



ОНТОЛОГИИ И ФОРМАЛИЗАЦИЯ ЗНАНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

Муртазаева У.И.¹,

Джаббарова Х.К.,

Хубиев М.³,

Худойназарова Э.⁴

*^{1,3,4}Самаркандский филиал Ташкентского университета информационных
технологий имени Мухаммада ал-Хоразмий*

*² Самаркандский государственный институт иностранных языков
(СамГИИЯ)*

e-mail: murtazayeva1982@yandex.ru¹

Аннотация. В статье рассматривается онтологии и формализацию знаний как фундаментальный слой архитектуры интеллектуальных систем. Проведен цельный обзор теоретических оснований (логики описаний, онтологические профили OWL, графовая модель RDF), инженерных практик (проектирование, выравнивание, валидация, управление версиями), а также эксплуатационных аспектов (интеграция с данными, механизм рассуждения, мониторинг качества).

Ключевые слова: онтология; формализация знаний; логики описаний; RDF; OWL; SPARQL; инженерия знаний; онтологическое моделирование; семантическая интероперабельность; классификация; рассуждение; выравнивание онтологий; объяснимый ИИ.

Annotatsiya. Maqolada intellektual tizimlar arxitekturasining fundamental qatlami sifatida ontologiyalar va bilimlarni formalizatsiya qilish ko'rib chiqiladi. Tavsiflash mantiqlari, OWL ontologik profillari va RDF graf modeli kabi nazariy



asoslar, shuningdek, muhandislik amaliyotlari (loyihalash, moslashtirish/alignment, validatsiya, versiyalarni boshqarish) hamda ekspluatatsion jihatlar (ma'lumotlar bilan integratsiya, mulohaza yuritish mexanizmi, sifat monitoringi)ning yaxlit sharhi beriladi.

Kalit so'zlar: ontologiya; bilimlarni formalizatsiya qilish; tavsiflash mantiqlari; RDF; OWL; SPARQL; bilim muhandisligi; ontologik modellashtirish; semantik o'zaro ishlash qobiliyati; klassifikatsiya; mulohaza yuritish; ontologiyalarni moslashtirish (alignment); tushuntiriladigan SI (XAI).

Abstract. The article examines ontologies and knowledge formalization as a fundamental layer in the architecture of intelligent systems. It provides an integrated overview of the theoretical foundations (description logics, OWL ontology profiles, the RDF graph model), engineering practices (design, matching/alignment, validation, version management), and operational aspects (data integration, reasoning engine, quality monitoring).

Keywords: ontology; knowledge formalization; description logics; RDF; OWL; SPARQL; knowledge engineering; ontological modeling; semantic interoperability; classification; reasoning; ontology alignment; explainable AI.

Интеллектуальные системы - от рекомендательных сервисов до клинических помощников - работают не только с данными, но и с семантикой предметной области. Именно поэтому качественное представление знаний становится ключевым фактором надежности и объяснимости решений. Онтологии выступают в роли формальных спецификаций концептов, отношений и ограничений, позволяя машинам рассуждать о сущностях и их связях. В отличие от неформальных словарей онтология стремится к строгой семантике, опираясь на логики описаний и стандарты семантического веба, что позволяет автоматически проверять согласованность и выводить новые факты [3; 4; 9; 10]. В прикладных экосистемах онтологии служат



семантическим «клеем» между гетерогенными системами, обеспечивая повторное использование знаний, устойчивость интеграций и прозрачность жизненного цикла артефактов [1; 6; 7].

Классические определения онтологии подчеркивают её роль как «явной спецификации концептуализации»: онтология фиксирует, какие сущности существуют в рамках предметной области, какие отношения между ними допустимы и какие аксиомы ограничивают множество возможных миров. Практическая ценность проявляется в двойной роли: онтология одновременно служит общим словарем для людей и машин и формальным артефактом, доступным механизмам рассуждения. Такой двойственный характер позволяет использовать онтологию как контракт межсистемного обмена и как средство автоматизированной проверки качества данных. Концептуальные основы и эвристики проектирования, предложенные в ранних работах по инженерии онтологий, остаются актуальными: начинать следует с компетентностных вопросов, нормализации терминов и идентификации ключевых отношений [5].

