

УДК 004.82

## ПРОЕКТИРОВАНИЕ ОНТОЛОГИИ ДЛЯ ПРАВОВОЙ ПРЕДМЕТНОЙ ОБЛАСТИ НА ОСНОВЕ ТЕКСТОВОГО КОНТЕНТА С ИСПОЛЬЗОВАНИЕМ НЕЧЁТКОЙ ЛОГИКИ

Ю.Р. Валькман<sup>1</sup>, Е.А. Хала<sup>2</sup>

Международный научно-учебный центр информационных технологий и систем  
НАН и МОН Украины, Киев, Украина  
<sup>1</sup>yur@valkman.kiev.ua, <sup>2</sup>cecerongreat@ukr.net

### Аннотация

Статья посвящена проблемам онтологического инжиниринга в правовой предметной области. Исследуются особенности построения правовой онтологии и проблемы реализации этой онтологии. Системы, которые обеспечивают обмен знаниями между существующими юридическими системами, должны поддерживать перевод юридических баз знаний, представленных в различных форматах и формах. Формализм представления этих знаний должен обеспечивать их совместимость в рамках более крупной архитектуры для разработки юридических систем знаний. Предлагается метод введения нечёткой логики в правовые онтологии. Описаны две возможности, посредством которых эксперт может добавить знание в онтологии. Предлагаемые решения позволяют представлять неточную информацию в системах и реализовать нечёткие рассуждения.

**Ключевые слова:** нечёткий вывод, нечёткая логика, структура онтологии, анализ текста, текстовый контент, представление знаний, онтология, Protege, базовая онтология, право.

### Введение

Правовая область знаний является достаточно сложной по сравнению с другими предметными областями (ПрО), потому что включает знания о физических и социальных мирах, а также типовые правовые знания, которые фактически создают новый слой представления знаний. Для моделирования и описания правовых явлений и процессов актуальна задача создания систем, основанных на знаниях. Такие системы, включающие в себя элементы экспертной обработки и интеллектуального поиска, могут стать незаменимым инструментом в правотворческой и правоприменительной деятельности, стать основой для различных семантических приложений. Кроме того, подобные системы можно применять для решения учебных задач в юриспруденции.

Первым шагом в разработке такой системы является построение и принятие *онтологии* – в роли единого источника документированной совокупности терминов ПрО и правил, согласно которым термины могут быть использованы для построения достоверных утверждений о состоянии системы, а также для логического вывода новых утверждений. На основе онтологий пользователи и системы могут общаться друг с другом, поддерживая информационный обмен, базирующийся на семантике этой единой терминологической базы.

Понятие онтологии является достаточно широким по степени формализации. Так, к онтологиям можно отнести даже неструктурированный словарь с определениями. Поскольку в текстах правовых актов достаточно широко практикуется введение определений (например, в преамбулах некоторых законов), построить словарь терминов конкретной отрасли права не так сложно, но такая простая онтология не будет эффективна в использовании.

В.Д. Соловьев и В.В. Иванов [1] приводят перечень свойств, которые характерны только для ресурсов онтологического характера. Рассмотрим эти свойства.

- *Формальность* - точное определение иерархического отношения между классами (is-a), которое строго соблюдает транзитивность этого отношения.
- *Отношения класс-экземпляр* – такие онтологии могут содержать, помимо имен классов, экземпляры (индивиды) на нижнем уровне.
- *Свойства концептов* – особенно полезны, когда они определены на верхних уровнях и наследуются подклассами.
- *Ограничения на область значений свойств* – дополнительные ограничения на то, экземпляром какого класса может являться значение данного свойства.
- *Дизъюнктивные классы, обратные свойства* – классы, заведомо не имеющие общих экземпляров. Обратные свойства представляют собой пару свойств у двух разных классов, которые определяют отношения между их экземплярами.
- *Аксиомы* – позволяют создавать произвольные логические ограничения.

Очевидно, что эти свойства по мере использования расширяют возможности применения онтологий для моделирования различных процессов и явлений и делают их уникальным инструментом именно в социально-правовой ПрО.

Широко известно обоснование потребности в разработке и использовании онтологий [2]:

- совместное использование людьми или программными агентами общего понимания структуры информации;
- возможность повторного использования знаний в ПрО;
- создание явных допущений в ПрО;
- отделение знаний в ПрО от оперативных знаний;
- возможности объективного анализа знаний в ПрО.

Все эти аспекты применения онтологических структур находят приложения в анализе правовых ситуаций, исследовании юридических процессов, принятии решений в правовой практике. Данное направление начало интенсивно развиваться в последние десятилетия, и одна из методологий уже закреплена в соответствующих стандартах семейства IDEF (ICAM/Integrated Definition), которые создавались в США в рамках программы компьютеризации промышленности ICAM.

IDEF5 – Ontology Description Capture – стандарт онтологического исследования сложных систем. С помощью методологии IDEF5 онтология системы может быть описана при помощи определённого словаря терминов и правил, на основании которых могут быть сформированы достоверные утверждения о состоянии рассматриваемой системы в некоторый момент времени. На основе этих утверждений формируются выводы о дальнейшем развитии системы, и производится её оптимизация [3]. Помимо IDEF5 среди известных методологий в области онтологического инжиниринга можно также перечислить такие, как METHONTOLOGY [4], DILIGENT [5], NeOn [6].

## 1 Информационный поиск текстовых документов

### 1.1 Задачи онтологического инжиниринга в правовой сфере

В последнее время наблюдается значительный рост публикаций, посвящённых проблемам разработки и использования онтологий в самых различных приложениях. Проектирование онтологий в правовой сфере имеет следующие характерные черты [7]:

- 1) большое количество общепринятых понятий с особой спецификой использования;
- 2) существенное различие в структурах разных отраслей права;
- 3) наличие промежуточного общетеоретического правового уровня между онтологией верхнего уровня и онтологией ПрО;

- 4) большое количество теоретических допущений и абстрактных конструкций, зависящих от специфики правовых взглядов.

В правовой области используется большой документооборот (информационных структур, представленных естественно языковыми данными – ЕЯ), что усложняет не только поиск необходимых сведений, но и документов в целом. В информационном поиске рассматривается проблема поиска нужной информации из больших источников данных, таких как World Wide Web, интранет и электронные библиотеки. Информационные подходы извлечения знаний из корпусов текстов часто используют ключевые фразы для индексирования и поиска документов. Например, методология извлечения ключевых фраз документа, а затем вычисление частоты и получение соотношения между фразами [7].

