нормализация и структуризация, а также ав-

томатизация ввода данных и наличие интерфейса для выполнения поисковых запросов, т.е., фактически, требуется построение ИС по типу АРМ (Автоматизированное Рабочее Место специалиста).

В 2019 г. на кафедре МОВС ПГНИУ был разработан демонстрационный прототип оболочки Magic Ontologies, автоматизирующий построение таких ИС с базой данных, автоматически пополняемой фактами из входных документов текстового формата. Демо-версия Magic Ontologies находится в свободном доступе по адресу <http://magic-ontologies.shgk.me/> (дата обращения: 30.05.2022). Однако она обладает целым рядом недостатков, в частности, в ней отсутствует возможность автоматического обновления уже существующей БД фактами, извлеченными из новых коллекций документов (требуется выполнить заново процесс извлечения, объединив прежнюю коллекцию документов с новой), нет проверки на дубликаты данных. Эти и другие недостатки устранены в настольной версии приложения MagNolis. Оба приложения – языконезависимые.

MagNolis является онтологически управляемым средством извлечения фактов из неструктурированных коллекций текстовых документов (особенности и проблемы такого извлечения хорошо описаны, например, в [5]), способным легко адаптироваться к специфике информационного содержания входных коллекций текстовых документов, а также унифицированным образом извлекать, хранить и обрабатывать факты. Система адаптируется к специфике предметной области за счет подаваемой ей на вход онтологии, которая строит-

ся в среде визуального редактора онтологий Онтолис [6] и описывает простейшую таксономию понятий, включающую наименования нужных фактов, значения которых необходимо извлечь из ЕЯ-текста.

Как показывает опыт практических занятий со студентами третьего курса разных факультетов и слушателями сетевого ИТ-университета, не являющихся квалифицированными ИТ-специалистами, создание таких простых онтологий, включающих вершины с именами нужных для извлечения фактов понятий и поддерживающих только иерархические связи типа "a\_part\_of" ("часть-целое"), не вызывает сложностей (см. рис. 1).

Для создания БД информационной системы с поддержкой запросов на ЕЯ от конечного пользователя требуется только создание указанной простейшей онтологии и открытие файлов исходных текстовых документов. После выбора соответствующего пункта меню происходит управляемое онтологией извлечение нужных данных из документов и автоматически создается новая либо пополняется уже существующая БД. Далее пользователю предоставляется возможность сделать запросы к БД на естественном языке (как в поисковых запросах в Интернет), но при этом в тексте запроса обязательно должны присутствовать понятия из онтологии либо их синонимы.

В текущей версии MagNolis пока еще не реализован автоматический учет синонимов в контексте и не решены некоторые другие проблемы (проблематика, связанная с автоматической обработкой ЕЯ-текстов и реализацией сервисов ЕЯ-запросов к реляционным БД, описана, например, в [7, 8]).

