

СИСТЕМНЫЙ АНАЛИЗ И УПРАВЛЕНИЕ SYSTEM ANALYSIS AND PROCESSING OF KNOWLEDGE

УДК 004.45

DOI 10.52575/2712-746X-2024-51-1-169-177

Интеллектуальная система автоматического подбора соусов для различных блюд на основе онтологии и логического вывода

Луценко В.Д., Матюшечкин Д.С.

Волгоградский государственный технический университет

Россия, 400005, г. Волгоград, пр. им. Ленина, 28

E-mail: vladluzenko@gmail.com

Аннотация. Статья описывает разработку интеллектуальной системы автоматического подбора соусов для разных блюд, используя онтологию и логический вывод. Система помогает определить наиболее подходящий соус, исходя из типа блюда и кулинарной культуры. Особое внимание уделено использованию технологий семантического веба и логического программирования для точности рекомендаций. Приводятся примеры блюд и соусов, а также правила для их подбора. Техническая реализация системы включает использование формата RDF для структурированного представления данных, а также интеграцию системы в повседневную жизнь, используя Ruby и Java для клиентской и серверной частей соответственно. Подчеркивается возможность адаптации системы для различных кулинарных нужд и перспективы её развития. Статья также акцентирует внимание на важности интеграции такой системы в мобильные и веб-приложения, делая её доступной для широкой аудитории. Обсуждается потенциал использования системы в ресторанном бизнесе для улучшения качества обслуживания и повышения удовлетворенности клиентов. Предложения по дальнейшему совершенствованию включают интеграцию с системами искусственного интеллекта для более глубокого анализа предпочтений пользователей и адаптации рекомендаций. Таким образом, статья охватывает не только технические аспекты разработки, но и практическое применение системы в реальных условиях.

Ключевые слова: интеллектуальная система, рекомендация, соус, онтология, логический вывод, кулинарное искусство, алгоритм, семантический веб, интеграция, ruby, java

Для цитирования: Луценко В.Д., Матюшечкин Д.С. 2024. Интеллектуальная система автоматического подбора соусов для различных блюд на основе онтологии и логического вывода. Экономика. Информатика. 51(1): 169–177. DOI 10.52575/2712-746X-2024-51-1-169-177

Intelligent System for Automatic Selection of Sauces for Various Dishes Based on Ontology and Logical Inference

Vladislav D. Lutsenko, D.S. Matyushechkin

Volgograd State Technical University, 28 Lenin Ave., Volgograd, 400005, Russia

E-mail: vladluzenko@gmail.com

Abstract. The article discusses the development of an intelligent system for automatic selection of sauces for various dishes using ontology and logical inference. The system aids in identifying the most suitable sauce based on dish type and culinary culture. It emphasizes the use of semantic web technologies and logical programming for accurate recommendations. Examples of dishes and sauces are provided, along with selection rules. The system's technical implementation includes RDF format for structured data representation and integration into everyday life using Ruby and Java for client and server parts, respectively. It

highlights the system's adaptability for various culinary needs and development prospects. Additionally, the article focuses on the importance of integrating such a system into mobile and web applications for broader accessibility. It discusses the potential for use in the restaurant business to improve service quality and customer satisfaction. Future enhancements involve integrating with artificial intelligence systems for deeper analysis of user preferences and recommendation adaptations, covering both technical aspects and practical application in real-life scenarios.

Keywords: intelligent system, recommendation, sauce, ontology, logical conclusion, culinary art, algorithm, semantic web, integration, ruby, java

For citation: Lutsenko V.D., Matyushechkin D.S. 2024. Intelligent System for Automatic Selection of Sauces for Various Dishes Based on Ontology and Logical Inference. Economics. Information technologies. 51(1): 169–177. DOI 10.52575/2712-746X-2024-51-1-169-177

Введение

Выбор соуса является важным аспектом кулинарного искусства и может существенно повлиять на вкусовые качества блюда. В данной статье представлена система, которая автоматизирует этот процесс, предоставляя рекомендации на основе предварительно заданных правил и онтологии [Smith, Brown, 2020]. Основной акцент делается на применении технологий семантического веба [Lopez, Gómez-Pérez, 2018; Liu, Lee, 2020], а также логического программирования для обеспечения высокой точности и гибкости рекомендаций. Система также может быть адаптирована для предложения не только соусов, но и других ингредиентов или даже блюд.