Формальная семантика онтологий опирается на логики описаний — фрагменты предикатной логики первого порядка с гарантированной разрешимостью ключевых задач. Диалекты OWL2 (EL, QL, RL, DL) предлагают различные компромиссы между выразительностью и трудоемкостью вывода. Например, профиль EL оптимизирован для масштабируемой классификации в больших онтологиях с богатой иерархией классов, QL ориентирован на эффективные запросы к данным, RL поддерживает правилообразные применения, а полный DL обеспечивает максимальную выразительность ценой возросшей сложности [3; 10; 15]. Выбор профиля — инженерное решение, зависящее от сценариев использования и ограничений инфраструктуры.



Графовая модель RDF и схема RDFS обеспечивают базовую типизацию и наследование, делая знания машиночитаемыми и расширяемыми. Этот стек стандартов поддерживается широким набором инструментов: тройственными хранилищами, индексами по предикатам и путям, языком запросов SPARQL с фильтрами и агрегациями. Онтологии в формате OWL сериализуются поверх RDF, а проверка согласованности и классификация выполняются специализированными рассуждателями (HermiT, Pellet, FaCT++), доступными как в виде самостоятельных приложений, так и библиотек (OWL API) [9; 11]. Такое разделение обязанностей - хранение графа, логический вывод, запросы - обеспечивает гибкость архитектуры и масштабируемость при промышленной нагрузке.

Проектирование онтологии начинается с выявления компетентностных вопросов — типичных запросов, на которые система должна отвечать. Эти вопросы направляют выбор концептов и свойств, а также определяют необходимые аксиомы. Полезной практикой является привязка утверждений к источникам происхождения (provenance), что повышает доверие пользователей и облегчает аудит. При сборе терминов следует избегать полисемии и скрытого дублирования, фиксируя словарные статьи с определениями, синонимами, примерами и связями. Верхнеуровневые онтологии (DOLCE, BFO, SUMO) обеспечивают общий каркас, позволяя упростить мэппинг между домашними моделями разных команд и отраслей [4; 6; 13].

Семантическая строгость требует явной фиксации ограничений: домены и диапазоны свойств, функциональность и инверсия, кардинальные ограничения, дизъюнктность классов, транзитивность или симметричность ролей. Эти аксиомы не только документируют требования, но и позволяют автоматически обнаруживать противоречия и скрытые выводы. Например, если класс А дизъюнктен классу В, то индивид не может одновременно



принадлежать обоим; рассуждатель выявит неконсистентные данные, а в ряде случаев подскажет минимальные конфликты (MUS). Настройка объяснимости — генерация деревьев обоснования для каждого вывода — облегчает обратную связь с экспертами домена и улучшает качество базы знаний [3; 9].

Валидация онтологии — многоуровневый процесс. На логическом уровне проверяется согласованность, выполнимость классов, корректность иерархий. На уровне данных проводится проверка соответствия потоков фактов онтологическим ограничениям, обнаружение аномалий кардинальности, «висящих» классов без экземпляров, избыточных эквивалентностей и циклов обобщения. В инженерной практике используются наборы тестов знаний — стабильные запросы и ожидаемые ответы, которые запускаются в конвейере непрерывной интеграции при каждом изменении онтологии. Такой подход позволяет предотвращать регрессии и контролировать эволюцию моделей знаний во времени [7; 12; 13].

Интеграция онтологий с внешними системами требует механизмов сопоставления и выравнивания. Сопоставление (ontology matching) обнаруживает соответствия между классами и свойствами разных онтологий на основе лексических, структурных и статистических признаков; выравнивание (alignment) формализует эти соответствия в виде мостовых аксиом эквивалентности, подтипов или ограничений. Стандартизованные методики и инструменты сопоставления систематизированы в учебниках и монографиях, где подробно анализируются метрики качества и стратегии разрешения конфликтов [12]. В сложных отраслях (медицина, биоинформатика) координация онтологий организуется сообществами и фондами, обеспечивающими единые правила развития и совместимости [8].