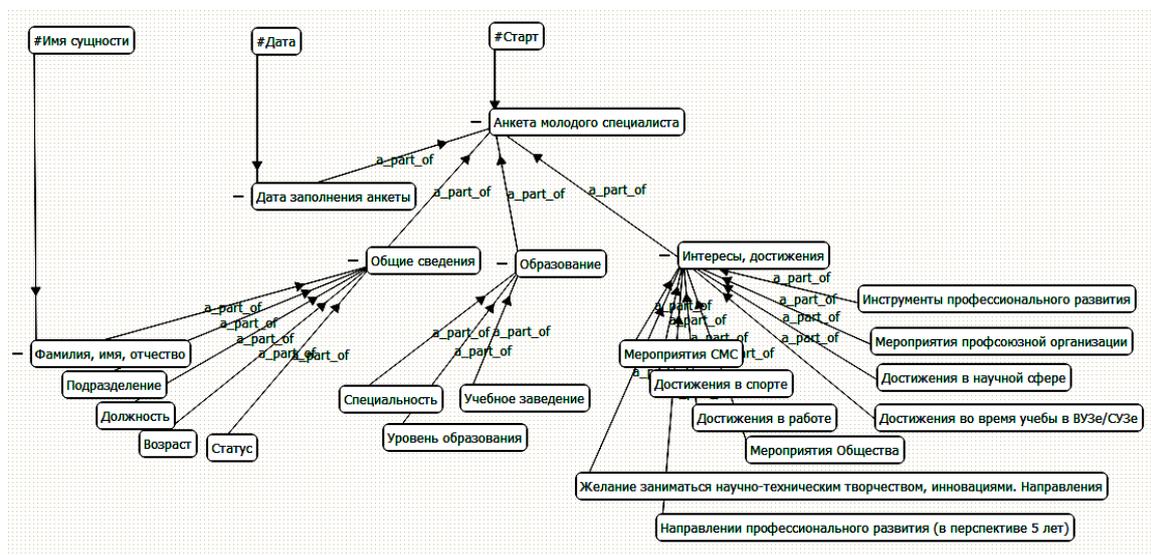


Рис. 1. Пример онтологии анкеты

Схема, представленная на рис. 2, демонстрирует основные этапы работы с MagNolis: загрузку онтологии предметной области и коллекции текстовых документов, содержащих различную информацию о некотором информационном объекте, атрибуты которого описаны в соответствующей онтоло-

гии; извлечение и предобработку текста документов; парсинг онтологии; онтологически управляемое извлечение фактов из текста; автоматическое создание и заполнение БД извлеченными фактами; сервис поиска с возможностью задания ЕЯ-запросов.

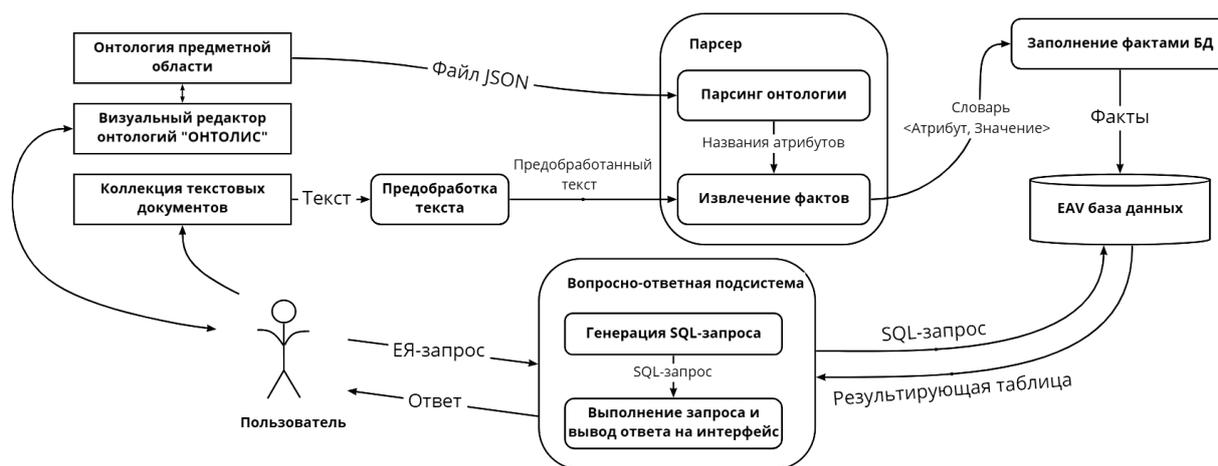


Рис. 2. Схема работы MagNolis

Для реализации унифицированного решения по хранению и доступу к данным мы выбрали EAV (англ., Entity-Attribute-Value) модель БД, которую по-русски называют моделью "Сущность-Атрибут-Значение" или "вертикальной" моделью (с вертикальным, а не горизонтальным, как в традиционных реляционных БД, хранением атрибутов).

Модель EAV предлагает универсальную структуру данных, в которой данные и метаданные хранятся единообразно в виде пар "ключ-значение" [9, 10]. Эта альтернативная модель позволяет максимально упростить структуры хранения данных, что обеспечивает высокую гибкость изменения логической схемы данных.

В частности, можно в динамике добавлять любые необходимые атрибуты в структуру данных без изменения самой схемы БД. Это очень важно для нашего подхода, так как

позволяет находить для разных предметных областей данные об информационных объектах, которые нужно автоматически извлекать из текстовых документов, использовать одну и ту же схему БД, обеспечивая тем самым адаптацию к специфике предметной области решаемой задачи.