При обработке ЕЯ-текстов общий подход заключается в анализе текстов с использованием грамматики и семантики. Компьютерные программы декомпозируют ЕЯ-текст на предложения, используя правила грамматики. Однако, определение значения предложения (его смысла) является весьма сложной проблемой.

Совокупность (или структура) знаний ПрО обычно представляется объектами, понятиями, сущностями и отношениями между ними. Всемирная паутина постоянно расширяет объём знаний, которые требуют чёткой структуры, то есть онтологии, чтобы описать их и сделать доступным для использования. Для этого был разработан специальный язык Resource Description Framework (RDF), предназначенный для описания метаданных о веб-ресурсах и для формирования онтологии (см., например, [1]). RDF — это разработанная консорциумом Всемирной паутины модель для представления данных является частью концепции семантической паутины. Основным элементом RDF является тройка: ресурс (*субъект*), связанный с другим ресурсом (*объект*), через дугу, маркированную ресурсом (*предикат*). Это означает, что «субъект» имеет свойство «предикат» со значением «объект».

Пользователи онтологии могут легко делиться знаниями через RDF, и некоторые исследователи используют онтологии в качестве подхода для повышения эффективности методов анализа текста. Палмер [8] предложил алгоритм, основанный на анализе расстояния с вычислением ценности подобия парных ключевых слов в онтологии. Так же был разработан алгоритм, который автоматически генерирует онтологию и классифицирует информацию с использованием нечётких нейронных сетей, и методология классификации документов, использующая не только автоматически построенную онтологию, но и частоты ключевого понятия документа для классификации. Нечёткая логика обеспечивает исследователей средствами для подражания правилам классификации экспертов.

В вычислительной среде знания представлены и хранятся с использованием специальных языков. Эти языки ориентированы на создание жёстких формальных структур. Знания же, представленные на ЕЯ, используют лингвистические конструкции, которые не могут быть чёткими, однозначными, непротиворечивыми. Поэтому в таких приложениях необходимо использовать средства нечёткой математики и другие методы мягких вычислений.

Кластеризация – общий метод для построения множеств, которые являются гомогенными в пределах групп, но гетерогенными между группами. Математический принцип объединения в кластеры максимизирует различие между группами и минимизирует различие в пределах групп. Методы кластеризации были успешно применены к текстовой обработке. Ранклер и Бездек сгруппировали тексты веб-страниц и последовательности веб-страниц, которые посещают пользователи. Другой пример кластеризации для анализа и синтеза текста, продемонстрировал Сюем, который использовал подход К-средних для того, чтобы сгруппировать доступные документы в кластеры. Фразы, извлечённые из документов, часто используются, чтобы установить отношения подобия между текстами документов. Эти отношения подобия используются в качестве основы для группирования документов в кластеры. Однако стати-

стический анализ ключевых фраз не может являться удовлетворительной основой для ЕЯ-кластеризации знаний в правовой ПрО. Поэтому представляется целесообразным анализировать и группировать документы, используя схему онтологии заданной области, а не подход имитации текста ключевой фразой. Для такой методологии необходимо, чтобы эксперты сначала построили онтологическую схему, то есть структуру знаний для ПрО, и затем обучили систему, используя заданный набор шаблонов. Здесь обработка ЕЯ применяется для построения онтологии доступных документов, а нечёткая логика – для вычисления онтологического сходства между документами для их группировки в различные кластеры.

## 1.2 Метод

Первоначально эксперты в ПрО определяют онтологию, используя базу знаний, инструменты редактирования RDF, язык Protege [9]. Затем слова и фразы (например, фрагменты текстов и аннотации) сопоставляются с соответствующими концептами онтологии области. Далее эксперты создают учебный набор шаблонов, используя простую и удобную в работе обработку ЕЯ с помощью инструмента под названием MontyLingua [10]. После этого вычисляются вероятности концептов в заданных фрагментах документа. Вероятности концепта рассчитываются в каждом конкретном документе, а затем используются для группировки шаблонов с нечёткими логическими выводами. Таким образом, иерархический алгоритм кластеризации уточняется, адаптируя нечёткую логику к процессу вывода понятия.

Введём некоторые определения используемых далее понятий.

*Шаблон* – повторяемая архитектурная конструкция, представляющая собой решение проблемы в рамках некоторого часто возникающего контекста.

*Корпус* – совокупность текстов, отобранных в соответствии с некоторой исследовательской задачей и специально подготовленных, размеченных, структурированных, представленных в унифицированном формате.

*Блок* – фрагмент документа, выделенный системой MontyLingua в качестве минимальной единицы хранения знаний для операций аналитической обработки.

Другие термины заимствованы из документации MontyLingua.

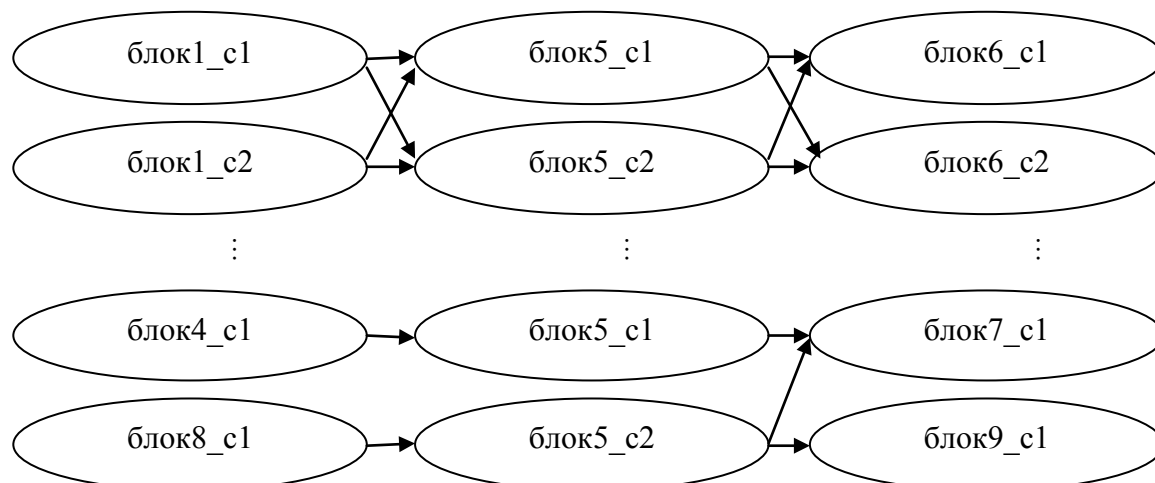
**Проектирование онтологии.** На первом шаге метода используется инструмент редактирования онтологий Protege. Этот редактор помогает экспертам ПрО построить онтологию, используя графическое редактирование [8]. Благодаря этим особенностям Protege считается подходящим инструментом для построения и редактирования онтологии. Онтология может быть автоматически преобразована в стандартные форматы данных (XML, RDF или OWL) для дальнейшего использования и интерпретации, анализа знаний и их синтеза.