С точки зрения вычислений, рассуждение в онтологиях сводится к задачам проверки моделей и классификации. Табличные алгоритмы (tableaux) поэтапно разворачивают аксиомы до тех пор, пока не будет построена модель



или выявлено противоречие. Важную роль играют эвристики упорядочивания, индексирование и инкрементальные алгоритмы, позволяющие перерассуждать только измененные части онтологии. Для запросов к данным применяются оптимизации на уровне планов SPARQL, включая выбор селективных шаблонов, джойны с учётом кардинальностей и кэширование промежуточных результатов. Композиция рассуждения и запросов — одна из причин высокой эффективности современных платформ семантического веба [9; 11].

Тесная связка онтологий с машинным обучением стала характерной чертой современных интеллектуальных систем. Модели извлечения информации из текстов и изображений порождают кандидатов фактов и терминов, а онтология обеспечивает схему и ограничения для их валидации. Семантические эмбединги графов знаний используются для предсказания пропущенных связей; логические аксиомы добавляются к функции потерь как штрафы, что стимулирует согласованность предсказаний с онтологией (approaches to neuro-symbolic learning). Такой симбиоз объединяет статистическую гибкость с формальной верифицируемостью, повышая доверие к результатам в критичных доменах [6; 9; 13; 14].

Экосистема инструментов включает онтологические редакторы (Protégé), серверные библиотеки (OWL API) и триплетные хранилища промышленного класса. Protégé предоставляет визуальное моделирование классов, ролей и аксиом, запуск классификаторов и плагины для шаблонов проектирования, аннотаций и импорта. OWL API даёт программный доступ к операциям над онтологиями, генерации аксиом и интеграции с рассуждателями [11]. На уровне хранилищ поддерживаются версирование, named graphs для разделения контекстов, политики доступа и механизмы указания происхождения (W3C PROV). Это делает онтологии первоклассными артефактами в конвейерах данных и аналитики.



Ключевым организационным вопросом является управление изменениями. Онтология — живой артефакт, и её эволюция должна сопровождаться миграцией данных и переговорами интерфейсов. Модульность позволяет выделять стабильные базовые слои и прикладные расширения, минимизируя область влияния локальных обновлений. Практика рекомендует выпускать онтологические версии с журналом изменений, поддерживать обратную совместимость через мостовые аксиомы и обеспечивать автоматический контроль регрессий тестами знаний и проверками согласованности [4; 8; 12].

Онтологический подход существенно влияет на объяснимость. В отличие от скрытых признаков пространств, онтология оперирует человеко-читабельными понятиями, а рассуждение образует деревья обоснований, которые могут быть представлены пользователю. Это позволяет объяснять не только факт классификации, но и её причины: какие аксиомы были задействованы, какие ограничения сработали и какие данные послужили исходными. В нормативных отраслях такой механизм объяснений помогает удовлетворить требования аудита и стандартизированных процедур контроля качества [1; 3; 6].

Семантическая интероперабельность базируется на согласовании верхнеуровневых категорий и чётко определенных границах контекстов. В многоотраслевых системах часто возникают конфликты значений терминов: одно и то же слово может использоваться для разных сущностей. Онтологии решают проблему за счет явной типизации, многоязычных аннотаций и механизмов ссылок на авторитетные источники. Контроль согласованности межотраслевых карт соответствия выполняется классификаторами на объединенной онтологии; для анализа качества используются матрицы согласия и объяснения конфликтов [12; 14].



Экономика онтологий проявляется на горизонте жизненного цикла информационных систем. Первоначальные затраты на моделирование и рассуждение компенсируются снижением стоимости интеграции, упрощением миграций и уменьшением технического долга. Онтологии повышают повторное использование знаний и компонентов, позволяя строить библиотеки модулей для типовых отраслевых задач. В результате ускоряется вывод продуктов на рынок и повышается устойчивость архитектур к изменчивости требований, что особенно заметно в компаниях с большим портфелем ИТ-систем [1; 4].

Несмотря на преимущества, онтологии предъявляют строгие требования к качеству моделирования. Типичные антипаттерны включают избыточную эквивалентность классов, неконсистентные домены/диапазоны, нарушения открытого/закрытого мира в одном контексте, смешение уровней абстракции и перегруженные классы. Использование шаблонов проектирования (Ontology Design Patterns) и руководств по инженерии снижает риск ошибок и унифицирует стиль моделирования [14; 7]. Рекомендуется проводить регулярные ревью и использовать автоматические линтеры, анализирующие структуру графа и аксиоматику.