В реализации описанного подхода не требуется ручной ввод данных, так как все данные поступают в БД непосредственно путем извлечения нужных фактов из входных текстовых документов. Однако не представляет никакой проблемы расширить систему дополнительной функциональной возможностью ручного ввода данных, сделав этот способ доступным по выбору.

На рис. 3 представлен результат автоматического извлечения фактов из коллекции текстовых документов анкет сотрудников некоего предприятия.

Атрибут	Значение
Фамилия, имя, отчество	Оглезнева Яна Петровна
Статус	Молодой специалист
Возраст	25
Должность	инженер 1 категории
Подразделение	УРНГМ ОМНГМ
Уровень образования	Высшее
Учебное заведение	Пермский Национальный Исследовательский Политехнический Университет
Специальность	Геология нефти и газа
Мероприятия Общества	Научные конференции, научная работа, Тренинги и обучения, Конкурс "Молодой лидер-нефтяник".

Рис. 3. Извлечение фактов из текста

Результат работы вопросно-ответной подсистемы (сервиса ЕЯ-запросов) для запроса "Какое учебное заведение окончил сотрудник?" представлен на рис. 4.

Полный путь к документу	Имя сущности	Дата	Ответ
D:\Университет\ Курсовая работа \Материал с анкетами\анкета1.docx	Красильникова Диана Анатольевна	19.04.2022	Пермский Национальный Исследовательский Политехнический Университет
D:\Университет\ Курсовая работа \Материал с анкетами\анкета2.docx	Петров Руслан Артемович	19.04.2022	Пермский Национальный Исследовательский Политехнический Университет
D:\Университет\ Курсовая работа \Материал с анкетами\анкета3.docx	Вилисов Дмитрий Александрович	19.04.2022	Пермский Национальный Исследовательский Политехнический Университет
D:\Университет\ Курсовая работа \Материал с анкетами\анкета4.docx	Безматерных Роман Петрович	19.04.2022	Пермский Нефтяной Колледж
D:\Университет\ Курсовая работа \Материал с анкетами\анкета5.docx	Оглезнева Яна Петровна	19.04.2022	Пермский Национальный Исследовательский Политехнический Университет

Рис. 4. Результат работы вопросно-ответной подсистемы

В текущей версии системы MagNolis ЕЯ-запросы в вопросно-ответной подсистеме фактически являются запросами на значение конкретного атрибута с опциональной возможностью поиска по ключевому слову. Тем не менее, есть возможность подключить под-

держку вопросов логического характера, свойственных экспертным системам с ответами вида "да/нет" (эта возможность обычно отсутствует в распространенных поисковых системах сети Интернет).

## Онтологически управляемая верификация исходного кода приложений БД

Далее мы представляем наш унифицированный подход к обработке текстов на искусственных языках для решения задач верификации исходных кодов приложений БД. Следует подчеркнуть общность концепций обоих описанных в данной статье подходов в части использования онтологически управляемых решений.

Верификация (англ., Verification) – это процесс оценки системы или ее компонентов с целью определения удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа [11]. Процесс верификации включает в себя инспекции, тестирование кода, анализ результатов тестирования, формирование и анализ отчетов о проблемах, таким образом, тестирование программ является составной частью верификации.

С ростом сложности ПО, потребности в его постоянном обновлении и усовершенствовании растет не только количество ошибок в программном коде, но и сложность их выявления. Ввиду того, что программное обеспечение лежит в основе почти любой инфраструктуры, задача обеспечения его качества и, соответственно, автоматизация процессов верификации и тестирования, сегодня по-прежнему актуальны.

Общая проблема заключается в отладке логики и семантики приложений, при этом отладка управляемых знаниями приложений по своей сути является как раз проверкой того, насколько семантика базы знаний адекватно интегрирована и учитывается при разработке и модификации ПО.

Отдельно подчеркнем, что автоматизация процессов верификации на основе управляемых знаниями решений способна помочь преподавателям образовательных учреждений в подготовке программистов высокой квалификации, а также может помочь самим обучающимся повысить свою квалификацию в области разработки ПО.

В контексте задачи автоматизации труда преподавателя он выступает и как постановщик задач, и как, фактически, валидатор

исходного кода студенческих программ, так как их верификацию обязаны выполнять сами обучающиеся.