**Обработка ЕЯ-текстов и обучение терминологии.** Чтобы измерить знание, содержащееся в документах относительно определённой структуры онтологии, система обучается, используя ряд доступных документов. Предложения из учебных документов помечаются для извлечения частей речи, фрагментов и аннотаций, с помощью инструмента обработки ЕЯ-текстов MontyLingua. Далее в таблице сопоставляются извлечённые слова с соответствующими концептами из онтологии. Система формирует запись вероятностей концептов, определяя, какое слово подразумевается в шаблоне. Условная вероятность  $P$  (явный концепт | слово  $C$  в блоке  $B$  корпусов), рассчитывается во время учебной сессии.

Чтобы поддержать полноту слов-понятий системы, процесс исследования включает повторяющийся механизм «переучивания». Для включения новых слов, которые не входят в текущую базу терминологии, новый термин сначала сохраняется в базе данных терминологии. После этого системный администратор определяет соответствующий онтологический концепт для данного термина. Это позволяет системе автоматически повторно вычислить и обновить онтологическую терминологию понятий базы знаний.

**Анализатор терминологии.** После обработки ЕЯ-текстов и обучения терминологии все выведенные концепты предложений являются случайной величиной, т.е. вероятности концептов для каждого блока вычислены.

**Извлечение знаний.** После анализа терминологии вычисляются вероятности концептов для каждого блока. Блоки, подразумевающие концепты как предикаты, первыми входят в онтологию. На рисунке 1 показано, что блок 5 подразумевает два концепта (кандидата) как предикаты в онтологии.



блок<sub>і</sub>\_с<sub>ј</sub> : понятие j, что блок i подразумевает.

Рисунок 1 – Фильтрация значений онтологии

Следующий шаг должен выбрать блоки, которые подразумевают концепты, как субъекты в онтологии от предыдущего предложения до следующего предложения. Поэтому концепты, которые подразумевает блок 1, блок 4 и блок 8, являются кандидатами на субъект. Тот же самый процесс используется, чтобы определить кандидатов объекта (блок 6, блок 7 и блок 9).

Если есть десять кандидатов на предмет, два кандидата на предикат и десять кандидатов на объект, то число заявленных кандидатов составит 200 (10×2×10). Запросы, которые не существуют в онтологии, устраняются. Наконец, выход генерируется с использованием вероятности, полученной из следующего выражения:

$$(1) \text{Max}_{\text{для\_всех\_значений\_блока\_5}} \times \frac{\text{кандидат(субъект)} + \text{кандидат(предикат)} + \text{кандидат(объект)}}{3}$$

Процесс, описанный выше, используется для блоков, которые содержат концепты предиката в документе-онтологии. Таким образом, документ превращается в набор операторов в онтологии. Эти запросы рассматриваются как индексы документа и являются основой сходства при сравнении с другими документами.

**Подобные совпадения.** Для того, чтобы вычислить подобие между доступными документами, нечёткая логика используется для получения меры сходства. Содержание доступных документов разделено на набор главных понятий и набор деталей. Их сочетания и выводы представлены в таблице 1.

Перед введением в модель вывода, документы переводятся в онтологический формат с включением главных понятий и их деталей. Главные понятия состоят из «верхних троек», а детали состоят из «нижних троек»

$$(2) \quad s = \frac{TT}{ST},$$



где  $S$  - мера сходства документа 1 и документа 2;  $TT$  - схожие тройки в документе 1 и в документе 2;  $ST$  - суммы троек в документе 1 и в документе 2.

Нечёткие логические представления «многие совпадения», «некоторые совпадения» и «немногие совпадения» определены функциями принадлежности [12, 13].

Таблица 1 – Нечёткие правила для вывода подобия документов

№	Если два документа, состоящие из предложений (полученные в ответ на запрос из онтологии) с ...	то, полнота подобия этих двух документов...
1	Множеством совпадений главных понятий и Множеством совпадений подробных описаний	Высокая
2	Множеством совпадений главных понятий и Некоторыми совпадениями подробных описаний	Высокая
3	Множеством совпадений главных понятий и Несколькими совпадениями подробных описаний	Средняя
4	Некоторыми совпадениями главных понятий и Множеством совпадений подробных описаний	Высокая
5	Некоторыми совпадениями главных понятий и Некоторыми совпадениями подробных описаний	Средняя
6	Некоторыми совпадениями главных понятий и Несколькими совпадениями подробных описаний	Средняя
7	Несколькими совпадениями главных понятий и Множеством совпадений подробных описаний	Средняя
8	Несколькими совпадениями главных понятий и Некоторыми совпадениями подробных описаний	Низкая
9	Несколькими совпадениями главных понятий и Несколькими совпадениями подробных описаний	Низкая

Нечёткая модель вывода применяет преимущества «если-то»-правил в условиях нечёткости входа и выхода. Очевидность формулировки модели, простого вычисления и ясности в представлении традиционной лингвистики поддерживает выбор этого подхода. Таким образом, нечёткая модель вывода Мамдани использует «мин-мин-макс»-операцию, рассматривая два правила: «принят» и «изменен». Изначальная «мин-мин-макс»-операция Мамдани [13] рассматривает подход с двумя правилами, но рассматриваемая версия манипулирует девятью правилами одновременно. Этапы этой процедуры заключаются в следующем.