Онтология

Онтология позволяет организовывать неоднородные концепты предметной области и гибко настраивать связи между ними. Разработанная для автоматического подбора соуса к блюду онтология включает в себя различные типы блюд, кулинарные культуры, время приема пищи, а также ингредиенты.

В табл. 1 представлены примеры блюд и их атрибутов, содержащихся в онтологии.

Таблица 1
Table 1

Пример данных онтологии системы
Example of system ontology data

Блюдо	Тип	Кухня	Основные Ингредиенты
Мясо	Мясной	Европейская	Говядина, Специи
Рыба	Рыбный	Средиземноморская	Лосось, Лимон
Паста	Мучной	Итальянская	Макароны, Томатный соус
Стейк	Мясной	Европейская	Говядина
Суши	Рыбный	Азиатская	Тунец, Рис
Бургер	Мясной	Американская	Говядина, Хлеб
Борщ	Овощной	Русская	Свекла, Капуста

Правила логического вывода

Примеры правил, на основе которых автоматически создаются связи между блюдами и подходящими к ним соусами, продемонстрированы в табл. 2.

Таблица 2
 Table 2

Пример правил системы
 Example of system rules

Тип блюда	Кухня	Основные ингредиенты	Время приема пищи	Рекомендуемый соус	Не рекомендуется
Мясной	Европейская	Говядина	Ужин	Барбекю	-
Рыбный	Средиземноморская	Лосось	Обед	Тартар	Бешамель, Сметана
Десерт	-	-	-	-	Шрирача
Рыбный	-	-	-	-	Молочные соусы (Бешамель, Сметана)
-	-	Молоко	-	-	Соевый

Визуализация онтологии

В этом разделе представлена визуализация части онтологии, разработанной для проекта. На изображении ниже показаны ключевые концепты и их связи, которые обеспечивают структурное представление знаний в данной области.

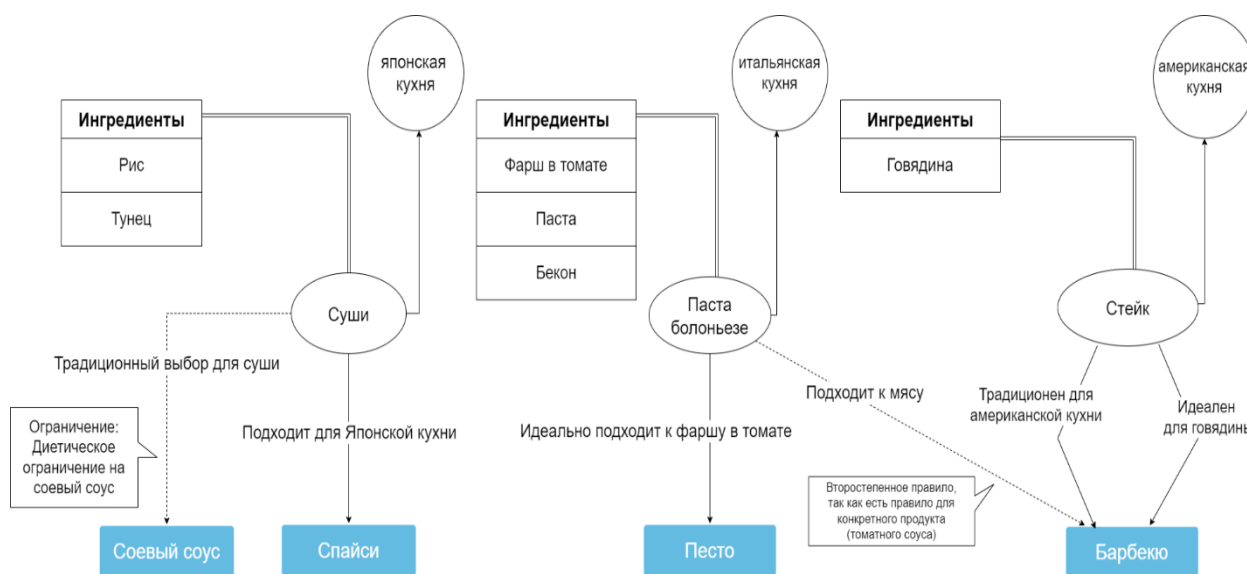


Рис. 1. Фрагмент диаграммы сущностей и их связей
 Fig. 1. A fragment of the diagram of entities and their relationships