Известно, что широкодоступным средствам тестирования и отладки приложений обычно не хватает семантической мощности как самих средств выявления ошибок, так и средств генерации текста сообщений об ошибках с учетом уровня квалификации пользователя (обычно сообщения об одном и том же типе ошибки неизменны и не зависят от категории пользователя, например, новичка или профессионала).

Кроме того, традиционные средства тестирования обычно не учитывают специфику и семантику приложений, концентрируясь, в основном, на выявлении синтаксических ошибок. Это является следствием как огромного разнообразия решаемых задач и большой сложности разрабатываемых приложений, так и сложностью самой проблемы верификации их логики и семантики. Поэтому мы пошли по пути разработки в первую очередь средств верификации только для определенной категории приложений – приложений реляционных баз данных.

Выбранный нами управляемый онтологиями подход в рамках OBDA-парадигмы способен за счет расширения онтологической базы знаний постоянно повышать уровень средств верификации ПО, а также в перспективе может быть расширен и для других категорий приложений.

Одним из главных преимуществ данного подхода является то, что для обеспечения изменений в работе верификатора, расширения его возможностей и адаптации к уровню квалификации пользователя и специфике решаемой задачи, требуется внести изменения в онтологию, управляющую работой верификатора, без необходимости внесения изменений в исходных код основных компонент верификатора.

На рис. 5 представлена архитектура разработанной нами демонстрационной версии верификатора SA2test (Smart Assistant to Test), который выступает в роли интеллектуального помощника (англ., SA – Smart Assistant) в процессе написания, верификации и валидации приложений БД.

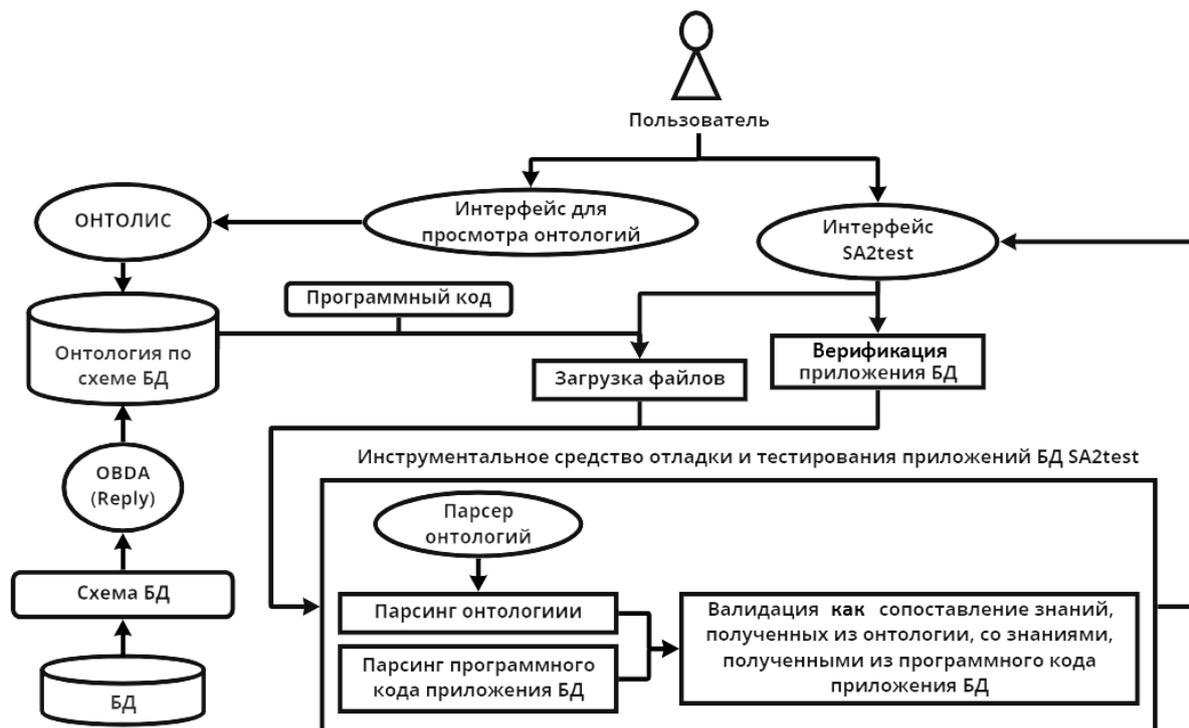


Рис. 5. Архитектура SA2test

Приложение состоит из трех основных блоков, реализующих парсинг внутреннего представления онтологий (в формате JSON), парсинг исходного программного кода тестируемого приложения и его верификацию в части проверки правильности составления SQL-запросов (на примере СУБД PostgreSQL).