- 1) Вычисляют сходство документов, удовлетворяющих основные понятия ( $X_{mc}$ ) и сходство документов, удовлетворяющих подробные описания ( $X_{dd}$ ).
- 2) Определяют оценки  $X_{mc}$  и  $X_{dd}$ , используя правила из таблицы 1, чтобы получить соответствующую принадлежность.
- 3) Сравнивают принадлежности и выбирают минимальное значение из двух наборов, соответствующего понятия (высокое сходство, среднее сходство и низкое сходство) для каждого правила.
- 4) Собирают принадлежности, которые представляют одно и то же самое понятие в одном наборе.
- 5) Получают максимальную принадлежность для каждого набора и вычисляют результат вывода.

**Дефазификация и объединение в кластеры.** Процедуры вывода генерируют принадлежность в представлении различных уровней сходства. Однако, эти значения - все ещё нечёткие, и они требуют дефазификации, чтобы помочь сгенерировать значения, представляющие сходство документов. Процессы дефазификации состоят из двух шагов. Первый этап должен решить, какое сходство («высокое сходство», «среднее сходство», и «низкое сходство») лучше всего представляет отношения между этими двумя документами. На вто-

ром этапе основное внимание уделяется преобразованию значения принадлежности из сходства. Подробное преобразование значения сходства зафиксировано в следующих трех случаях.

*Случай 1 – высокое сходство ( $U_H > U_L$  и  $U_H > U_M$ ).* Если значение, вычисленное в вышеупомянутой процедуре (нечёткого вывода Мамдани), происходит от понятия «высокого сходства», следующее выражение используется для определения сходства значений документов  $i$  и  $j$ :

$$(3) \quad r_{ij}(U_H) = \left\{ \frac{2+U_H}{3} \right\},$$

где  $U_H$  – значение функции принадлежности для переменной «высокое сходство»;  $U_M$  – то же для «среднее сходство»;  $U_L$  – тоже для «низкое сходство»,  $0 \leq U_H, U_M, U_L \leq 1$ .

*Случай 2 – среднее сходство ( $U_M > U_H$  и  $U_M > U_L$ ).* Если значение, рассчитанное по вышеупомянутой процедуре, исходит от «среднего сходства», приводимое далее выражение используется, чтобы определить значение сходства. При определении значения сходства от «среднего сходства» взаимосвязь между «высоким сходством» и «низким сходством» влияет на сдвиг значения дефаззификации. В результате используются три выражения для соответствия различным отношениям между «высоким сходством» и «низким сходством».

$$(4) \quad r_{ij}(U_M) = \begin{cases} \frac{2+U_M}{6}, & U_L > U_H \\ \frac{4-U_M}{6}, & U_H > U_L \\ \frac{1}{3}, & U_L = U_H \end{cases}$$

*Случай 3 – низкое сходство ( $U_L > U_H$  и  $U_L > U_M$ ).* Если значение, рассчитанное согласно вышеупомянутой процедуре, происходит от понятия «низкого сходства», используется следующее выражение:

$$(5) \quad r_{ij}(U_L) = \left\{ \frac{1-U_L}{3} \right\}.$$

После того, как все меры сходства вычислены, генерируется матрица подобия. Затем используется иерархический алгоритм кластеризации, чтобы последовательно искать различные кластеры в соответствии с различной степенью связи между объектами, как выражено в матрице

$$(6) \quad \begin{bmatrix} 1 & \cdots & r_{1j} \\ \vdots & \ddots & \vdots \\ r_{i1} & \cdots & 1 \end{bmatrix},$$

где  $r_{ij}$  – сходство документа  $i$  и документа  $j$  (следовательно, значение  $r_{ij}$  равно  $r_{ji}$ ).

Применение иерархического алгоритма кластеризации заключается в следующем.

- 1) найти максимальное ( $r_{ij}$ ) в матрице и сгруппировать документы  $i$  и  $j$  в новую группу.
- 2) вычислить отношение между новыми кластерами и другими документами при помощи метода средней связи.
- 3) перейти к шагу 1) до тех пор, пока не останется один кластер слева.

## 2 Нечёткое значение и обработка неопределённых данных

Концептуального формализма стандартной онтологии может быть недостаточно для представления неточной информации. Это часто встречается во многих прикладных областях. Например, ключевые слова, полученные многократными запросами в одной и той же

области нельзя рассматривать как базу, так как некоторые ключевые слова могут быть более значимыми, чем другие. Таким образом, возникает необходимость в различной интерпретации ключевых слов в зависимости от контекста.

Одно из возможных решений обработки неопределённых данных заключается в обеспечении включения нечёткой логики в онтологию [14]. Цель использования теории нечётких множеств состоит в описании неопределённого понятия через обобщённое понятие множества. Например, семантическое содержание заявления, такого как «*Правовые отношения – одна из важнейших и конкретных форм социального бытия*», может иметь степень, или значение истинности, 0,6. В настоящее время нечёткие множества и онтологии совместно используются для устранения неопределённой информации в различных областях. Активно ведутся исследования применения теории нечёткой логики в онтологиях. Нечёткая логика интегрируется в онтологию для того, чтобы получить расширенную онтологию, которая больше подходит для решения проблем неопределённости рассуждений.

## 2.1 Нечёткая логика и определение нечёткого значения

Здесь представлена логическая основа работы с неопределённостью. Это представляет интерес для всех приложений онтологии, в которых пользователь использует информацию, которая содержит неточное и расплывчатое описание понятий. Например, могут представлять интерес рассуждения с такими понятиями как «дорогой товар», «высокая скорость», и т.д.

Для того чтобы справиться с этими проблемами предлагается подход, основанный на теории нечётких множеств. Идея состоит в расширении редактора онтологии, так чтобы непосредственно обрабатывать неопределённости в процессе построения онтологии с обогащенной ПрО.

Рассмотрим непустое множество объектов  $U$ , называемое универсумом. Нечёткое множество определяется с помощью функции  $f$  на  $U$ ,  $f: U \rightarrow [0, 1]$ . Для  $x \in U$  значение  $f(x)$  указывает принадлежность элемента  $x$  рассматриваемому нечёткому множеству [15, 16].

Первая проблема заключается в том, как назначить нечёткое значение субъекта онтологии. Высокая степень понятности и точности являются противоречивыми целями и на практике одно из двух свойств преобладает над другим. В зависимости от того, какое требование реализуется в первую очередь, нечёткую область моделирования можно разделить на две различные области:

- *лингвистическое нечёткое моделирование* - основной целью здесь является получение нечётких моделей с хорошей интерпретируемостью;
- *точное нечёткое моделирование* - основной целью является получение нечёткой модели с хорошей точностью.