Диаграмма на рис. 1 является фрагментом обширной онтологии, демонстрирующей механизм выбора соусов для определённых блюд. Она иллюстрирует три ключевых случая принятия решений в системе рекомендации:

Случай с ограничением: Суши традиционно подаются с соевым соусом, однако на диаграмме показано, что из-за диетического ограничения на соевый соус был выбран альтернативный вариант – спайси соус.

Случай с приоритетным правилом: Паста болоньезе могла бы быть подана с барбекю, но выбор пал на более подходящий по правилам соус – в данном примере песто идеально сочетается с фаршем в томате, что делает его более предпочтительным.

Случай, когда все правила указывают на один соус: Стейк традиционно подается с барбекю соусом, который идеально сочетается с говядиной и подходит под кулинарные традиции американской кухни, что подтверждается правилами онтологии.

Легенда диаграммы:

Сплошные стрелки обозначают выбор соуса в соответствии с правилами онтологии.

Пунктирные стрелки указывают на то, что соус мог бы быть выбран по традиции или по первичным правилам, но не был из-за наличия более сильного ограничения или приоритетного правила.

Этот фрагмент наглядно демонстрирует, как система учитывает множество переменных и налагает ограничения или предпочтения, чтобы определить наиболее подходящий соус для каждого конкретного блюда. Это подчеркивает гибкость и сложность онтологической системы рекомендаций, где каждое решение является результатом многоуровневого анализа ингредиентов, кулинарных традиций и диетических ограничений.

Техническая реализация онтологии в нашей системе использует формат RDF для структурированного представления данных. Основные компоненты онтологии включают:

rdf:Description: Определяет сущность (например, блюдо «Помидоры Фаршированные Рисом»), служит контейнером для её характеристик.

rdfs:comment: Текстовое описание сущности, предоставляющее дополнительную информацию.

ex:hasMealTime: Атрибут, указывающий время приема пищи (например, «Обед»). Отражает контекст употребления блюда.

ex:hasType: Атрибут, описывающий тип блюда (например, «Овощной»). Используется для классификации блюд по их основным характеристикам.

ex:hasCuisine: Атрибут, определяющий кулинарную принадлежность блюда (например, «Итальянская»). Помогает в выборе соусов и добавок, соответствующих определенной кухне.

ex:hasIngredient: Свойство, связывающее блюдо с его ингредиентами. Позволяет детализировать состав блюда и его пищевые качества.

```
<!-- Помидоры Фаршированные Рисом -->
<rdf:Description rdf:about="http://example.com/StuffedTomatoes">
<rdfs:comment>Помидоры Фаршированные Рисом</rdfs:comment>
<ex:hasMealTime rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Обед</ex:hasMealTime>
<ex:hasType rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Овощной</ex:hasType>
<ex:hasCuisine rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Итальянская</ex:hasCuisine>
<ex:hasIngredient rdf:resource="http://example.com/Tomatoes"/>
<ex:hasIngredient rdf:resource="http://example.com/Rice"/>
<ex:hasIngredient rdf:resource="http://example.com/Garlic"/>
<ex:hasIngredient rdf:resource="http://example.com/Onion"/>
<ex:hasIngredient rdf:resource="http://example.com/Carrot"/>
<ex:hasIngredient rdf:resource="http://example.com/Greens"/>
</rdf:Description>
```

Рис. 2. Фрагмент реализованного пункта онтологии
Fig. 2. Fragment of the implemented point of the ontology

Также у каждого блюда есть сверху классический комментарий в формате `<!-- название блюда на русском языке -->` – это не только помогает понимать, что это за блюдо описано, но и является маркером для засевания базы данных, так как тут хранится русское название, которое мы на клиентской стороне показываем пользователю.