На вход SA2test поступает программный код приложения БД и онтология, которая строится автоматически по схеме БД с помощью сервиса Reply [8].

Вершины этой онтологии, назовем ее онтологией схемы БД, описывают таблицы БД и их атрибуты, а дуги представляют связи таблиц с их атрибутами, а также взаимосвязи между таблицами.

Кроме того, онтология схемы БД содержит знания о поддерживаемых типах данных. Рассмотрение вопросов, связанных с адаптацией описываемого приложения к сре-

де различных СУБД выходит за рамки данной работы.

Основная цель парсинга внутреннего представления онтологии – получить знания о взаимосвязях между вершинами онтологии. В процессе верификации происходит сопоставление данных, полученных при парсинге онтологии схемы БД, с данными, полученными из программного кода в процессе автоматического разбора текстов SQL-операторов.

Результаты верификации отправляются на интерфейс пользователя, который представлен на рис. 6.

Приложение обрабатывает выявленные ошибки и выводит пользователю данные о местоположении ошибки в исходном тексте (выделение цветом), а также само описание ошибки. Текст сообщения об ошибке может быть изменен на уровне интерфейса пользователя лицами, имеющими права администратора или инженера по знаниям.

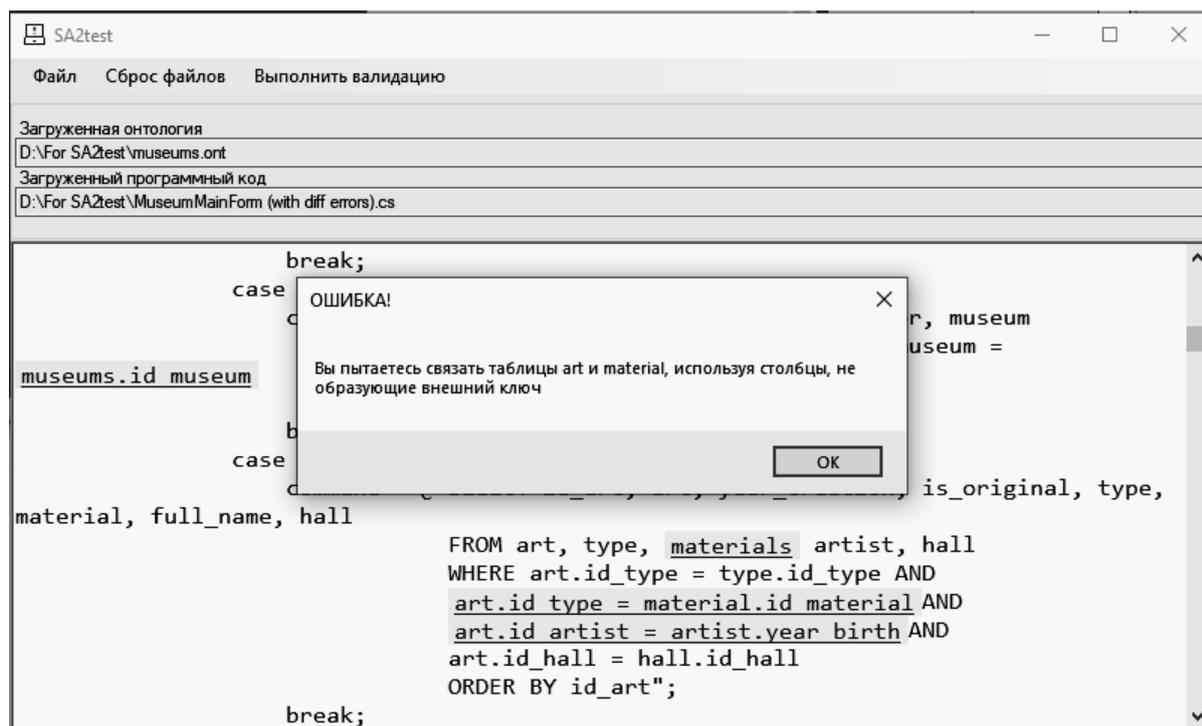


Рис. 6. Интерфейс пользователя приложения SA2Test

Кроме того, в базе знаний системы может храниться несколько вариантов текстов сообщений на один и тот же тип ошибки в привязке к определенным категориям конечных пользователей, что позволяет выдавать разную информацию (разной степени детализации и с разными формулировками) с учетом квалификации пользователя (уровень квалификации определяется при регистрации пользователя в системе). Сами сообщения могут быть сформулированы таким образом, что фактически станут выступать в качестве аналога объяснительной компоненты экспертных систем.

Создание такого рода инструментальных средств облегчает тестирование программ и сокращает общее время отладки, особенно для начинающих пользователей.

Кроме того, это приложение можно рассматривать как средство обучения построению правильных SQL-запросов, а также как средство автоматизации труда преподавателя по валидации программного кода студенческих приложений.

## Заключение

Представленный в данной работе подход к разработке управляемых онтологиями

систем позволяет унифицированным образом решать самые разные задачи, связанные с автоматической обработкой текстов как на естественных, так и на искусственных языках.

Основным достоинством описанных средств автоматизации построения реляционных ИС с точки зрения конечного пользователя является возможность автоматического построения ИС по типу АРМ без проектирования БД и ручного ввода данных, а в представленных инструментах верификации – возможность проведения верификации программного кода без необходимости написания полного набора отладочных тестов и запуска программ. С точки зрения разработчика ПО описанные инструментальные средства выступают в качестве интеллектуальных помощников, способных значительно сократить общее время разработки и отладки приложений БД.