Необходимо быть как можно более объективными к обоим целям, которые стоят перед экспертом: определить точное значение или определить язык.

В первом случае эксперт при создании онтологии определяет функцию  $f$ : (Понятие  $\cup$  Экземпляр)  $\times$  Свойство  $\rightarrow$  Значение свойства  $\times [0,1]$ , т.е.  $f(o, p)$  – это значение, которое свойство  $p$  у понятия или экземпляра  $o$  принимает с соответствующей степенью. Например, в гипотетической онтологии о продуктах,  $f(\text{товар}, \text{цена}) = (\text{дорогой}, 0,8)$  означает, что для свойства «цена», экземпляр товара будет иметь значение «дорогой» со степенью 0,8 [17, 18].

Очевидно, что могут существовать ситуации, в которых нет «Значения свойства», определённых для данного свойства. Например, «У товара есть побочный эффект со значением 0,9» не может быть корректно выражено с помощью только формализма. В этой ситуации, необходимо сопоставление пары (понятия/экземпляр, свойство) значению из  $[0, 1]$ , то есть



$f'$ : (Понятие  $\cup$  Экземпляр)  $\times$  Свойство  $\rightarrow [0,1]$ , и выше упомянутый пример будет выглядеть как  $f'$ (товар, побочный эффект) = 0,9 [5].

Для упрощения обозначений определим функцию  $g$ : (Понятие  $\cup$  Экземпляр)  $\times$  (Свойство  $\cup$  Значение свойства)  $\rightarrow [0, 1]$ . Таким образом, «Товар по дорогой цене со значением 0,8» будет  $g$ (товар, дорогой) = 0,8. Используя такую функцию  $g$ , у эксперта есть шанс выбрать ценность значения с любой точностью, то есть точность предпочтительней интерпретируемости.

С другой стороны, вторая возможность заключается в том, чтобы выбрать в качестве членов значения метку в заданном множестве. Возьмём множество  $L = \{\text{мало, достаточно, умеренно, вполне, очень, полностью}\}$  которое, очевидно, не является исчерпывающим из всех возможных меток, но которое может быть интуитивно изменено.

В этом случае значение  $g(o, p)$  формируется в соответствии с функцией, представленной в таблице 2.

Таблица 2 – Сопоставление нечёткого значения метке

Метка	Значение
мало	0.2
достаточно	0.4
умеренно	0.6
вполне	0.7
очень	0.8
полностью	1.0

Подводя итог, мы даём возможность добавить принадлежность значения в паре (понятия/экземпляр, свойство) двумя различными способами: через точное значение  $v \in [0, 1]$  или выбрав метку в определенном заранее множестве  $L$ . Таким образом, через функцию  $g$  определим новое отношение в онтологии.

Другая возможность состоит в том, чтобы нечётким считать сам объект (понятие или экземпляр). В этом случае можно определить функцию  $h$ : Понятие  $\cup$  Экземпляр  $\rightarrow [0, 1]$ .

Итак, значение принадлежности задаётся с помощью одной из двух функций  $g$  и  $h$  числом в единичном интервале  $[0, 1]$ . Это позволяет приложениям, основанным на нечёткой онтологии, использовать стандарты и хорошо изученные инструменты. Например, для того, чтобы задать два (или более) различных нечётких значения, может использоваться оператор накопления. Самый простой пример  $t$ -норма и  $t$ -конорма, то есть двоичное отображение, которое соответствует семантике с «OR», «AND» операторами. Наиболее известны геделевские норма и конорма, т.е. минимаксные операторы. Учитывая примеры, приведённые выше, можно осуществить вычисление значения истинности выражения «Товар по дорогой цене и имеет побочный эффект». Если известно, что  $f$ (товар, дорогой) = 0,8 и  $f$ (товар, побочный эффект) = 0,9, то  $[f$ (товар, дорогой) и  $f$ (товар, побочный эффект)] =  $\min\{0.8, 0.9\} = 0,8$  [18, 19].

Определение для нечёткой онтологии будет следующим: нечёткая онтология – онтология, расширенная нечёткими значениями, которые задаются двумя функциями:

$$g: (\text{Понятие} \cup \text{Экземпляр}) \times (\text{Свойство} \cup \text{Значение свойства}) \rightarrow [0, 1] \text{ и}$$

$$h: \text{Понятие} \cup \text{Экземпляр} \rightarrow [0, 1].$$

## 2.2 Обновление нечёткого значения

После того, как эксперт разработал нечёткую онтологию, будет неверно предполагать, что она будет эффективна, и что любое нечёткое значение чётко определено и подходит для применения в любой ситуации. Таким образом, необходима разработка механизма измене-

ния нечётких значений для того, чтобы наилучшим образом настроить их на конкретную среду, ситуацию или обобщить.

Предположим, что текущее нечёткое значение  $f$ . И, как результат запроса, оно должно быть обновлено до  $f_{new}$ . Простейший способ - это задать  $f := f_{new}$ . Закономерно предположить, что после того, как некий запрос нечёткого свойства достигнет стабильного значения, будет бесполезно изменять его на  $f_{new}$ , теряя всю историю приобретённых знаний. Решение этой проблемы может заключаться в уменьшении важности  $f_{new}$  при любом изменении:

$$(7) \quad f := f + \frac{f_{new} - f}{Q + 1},$$

где  $Q$  – количество обновлений, выполненных для заданного значения. Понятно, что значение  $Q$  должно храниться в онтологии для любого определения нечёткого значения [20].

Теперь вопрос заключается в том, чтобы вычислить новое нечёткое значение  $f_{new}$ . Маловероятно найти в документе точное определение нечёткого значения, но, как правило, можно найти языковой классификатор. Например, мы не нашли «Отношения – важная форма бытия со значением 0,8». Но смысл этого тезиса заключается в том, что «Отношения – очень важная форма бытия». Для этого рассматривается подход на основе модификаторов понятия.

Модификаторы понятий имеют влияние на изменение нечёткого *Значения свойства*. Зададим ряд лингвистических ограничений, таких как «очень», «более или менее», «едва». Тогда модификатор понятия - система одного или более ограничений, таких как «очень немного» или «очень-очень немного». Любому (лингвистическому) модификатору понятия необходимо сопоставить (числовой) член модификатора.

Член модификатора – это значение  $\beta > 0$ , которое используется в качестве *показателя* для изменения значения функции принадлежности  $f$  как  $f^\beta$ .