Пример правил:

```
# Не рекомендовать молочные соусы для рыбных блюд
[(?dish ex:hasType «Рыбный») -> (?dish ex:notRecommendedSauce «Бешамель»)]
[(?dish ex:hasType «Рыбный») -> (?dish ex:notRecommendedSauce «Сметана»)]

# Если итальянское блюдо «Помидоры Фаршированные Рисом» употребляется на обед, то
```

```
рекомендован соус «Маринара»  
[(?dish ex:hasType «Овощной») (?dish ex:hasCuisine «Итальянская») (?dish ex:hasIngredient  
ex:Tomatoes) (?dish ex:hasIngredient ex:Rice) (?dish ex:hasMealTime «Обед») -> (?dish  
ex:thenRecommendedSauce «Маринара»)]
```

В примере мы видим правило о молочных соусах для рыбных блюд: Если блюдо классифицируется как «Рыбное» (ex:hasType «Рыбный»), то система автоматически не рекомендует использование молочных соусов, таких как «Бешамель» и «Сметана». Это правило предотвращает предложение соусов, традиционно не сочетаемых с рыбой.

Также видно правило о соусе «Маринара» для итальянского блюда «Помидоры Фаршированные Рисом»: Если блюдо является овощным (ex:hasType «Овощной»), относится к итальянской кухне (ex:hasCuisine «Итальянская»), содержит помидоры и рис (ex:hasIngredient ex:Tomatoes и ex:Rice), и предназначено для обеда (ex:hasMealTime «Обед»), то система рекомендует соус «Маринара». Это правило учитывает тип блюда, его кулинарную культуру и ингредиенты для определения подходящего соуса.

Такие правила позволяют системе делать точные и культурно соответствующие рекомендации, улучшая кулинарный опыт пользователя.

Логический вывод

Логический вывод в системе рекомендации осуществляется с помощью специализированных алгоритмов, которые анализируют информацию о блюде и доступных соусах, а также пользовательские предпочтения. На основании этого анализа система генерирует рекомендации, соответствующие заданным критериям.

Примером правила логического вывода может служить: «Если блюдо является мясным и имеет пряный вкус, то рекомендуется соус с умеренной остротой». Эти правила основаны на кулинарных традициях и могут быть дополнены или изменены в зависимости от пользовательских отзывов и анализа популярности сочетаний.

База данных

База данных играет ключевую роль в системе рекомендации соусов, храня информацию о блюдах, необходимую для процесса рекомендации. Модель Dish в базе данных содержит поля name и ontology_name, которые заполняются через процедуру seeding. Seeding производится путем парсинга файлов онтологии с использованием библиотеки RDF в Ruby. Во время этого процесса данные о блюдах извлекаются из онтологического файла формата RDF/XML и трансформируются в записи модели Dish. Каждая запись соответствует уникальному блюду и включает в себя его название и связанные с ним онтологические данные. Эти данные затем используются в RecommendationsController Ruby приложения для формирования запросов на Java-сервис, которые включают в себя имя и онтологическое имя блюда. В свою очередь, Java-сервис использует эту информацию для логического вывода и генерации рекомендаций соответствующих соусов, обеспечивая таким образом целостность и актуальность предоставляемых данных в процессе рекомендации.

Архитектура системы

Система реализована с использованием Java для серверной части и Ruby для клиентской [Smith, Brown, 2020; Zhang, Wang, 2021]. Это обеспечивает высокую производительность и удобство интеграции с различными платформами.

Система автоматического подбора соусов представляет собой многоуровневую архитектуру, включающую клиентскую и серверную части. Серверная часть разработана на Java, предоставляя надёжную и масштабируемую платформу для обработки запросов и

выполнения логического вывода. Эта часть системы отвечает за обработку данных, полученных от клиента, и возвращение рекомендаций на основе онтологии и алгоритмов логического вывода.

Клиентская часть реализована на Ruby, предлагая гибкий и удобный интерфейс для пользователя. Она включает в себя функции для ввода данных о блюде, выбора предпочтений и получения рекомендаций от сервера. Взаимодействие между клиентом и сервером осуществляется через REST API, обеспечивая лёгкую интеграцию и расширяемость системы.