В перспективе планируется создание интеллектуального репозитория для хранения проектов, созданных в среде описанных инструментальных средств, и переиспользования имеющихся онтологий в новых проектах. Кроме того, имеется потребность в совершенствовании интерфейса и обогащении систем расширенными средствами NLP (Natural Language Processing).

### Список литературы

1. *David S. Frankel. Model Driven Architecture: Applying MDA to Enterprise Computing. Willey publishing Inc: Indianapolis Indiana USA, 2003.*
2. *Гаврилова Т.А. Инженерия знаний. Модели и методы: учебник // Т.А. Гаврилова, Д.В. Кудрявцев, Д.И. Муромцев. Санкт-Петербург: Изд-во "Лань", 2016. 324 с.*
3. *Kontchakov, R., Rodriguez-Muro M., Zakharyashev M. Ontology-Based Data Access with Databases: A Short Course. In: Rudolph, S., Gottlob, G., Horrocks, I., van Harmelen, F. (eds) Reasoning Web. Semantic Technologies for Intelligent Data Access. Reasoning Web 2013. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2013. Vol. 8067.*
4. *Svetlana Chuprina, Igor Postanogov, and Olfa Nasraoui. Ontology Based Data Access Methods to Teach Students to Transform Traditional Information Systems and Simplify Decision Making Process // Procedia Computer Science, 2016. Vol. 80. P. 1801–1811.*
5. *Гаврилова Т.А., Червинская К.Р. Извлечение и структурирование знаний для экспертных систем. М.: Радио и связь, 1992. 200 с.*
6. *Чуприна С.И., Зиненко Д.В. ОНТОЛИС: адаптируемый визуальный редактор онтологий // Вестник Пермского университета. Математика. Механика. Информатика. 2013. Вып. 3(22). С. 106–110.*
7. *Чуприна С.И., Постаногов И.С. Концепция обогащения унаследованных информационных систем сервисом запросов на естественном языке // Вестник Пермского университета. Математика. Механика. Информатика. 2015. Вып. 2 (29). С. 78–86.*
8. *Постаногов И.С. Использование онтологий для доступа к данным с целью интеллектуализации информационных систем // Фундаментальные и прикладные проблемы механики, математики, информатики: сб. докл. всерос. науч.-практ. конф. с междунар. участием. Пермь, 2015. С. 360–363.*
9. *Рогозов Ю.И., Свиридов А.С., Кучеров С.А., Жибулис Ю.А. Подход к реализации БД со статической структурой на основе модели данных EAV // Известия Южного федерального университета. Технические науки. 2010. Т. 103, № 2. С. 87–92.*
10. *Гмарь Д.В., Игнатова Ю.А., Цуранов Э.В., Шахгельдян К.И. Эффективные методы работы с вертикальной моделью данных // ИТиВС. 2015. Вып. 2. С. 68–78.*
11. *Медведева О.А. Программное обеспечение гарантии качества. Казань: Казан. ун-т, 2018. 112 с.*