В соответствии с их воздействием на нечёткое значение, ограничения можно разделить на две группы: концентрирующие и растягивающие. Влияние концентрирующего модификатора заключается в снижении степени принадлежности значения. Таким образом, в данном случае он должен быть  $\beta > 1$ . Например, для ограничения «очень», назначается  $\beta = 2$ . Таким образом, если мы знаем, что  $g(\text{отношения, важная форма}) = 0,8$ , т.е. «Отношения – важная форма бытия со значением 0,8», то «отношение есть очень важная форма» со значением  $0,8^2 = 0,64$ . Напротив, ограничение типа расширения может повышать значение принадлежности, то есть  $\beta \in (0, 1)$ . Например, если для «достаточно» присваивается  $\beta = 0,4$  и  $g(\text{отношения, важная форма}) = 0,8$ , то у «отношение достаточно важная форма» со значением  $0,8^{0,4} = 0,91$ , интуитивно понятно, что если что-то является «важным», тогда оно в какой-то степени и «достаточно важное». Заметим, что такой подход отличается от подхода Заде, в котором «достаточно», а также другие модификаторы, обрабатываются более сложным образом. Преимущество этого метода заключается в том, что он обеспечивает универсальный и простой способ управления модификаторами понятий, даже если при углублении таким способом в семантику необходимо обращаться к цепям модификатора [17, 20, 21].

Таким образом, модификаторы понятий используются в публикациях обычно для определения нового нечёткого члена с учётом уже существующих. Например, если мы знаем нечёткое значение свойства «важный», можно вывести нечёткое свойство члена «очень важный» простым возведением в степень 2 значения «важный». Вместе с тем, здесь можно говорить о противоположной ситуации. В качестве примера, предположим, что мы знаем из онтологии свойство «важный», а также (как результат запроса), что некий объект «очень важный». Таким образом, из «очень важный» мы должны вывести *новое* свойство «важный» для этого объекта. Так, если в онтологии  $g(o, \text{важный}) = 0,7$ , необходимо увеличить это значение, например  $g(o, \text{важный}) = 0,7^{0,5}$ . Эффект «очень» повышает значение свойства, не уменьшая его величину. Таким образом, можно сказать, что в обычном случае выполняется

вывод: «важный» → «очень важный», в то время как в данной ситуации: «важный и очень важный» → «важный».

Этот случай также распространяется на все другие концепции модификаторов. Таким образом, то, что обычно считают модификатором концентрации, становится растяжением и наоборот.

Необходимо решить следующую проблему: определить, какой модификатор рассматривается (и каково его значение) и определить метод вычисления  $\beta$ , значения для цепей концепции модификаторов.

Выберем множество ограничений  $H = \{\text{очень, гораздо больше, больше, больше или меньше, средне, слегка}\}$ . Это только один из возможных вариантов. Множество  $H$  может быть изменено в зависимости от потребностей, при условии, что оно удовлетворяет следующим двум свойствам.

- $H$  - множество упорядочено, т.е. *очень* < ... < *слегка*, и значение  $\beta$  фиксировано только для меньших и больших элементов соответственно как 0,5 и 2.
- Два подмножества концентрирующих ограничений {более или менее, средне, слегка} или растягивающих ограничений {очень, гораздо больше, более} имеют одинаковую мощность (одинаковое количество элементов).

Эти два условия необходимы для расчета принадлежности модификатора последовательности ограничений. Единственная разница - в задании инверсии ограничения, связанная с использованием модификаторов понятий [21].

Открытым вопросом является то, что множество  $H$  модификаторов понятий не исчерпывает всех нюансов ЕЯ. Многие элементы могут быть добавлены в  $H$ , а это потребует новых алгоритмов для их обработки; поскольку не все существующие модификаторы понятий могут быть полностью заданы или точно разделены на две подгруппы одинаковой мощности.

### 2.3 Примеры применения

Приведём два примера возможного использования нечётких онтологий. Первый основан на нечётком значении связанной (случай, свойства) пары, а второй способ использует понятия с нечётким значением для решения проблемы перегрузки.

*Расширение запросов.* При выполнении запроса на документ обычной практикой является расширение набора значений понятия, которые уже присутствуют в запросе, теми, что могут быть получены из онтологии. Очевидно, что такое расширение запроса возможно за счет родителей и потомков понятия. Возможность использования нечётких онтологий также расширяет запросы.

Чтобы принять решение, в каком случае производить добавление к расширенному запросу, мы должны определить, как лингвистические метки соотносятся с численными значениями. Решение такое же, как раннее описанное, только метки, принадлежащие множеству  $L$ , допускаются в запросах, и они преобразуются в числовые значения в соответствии с таблицей 1. Если  $c$  – понятие,  $p$  – свойство и  $l$  – метка, тогда  $\mu(c, p, l)$  – значение данной метки  $l$  для свойства  $p$  и понятия  $c$ . Теперь рассмотрим все случаи  $i$  понятия  $c$ . Они включаются в расширенный запрос, если и только если:

$$(8) \quad |\mu(c, p, l) - g(i, p)| \leq e,$$

где  $e \in [0,1]$  - уровень доверия.

Очевидно, количество экземпляров, которые будут добавлены к расширенному запросу, зависит от значения  $e$ . Граница случаев  $e = 0$  содержит только те случаи, которые точно соответствуют запросу, и входят в него; при  $e = 1$  все экземпляры включены.

Понятно, что в простейшем случае только одно свойство присутствует в запросе. Если два или более запроса должны быть удовлетворены, необходимо обобщить уравнение (8). Предположим, что в запросе есть  $n$  свойств, ссылающихся на те же понятия, что и экземпляр  $c$ , тогда

$$(9) \quad \frac{\sum_{j=1}^n |\mu(c, p, l) - g(i, p)|}{n} \leq e.$$

Т.е. требуем, чтобы среднее значение расстояний между *Значениями свойств* в запросе и *Значениями свойств* в экземплярах было меньше допустимого  $e$ .

*Перегрузка понятия.* Предположим, что понятие  $c$  присутствует в различных фрагментах онтологии. Цель состоит в том, чтобы дать указания о том, какое место является более значительным по отношению к определённому домену. На первом этапе понятию, которое присутствует в нескольких местах, дается одинаковое во всех местах нечёткое значение, сумма этих значений составляет 1.