Разработка онтологий тесно связана с открытыми данными и лицензированием. Переиспользование общедоступных онтологий ускоряет старт проекта, однако требует соблюдения условий лицензий и корректного указания авторства. Рекомендуется хранить онтологию вместе с документацией, примерами SPARQL-запросов, описанием уровней доверия к источникам и сценариями развертывания. Такой комплект облегчает передачу проекта и упрощает включение новых команд [8; 9].

Перспективным направлением является синтез онтологий с причинными графами и формальными спецификациями поведения систем. Онтологии задают типы сущностей и инварианты, а причинные модели



обеспечивают контрфактический анализ и прогнозирование эффектов вмешательств. Совместное использование позволяет строить системы, способные отвечать на вопросы «что будет, если» и обосновывать выбор действий. На практике такие гибриды требуют аккуратной формализации межуровневых связей и согласования семантики между описательными и причинными слоями [6; 9].

В индустриальных пайплайнах онтологии становятся центром управления метаданными: от каталогов данных и словарей бизнес-терминов до политик доступа и нормативных ограничений. Единое семантическое пространство облегчает поиск и повторное использование наборов данных, выстраивает связи между источниками и отчетами, а также позволяет внедрять автоматические проверки корректности. В сочетании с механизмами lineage это дает сквозную трассируемость результатов аналитики и ИИ, что важно для соблюдения регуляторных требований [1; 9; 10].

Важной практикой является документирование компетентностных вопросов и эталонных сценариев использования. Эти сценарии служат якорями при развитии онтологии: каждое изменение проверяется на отсутствие регрессий в ответах системы. Совместно с этим рекомендуется поддерживать набор негативных тестов — вопросов, на которые система не должна отвечать утвердительно. Такой набор помогает выявлять чрезмерную общность или случайные последствия новых аксиом [7; 12; 13].

Системы рассуждения развиваются: помимо классических tableaux применяются SAT/SMT-редукции, Datalog-подмножества для запросов к графам и гибридные движки, объединяющие правила и онтологии. Выбор механизма влияет на требования к моделированию: например, профили RL и правила позволяют эффективно решать оперативные задачи, тогда как DL-полнота оправдана для глубокой валидации и сложных таксономий.



Современные платформы реализуют адаптивные стратегии, переключая уровни выразительности в зависимости от запроса [3; 10; 15].

На уровне пользовательского взаимодействия онтологии обеспечивают подсказки при формировании запросов и проверку согласованности вводимых данных. Формальные определения терминов и многоязычные аннотации снижают когнитивную нагрузку и уменьшают количество ошибок. Интеграция онтологии с интерфейсом обеспечивает автодополнение и предотвращение недопустимых комбинаций значений, повышая качество данных на входе и сокращая затраты на последующую очистку.

В условиях данных, подверженных дрейфу, онтологии играют роль стабилизатора смысла. Тогда как статистические распределения признаков меняются, семантические инварианты остаются, и система может обнаруживать несоответствия между ожиданиями онтологии и текущими потоками. Мониторинг таких расхождений и автоматическое создание тикетов на проверку источников — важные элементы эксплуатации онтологической инфраструктуры [1; 9].

Таблица 1.

Типы онтологий и области применения

Тип онтологии	Глубина / выразительность	Профиль OWL / логика	Сильные стороны	Ограничения	Типичные применения
Легковесная таксономия	Низкая	RDFS, OWL 2 RL	Простота, быстрый вывод	Ограниченная семантика	Каталоги, навигация
Доменная онтология	Средняя	OWL 2 EL/QL	Скалируемая классификация	Меньше выразительности	Промышленные словари



Функциональная/прикладная	Средняя+	OWL 2 DL	Богатые ограничения	Тяжелее классификация	Правила бизнес-логики
Верхнеуровневая (upper)	Абстрактная	OWL 2 DL	Унификация терминов	Кривая обучения	Интероперабельность
Гибрид с правилами	Переменная	OWL + SWRL	Выразительность	Риск неразрешимости	Комплаенс, трансформации

Отдельной задачей является оценка стоимости вывода. В реальных системах требуется предсказуемость времени отклика. Для этого используются стратегии усечения доказательств, ограничение глубины правилами, шардирование графа по областям ответственности и кэширование промежуточных результатов. Инженерная дисциплина предполагает SLO для рассуждения и регулярные профилировки тяжелых аксиом, чтобы не допустить деградации сервиса [3; 11].