### References

1. *David S. Frankel. Model Driven Architecture: Applying MDA to Enterprise Computing. Willey publishing Inc: Indianapolis Indiana USA, 2003.*
2. *Gavrilova T.A. Inzheneriya znaniy. Modeli i metody: uchebnik // T.A. Gavrilova, D.V. Kudryavcev, D.I. Muromcev. Sankt-Peterburg: Izd-vo "Lan' ", 2016. 324 s.*
3. *Kontchakov R., Rodriguez-Muro M., Zakharyashev M. Ontology-Based Data Access with Databases: A Short Course. In: Rudolph, S., Gottlob, G., Horrocks, I., van Harmelen, F. (eds) Reasoning Web. Semantic Technologies for Intelligent Data Access. Reasoning Web 2013. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2013. Vol. 8067.*
4. *Svetlana Chuprina, Igor Postanogov and Olfa Nasraoui. Ontology Based Data Access Methods to Teach Students to Transform Traditional Information Systems and Simplify Decision Making Process // Procedia Computer Science, 2016. Vol. 80. P. 1801–1811.*
5. *Gavrilova T.A., CHervinskaya K.R. Izvlechenie i strukturirovanie znaniy dlya ekspertnyh sistem. M.: Radio i svyaz', 1992. 200 s.*
6. *CHuprina S.I., Zinenko D.V. ONTOLIS: adaptiruemyj vizual'nyj redaktor ontologij // Vestnik Permskogo universiteta. Matematika. Mekhanika. Informatika. 2013. Vyp. 3(22). S. 106–110.*
7. *CHuprina S.I., Postanogov I.S. Konceptsiya obogashcheniya unasledovannyh informacionnyh sistem servisom zaprosov na estestvennom yazyke // Vestnik Permskogo universiteta. Matematika. Mekhanika. Informatika. 2015. Vyp. 2 (29). S. 78–86.*
8. *Postanogov I.S. Ispol'zovanie ontologij dlya dostupa k dannym s cel'yu intellektualizacii informacionnyh sistem // Fundamental'nye i prikladnye problemy mekhaniki, matematiki, informatiki: sb. dokl. vseros. nauch.-prakt. konf. s mezhdunar. uchastiem. Perm', 2015. S. 360–363.*

9. Rogozov YU.I., Sviridov A.S., Kucherov S.A., ZHibulis YU.A. Podhod k realizacii BD so staticheskoj strukturoj na osnove modeli dannyh EAV // Izvestiya YUzhnogo federal'nogo universiteta. Tekhnicheskie nauki. 2010. T. 103, № 2. S. 87–92.
10. Gmar' D.V., Ignatova YU.A., Curanov E.V., SHahgel'dyan K.I. Effektivnye metody raboty s vertikal'noj model'yu dannyh // ITiVS. 2015. Vyp. 2. С. 68–78.
11. Medvedeva O.A. Programmnoe obespechenie garantii kachestva. Kazan': Kazan. un-t, 2018. 112 s.

**Просьба ссылаться на эту статью:**

Гимашева К.В., Гладких Е.А., Чуприна С.И. Унифицированное применение методов онтологического инжиниринга в задачах обработки текстовых данных // Вестник Пермского университета. Математика. Механика. Информатика. 2022. Вып 3(58). С. 49–58. DOI: 10.17072/1993-0550-2022-3-49-58.

**Please cite this article as:**

Gimasheva K.V., Gladkikh E.A., Chuprina S.I. An Ontology-Driven Methods Unified Application in Text Data Processing Problems // Bulletin of Perm University. Mathematics. Mechanics. Computer Science. 2022. Issue 3(58). P. 49–58. DOI: 10.17072/1993-0550-2022-3-49-58.