Для любого  $c_i$  множество его местных условий  $L_i$ , определяемых родителями и потомками, вычисляются из онтологии. Затем все элементы  $l \in L_i$  ищутся в документах с назначенным на них весом  $w_i^j$ . Допустим, что в рассматриваемом документе есть  $l_j$  вхождений для элемента  $l$ . Тогда для любого понятия  $c_i$  и для любого документа  $d$  вычисляются следующие функции:

$$(10) \quad \mu_d(c_i) = \sum_j l_j w_i^j.$$

Сумма по всем  $n$  документам  $\mu_d$  обозначена как  $\mu$ :

$$\mu(c_i) = \sum_{d=1}^n \mu_d.$$

Затем новое значение принадлежности для понятия  $c_i$  и документа  $d$ :

$$(11) \quad h_{new}(c_i) = \frac{\mu_d(c_i)}{\mu(c_i)}.$$

Эти значения используются для обновления  $h(c_i)$  в соответствии с выражением (7), и они применяются с учетом релевантности, так что значения из более значимых документов обрабатываются в первую очередь и имеют большее влияние, чем следующие.

### 3 Разработка онтологии для правовой области

Онтологическая парадигма поддерживает проектирование системы онтологических моделей и обеспечивает взаимодействие с лингвистической компонентой при разработке лингвистических приложений. Одной из важных задач является сопряжение онтологических и лингвистических моделей представления знаний. Онтологическая модель задаёт способ структурирования знаний, лингвистическая модель определяет способы функционирования языковых единиц в текстах. Другими словами, лингвистическая модель определяет лексические, семантические и синтаксические модели концептуальных единиц, в том числе, онтологических единиц. Это особенно характерно для восходящей стратегии моделирования, но и при нисходящей стратегии конечная онтология в итоге всё равно должна содержать термины, которые используются в правовых актах. Это позволяет относить правовые онтологии скорее к лингвистическим, чем к формальным.

В случае с правовым материалом, особенно при построении отраслевой онтологии, основной текстовый массив совпадает с законодательной базой данной отрасли, т.е. представ-

ляет собой чётко определённый перечень правовых актов, доступ к которым обеспечен всем гражданам.

Несмотря на разнообразие правовых актов и возможные недостатки юридической техники, тексты правовых актов в идеальном случае всегда должны отличаться свойствами, которые облегчают формализацию ПрО. Среди свойств можно выделить следующие:

- нейтральность (отсутствие экспрессивности);
- безличный характер;
- связность;
- последовательность;
- точность мысли;
- наличие большого количества стандартных оборотов;
- сильно выраженная структурированность текста.

Во многих случаях для решения прикладных задач онтологического инжиниринга в правовой сфере удобно использовать некоторую общую базовую юридическую онтологию, которая разрабатывается без учета конкретной решаемой задачи и которая может быть легко расширена. В настоящий момент отсутствует единая общеприменимая правовая онтология, в то время как опыт зарубежных исследователей показывает, что создание подобной онтологии не только возможно, но и необходимо. Такая онтология могла бы быть определена в качестве основы для разработки практически применимых онтологий для решения задач в конкретных отраслях права.

Представляется нецелесообразным проводить всю работу по формализации правовой сферы без использования данных других онтологических проектов. Правовая онтология, которая является частью общей архитектуры для систем правовых знаний, обеспечивающих обмен такими знаниями между существующими системами правовых знаний, - это формат обмена правовыми знаниями LKIF (Legal Knowledge Interchange Format) [22]. LKIF выполняет две основные роли:

- включает перевод между правовыми базами знаний, представленными в различных форматах и формах;
- представляет формализм знаний, который является частью большой архитектуры для разработки правовых систем знаний.

Эти случаи использования LKIF приводят нас к классическому компромиссу между уступчивостью и выразительностью. Дополнительное требование заключается в том, что формат LKIF должен соответствовать текущему Semantic Web стандарту для включения службы правовой информации через сеть из комбинации языков OWL DL и SWRL.

Таким образом, актуальной является задача: на основе базовой юридической онтологии LKIF создать юридическую онтологию верхнего уровня применительно к системе права с использованием русскоязычных концепций.

### 3.1 Реализация онтологической модели

Реализация модели невозможна без выбора соответствующих средств моделирования и определения стратегии разработки.

Выбор средства моделирования не представляет особой сложности: за основу взята «родительская» онтология LKIF [22]. Онтология LKIF не только обеспечивает теоретическое понимание правовой области, но её прагматическое использование - как инструмент, чтобы облегчить приобретение знания, обмена и его представления (в частности, формализовать части существующего законодательства).

Онтология LKIF должна содержать «*фундаментальные понятия закона*». От потенциальных пользователей зависит, какой словарь будет использоваться. Выделено и идентифи-



цировано три главных группы пользователей: *граждане* (непрофессионалы), *юристы* и *правоведы*. И, хотя юристы используют правовой словарь более грамотно и тщательно, чем непрофессионалы, для большинства из используемых терминов по-прежнему существует некоторое общее понимание (и можно рассматривать соответствующие термины как более или менее схожие).

Вместе с тем, множество основных терминов имеет определенное юридическо-техническое значение, например, «*ответственность*» или «*правовой факт*».

В роли стратегии моделирования была выбрана система METHONTOLOGY [4]. METHONTOLOGY, как подход к построению и сопровождению онтологий, разработан Асунсьон Гомез-Перез (A. Gomez-Perez) с коллегами. В рамках этой методологии реализуются принципы Тома Грубера, а также разработан программный комплекс спецификации онтологии ODE (Ontology Design Environment).

Данный подход выделяет следующие процедуры в «жизненном цикле» создания онтологии: управление проектом, собственно разработка и поддержка разработки.

Процедуры управления проектом включают планирование, контроль и гарантии качества. Планирование определяет, какие задачи должны быть выполнены, как они организуются, как много времени и какие ресурсы нужны для их выполнения. Контроль гарантирует, что запланированные задачи выполнены и именно так, как это предполагалось. Гарантии качества нужны для того, чтобы быть уверенным в том, что компоненты и продукт в целом находятся на заданном уровне.

### 3.2 Разработка онтологии правового субъекта

Технология разработки онтологии включает следующие процессы: спецификацию, концептуализацию, формализацию и реализацию. На рисунке 2 приведён пример фрагмента таксономии понятия правового объекта онтологии.