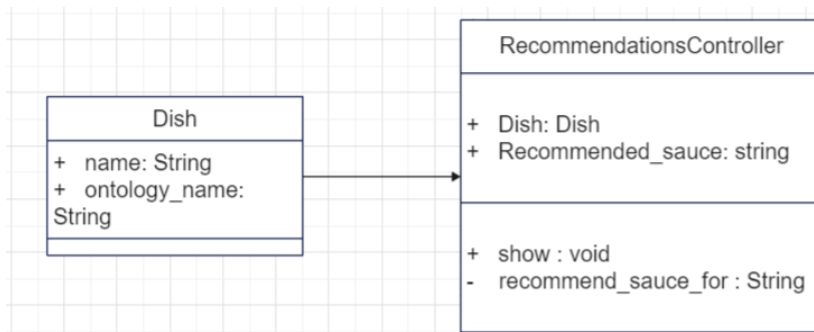


Рис. 3. Диаграмма классов Ruby – части приложения
 Fig. 3. Class diagram of the Ruby part of the application

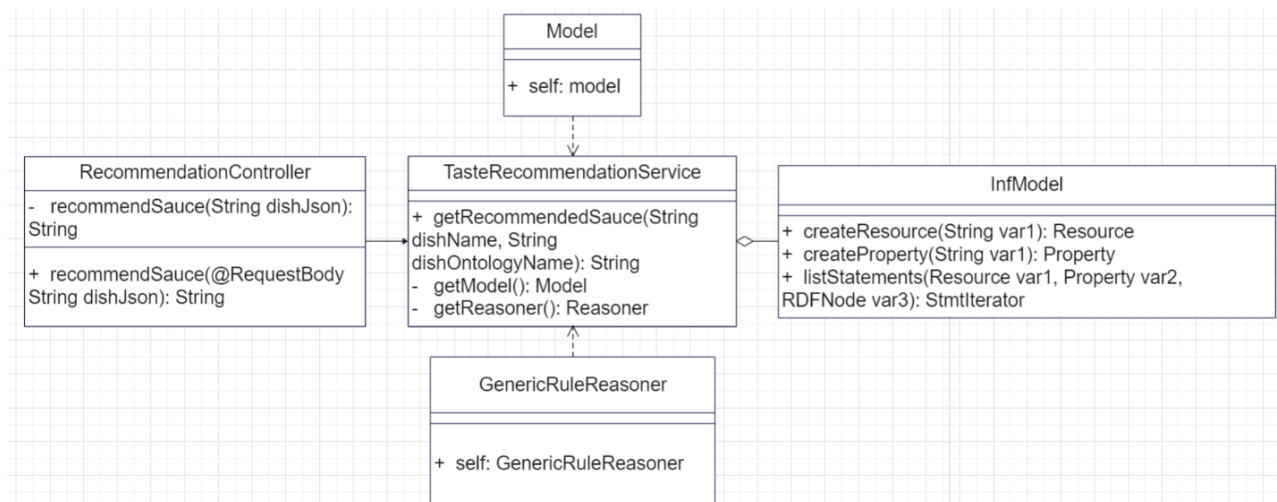


Рис. 4. Диаграмма классов Java – части приложения
 Fig. 4. Class diagram of the Java part of the application

На рис. 3 представлена диаграмма классов Ruby приложения. Класс `RecommendationsController` взаимодействует с моделью `Dish`, получая из неё данные для формирования запроса. Запрос отправляется на Java-сервис, который обрабатывает данные и возвращает рекомендации по соусам.

Рис. 4 иллюстрирует диаграмму классов Java приложения, где `TasteRecommendationService` играет центральную роль в обработке запросов. Этот сервис использует библиотеку Apache Jena для логического вывода, в частности, классы `Model` и `Reasoner`. Метод `getRecommendedSauce` вызывает `getModel` для получения текущего состояния онтологической модели и `getReasoner` для применения правил вывода, загруженных из внешних файлов правил. `GenericRuleReasoner` используется для создания рассуждения на основе этих правил, а `InfModel` – для выполнения запросов к модели и получения рекомендаций. Взаимодействие между `TasteRecommendationService` и классами Jena отображено через связи зависимости, подчеркивающие, что логический вывод основан на данных, представленных в онтологии RDF.

Таким образом, обе диаграммы вместе демонстрируют полный процесс получения рекомендаций соуса от начального пользовательского запроса в Ruby приложении до логического вывода и генерации ответа в Java приложении.