Наконец, образование и стандартизация практик играют ключевую роль в распространении онтологий. Учебники по семантическим технологиям и инженерии онтологий предлагают методологии, шаблоны и антишаблоны, которые ускоряют обучение новых специалистов и повышают качество проектов [9; 12; 13; 14]. Комьюнити и открытые репозитории примеров способствуют переиспользованию решений и обмену опытом между отраслями.



Рис.1. Конвейер работы с онтологиями

Подводя итог, онтологии и формальная инженерия знаний формируют семантический фундамент интеллектуальных систем. Их сила - в точных определениях и возможности алгоритмической проверки, в объяснимости и переносимости моделей, в долговечности знаний и снижении интеграционных затрат. В сочетании с машинным обучением онтологии задают правила игры для продвинутых ИИ, делая их прозрачными и надежными. Именно поэтому



инвестиции в онтологическое проектирование и инфраструктуру рассуждения – это инвестиции в устойчивую эволюцию ИТ-ландшафтов [1; 3; 4; 6; 9; 12; 15].

В заключение можно сказать, что онтологии и формальная инженерия знаний образуют семантический каркас интеллектуальных систем. Они позволяют задавать явные определения, ограничения и инварианты, обеспечивая автоматическую проверку согласованности, объяснимость выводов и устойчивую интеграцию гетерогенных источников. Практика подтверждает, что зрелые процессы проектирования, валидации и управления изменениями снижают риски и технический долг, а также повышают повторное использование знаний. Вектор дальнейших исследований — нейросимвольные методы, масштабируемые алгоритмы выравнивания и причинно-онтологические гибриды, способные отвечать на контрфактические вопросы и поддерживать принятие решений в критичных доменах.

Список литературы:

1. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем. — СПб.: Питер, 2000. — 384 с.
2. Журавлёв Ю. И., Кириченко А. В. Логико-алгоритмические основы искусственного интеллекта. — М.: Наука, 2016. — 328 с.
3. Baader F., Calvanese D., McGuinness D., Nardi D., Patel-Schneider P. (eds.). The Description Logic Handbook: Theory, Implementation, and Applications. — 2nd ed. — Cambridge: Cambridge University Press, 2017. — 648 p.
4. Staab S., Studer R. (eds.). Handbook on Ontologies. — 2nd ed. — Berlin: Springer, 2009. — 829 p.
5. Gruber T. R. A Translation Approach to Portable Ontology Specifications // Knowledge Acquisition. — 1993. — Vol. 5, No. 2. — P. 199–220.



6. Guarino N., Oberle D., Staab S. What Is an Ontology? // Handbook on Ontologies. — Springer, 2009. — P. 1–17.
7. Noy N. F., McGuinness D. L. Ontology Development 101: A Guide to Creating Your First Ontology. — Stanford KSL Technical Report, 2001. — 25 p.
8. Smith B., Ashburner M., Rosse C., et al. The OBO Foundry: Coordinated Evolution of Ontologies to Support Biomedical Data Integration // Nature Biotechnology. — 2007. — Vol. 25. — P. 1251–1255.
9. Hitzler P., Krötzsch M., Rudolph S. Foundations of Semantic Web Technologies. — Boca Raton: CRC Press, 2010. — 403 p.
10. OWL 2 Web Ontology Language: Document Overview. — W3C Recommendation, 11 December 2012. — 32 p.
11. Horridge M., Bechhofer S. The OWL API: A Java API for OWL Ontologies // Semantic Web. — 2011. — Vol. 2, No. 1. — P. 11–21.
12. Euzenat J., Shvaiko P. Ontology Matching. — 2nd ed. — Berlin: Springer, 2013. — 570 p.
13. Keet C. M. An Introduction to Ontology Engineering. — 1st ed. — University of Cape Town, 2018. — 300 p.
14. Gangemi A., Presutti V. Ontology Design Patterns // Handbook on Ontologies. — Springer, 2009. — P. 221–243.
15. Motik B., Grau B. C., Horrocks I., Wu Z., Fokoue A., Lutz C. OWL 2 Web Ontology Language Profiles. — W3C Recommendation, 11 December 2012. — 56 p.