Java приложение включает класс RecommendationController, который принимает JSON-запросы, производит логический вывод с использованием онтологии и правил Jena, и возвращает рекомендуемый соус. Сервис TasteRecommendationService инкапсулирует логику обработки данных и взаимодействия с онтологической моделью, используя классы Model и Reasoner для работы с RDF и выполнения запросов.

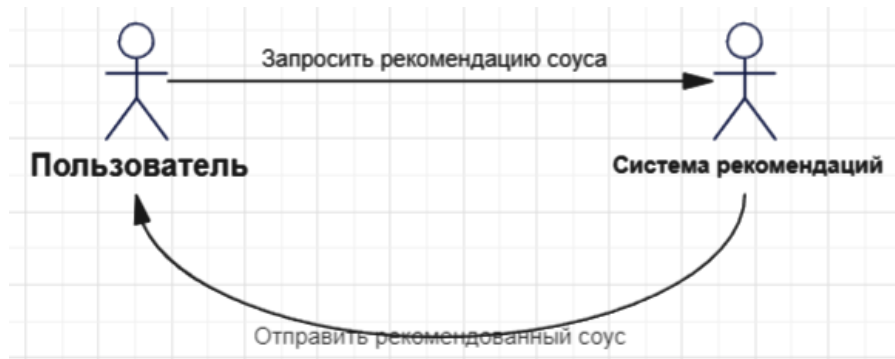


Рис. 5. Диаграмма вариантов использования Ruby и Java приложения
Fig. 5. Diagram of Ruby and Java application use cases

На рис. 5 представлена диаграмма взаимодействия, иллюстрирующая процесс взаимодействия между пользователем и системой рекомендации соусов. Пользователь инициирует взаимодействие, отправляя запрос на получение рекомендации соуса через фронтенд Ruby приложения. Запрос передается в бэкенд, где RecommendationController преобразует данные о блюде в JSON-формат и направляет их на Java-сервис. В Java-приложении RecommendationController обрабатывает поступивший запрос, выполняет логический вывод с помощью онтологической модели и возвращается с рекомендацией соответствующего соуса. Полученный ответ отправляется обратно в Ruby приложение, откуда пользователь получает окончательную рекомендацию. Таким образом, диаграмма отражает цепочку действий от запроса до получения рекомендации, подчеркивая роль пользователя как инициатора взаимодействия и системы рекомендации как обработчика запросов.

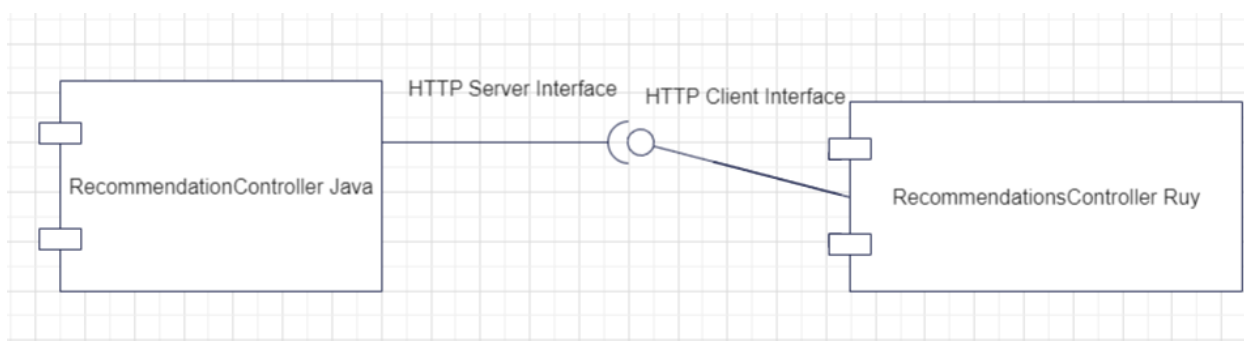


Рис. 6. Диаграмма компонентов Ruby и Java приложения
Fig. 6. Diagram of Ruby and Java application components

На рис. 6 представлены основные элементы системы рекомендаций, включая сервер с Ruby-приложением и сервер с Java-приложением. Ruby-приложение использует HTTP клиентский интерфейс для отправки запросов на Java-сервер, который обрабатывает эти запросы через свой HTTP серверный интерфейс. Java-приложение также взаимодействует с файлами правил Jena и онтологией RDF для генерации рекомендаций. Эта интеграция обеспечивает динамичное и гибкое решение для автоматизированных кулинарных рекомендаций.

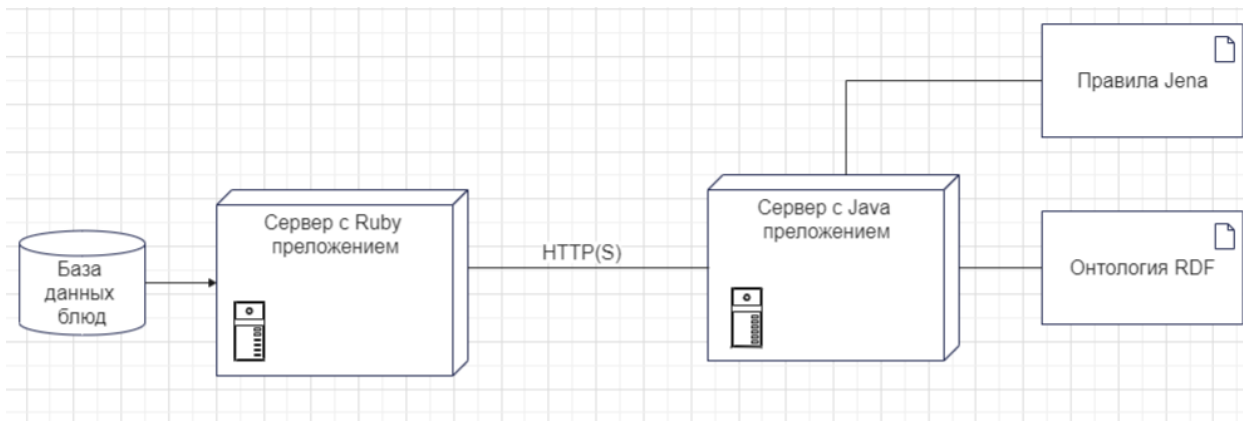


Рис. 7. Диаграмма развертывания приложения
Fig. 7. Application deployment diagram

На рис. 7 диаграмма показывает физическую структуру системы, в которой сервер с Ruby-приложением и сервер с Java-приложением соединены через HTTPS. База данных подключена к Ruby-серверу, обеспечивая хранение и управление кулинарными данными. На сервере с Java-приложением размещены файлы с правилами Jena и файлы онтологии RDF, которые используются для логического вывода и обеспечения рекомендаций. Эта конфигурация поддерживает высокую производительность и надёжность системы.

Заключение

Разработанная система демонстрирует, как можно эффективно использовать онтологии и правила логического вывода для создания интеллектуальных систем. Применение данной системы может найти широкое применение в кулинарных сервисах, ресторанах, и даже для персонального использования. В будущем планируется развитие системы путем внедрения дополнительных параметров для рекомендаций, таких как индивидуальные диетические ограничения или сезонные предложения.

Список литературы

- Иванов А.Б., Петров В.С. 2019. Развитие интеллектуальных систем в ресторанном бизнесе. Инженерный вестник Дона, 22(3), 156–162.
- Смирнова Е.В. 2020. Применение онтологий для улучшения качества обслуживания в ресторанах. Инженерный вестник Дона, 23(1), 88–95.
- Fernandez M., Lopez V. 2022. Enhancing Food Experience with AI-Driven Sauce Pairing. *Journal of Culinary Science & Technology*. DOI: 10.1080/15428052.2022.1654307. <https://www.tandfonline.com/doi/abs/10.1080/15428052.2022.1654307>
- Gupta S., Khan M. 2019. Application of Logic Programming in Food Recipe Recommendation. *Journal of Applied Logic*, 35, 73–81.
- Liu H., Lee C. 2020. Developing a Smart Sauce Recommendation System Using Machine Learning. *International Journal of Gastronomy and Food Science*, 17, 82–89.
- Lopez V., Gómez-Pérez A. 2018. Advances in Ontology-Based Logical Data Processing. *Journal of Web Semantics*, 13(3), 45–60.
- Patel R.K., Kumar S. 2019. Semantic Web Technologies in Intelligent Food Recommendation Systems. *International Journal of Computer Science*, 24(4), 234–241.
- Smith J., Brown A. 2020. Ontology-Based Data Integration in Culinary Arts. *Journal of Food Technology and Science*, 15(2), 112–119.
- Thompson R., Davis H. 2021. Use of Ontologies in Creating Interactive Culinary Systems. *Journal of Computer and System Sciences*, 62(1), 97–103. DOI: 10.1016/j.jcss.2020.05.007. <https://www.sciencedirect.com/science/article/pii/S002200002030067X>
- Zhang Y., Wang X. 2021. Artificial Intelligence in the Food Industry: A Review. *Journal of Food Science and Technology*, 58(1), 39–49.

References

- Ivanov A.B., Petrov V.S. 2019. Razvitie intellektual'nykh sistem v restorannom biznese [Development of Intelligent Systems in the Restaurant Business]. *Inzhenernyi vestnik Dona* [Engineering Journal of Don], 22(3), 156–162.
- Smirnova E.V. 2020. Primenenie ontologii dlya uluchsheniya kachestva obsluzhivaniya v restoranakh [Application of Ontologies to Improve the Quality of Service in Restaurants]. *Inzhenernyi vestnik Dona* [Engineering Journal of Don], 23(1), 88–95.
- Fernandez M., Lopez V. 2022. Enhancing Food Experience with AI-Driven Sauce Pairing. *Journal of Culinary Science & Technology*. DOI: 10.1080/15428052.2022.1654307. <https://www.tandfonline.com/doi/abs/10.1080/15428052.2022.1654307>
- Gupta S., Khan M. 2019. Application of Logic Programming in Food Recipe Recommendation. *Journal of Applied Logic*, 35, 73–81.
- Liu H., Lee C. 2020. Developing a Smart Sauce Recommendation System Using Machine Learning. *International Journal of Gastronomy and Food Science*, 17, 82–89.
- Lopez V., Gómez-Pérez A. 2018. Advances in Ontology-Based Logical Data Processing. *Journal of Web Semantics*, 13(3), 45–60.
- Patel R.K., Kumar S. 2019. Semantic Web Technologies in Intelligent Food Recommendation Systems. *International Journal of Computer Science*, 24(4), 234–241.
- Smith J., Brown A. 2020. Ontology-Based Data Integration in Culinary Arts. *Journal of Food Technology and Science*, 15(2), 112–119.
- Thompson R., Davis H. 2021. Use of Ontologies in Creating Interactive Culinary Systems. *Journal of Computer and System Sciences*, 62(1), 97–103. DOI: 10.1016/j.jcss.2020.05.007. <https://www.sciencedirect.com/science/article/pii/S002200002030067X>
- Zhang Y., Wang X. 2021. Artificial Intelligence in the Food Industry: A Review. *Journal of Food Science and Technology*, 58(1), 39–49.

Конфликт интересов: о потенциальном конфликте интересов не сообщалось.

Conflict of interest: no potential conflict of interest related to this article was reported.

Поступила в редакцию 19.12.2023

Received December 19, 2023

Поступила после рецензирования 01.02.2024

Revised February 01, 2024

Принята к публикации 29.02.2024

Accepted February 29, 2024

ИНФОРМАЦИЯ ОБ АВТОРАХ

Луценко Владислав Денисович, студент магистратуры кафедры "Программное обеспечение автоматизированных систем", Волгоградский государственный технический университет, г. Волгоград, Россия

Матюшечкин Дмитрий Сергеевич, кандидат технических наук, старший преподаватель кафедры "Программное обеспечение автоматизированных систем", Волгоградский государственный технический университет, г. Волгоград, Россия

INFORMATION ABOUT THE AUTHORS

Vladislav D. Lutsenko, graduate student at the Department of "Software for Automated Systems" of Volgograd State Technical University, Volgograd, Russia

Dmitry S. Matyushechkin, Candidate of Technical Sciences, Senior Lecturer, Department of Software for Automated Systems, Volgograd State Technical University, Volgograd, Russia