В соответствии с обсуждаемой здесь методологией сначала строится глоссарий терминов, включающий все термины (концепты и их экземпляры, атрибуты, действия и т. п.), важные для ПрО, и их ЕЯ-описания. Фрагмент этого глоссария представлен в таблице 3.

Когда глоссарий терминов достигает «существенного» объёма, строятся деревья классификации концептов. Таким образом, идентифицируются основные таксономии ПрО, а каждая таксономия, согласно рассматриваемой методологии, и представляет собой, в конечном счёте, онтологию.

Следующим шагом является построение диаграмм бинарных отношений. Целью их создания является фиксация отношений между концептами одной или разных онтологий. Заметим, что в дальнейшем эти диаграммы могут послужить исходным материалом для интеграции разных онтологий. Пример фрагмента таксономии понятия правового объекта онтологии представлен на рисунке 2.

Таблица 3 – Фрагмент глоссария терминов онтологии правового объекта

Имя	Синонимы	Акронимы	Описание	Тип
Возраст совершеннолетия	-	-	Возраст совершеннолетия с 18 лет	постоянная
Суд	судебный трибунал	-	Хотя «суд» может быть понят как физическое место или как судья, считаем, что суд - судебный трибунал	концепт
Дата рождения	-	-	Дата, когда человек родился	отношение
Ответчик (человек, судебный процесс)	-	-	Судебный процесс ответчика	Атрибут экземпляра

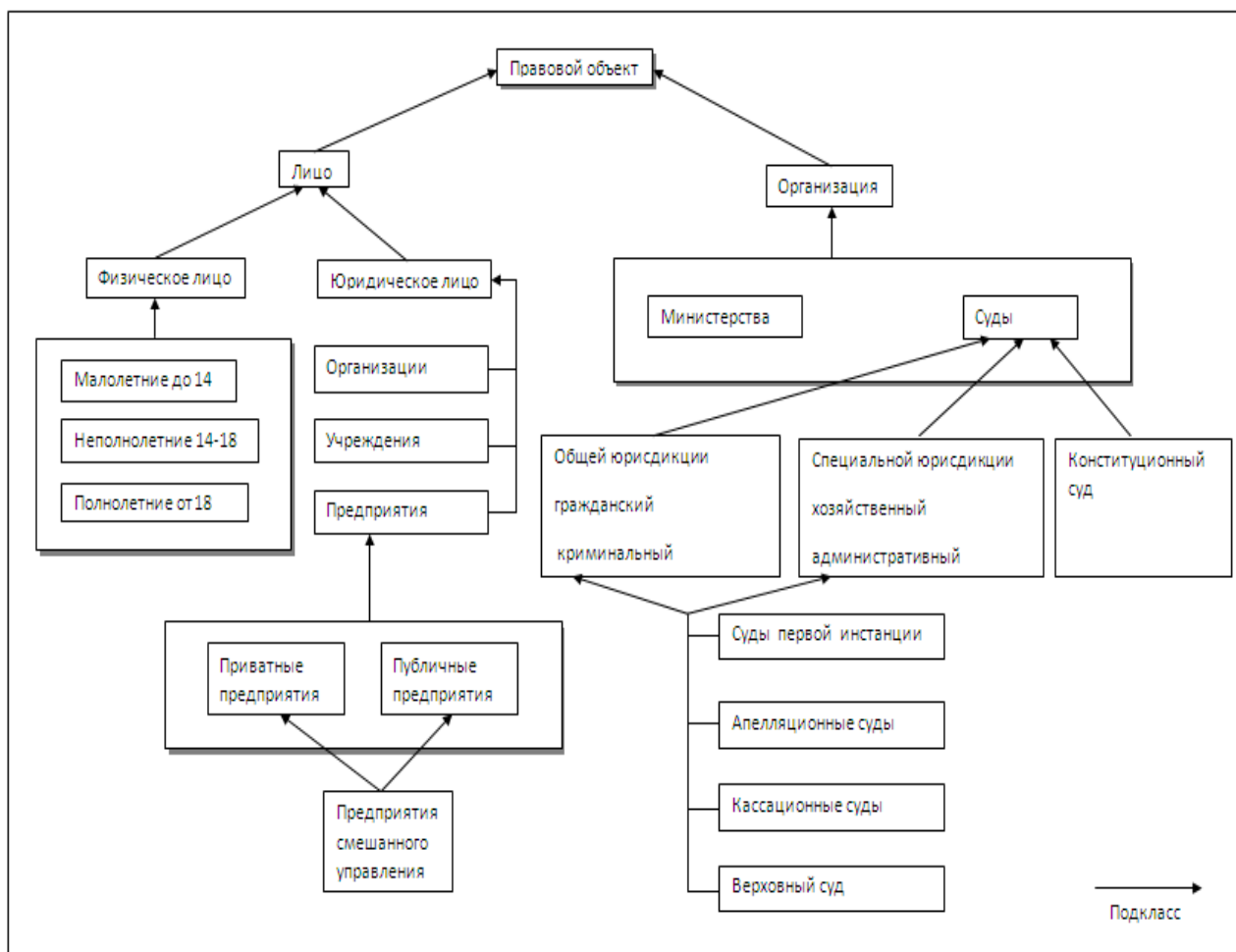


Рисунок 2 – Пример фрагмента таксономии понятия правового объекта онтологии

После фиксации представлений, отмеченных выше, для каждого дерева классификации концептов определяется следующее.

- 1) в *словаре концептов* описываются все концепты ПрО, их экземпляры, атрибуты экземпляров концептов, отношения, источником которых является концепт, а также (опционально) синонимы и акронимы концепта. Фрагмент словаря концептов представлен в таблице 4.
- 2) в *таблице бинарных отношений* для каждого отношения фиксируются его имя, имена концепта-источника и целевого концепта, инверсное отношение. На рисунке 3 представлен фрагмент диаграммы специальных бинарных отношений. В таблице 5 представлен фрагмент этой таблицы.
- 3) в *таблице атрибутов экземпляра* описываются следующие свойства атрибутов: его имя, имя концепта, тип значения, единица измерения, точность, диапазон изменения, значение «по умолчанию», мощность, атрибуты, формула или правило для вывода атрибута (для каждого экземпляра из словаря концептов). Под *мощностью* понимается кардинальное число соответствующего кортежа: минимальное и максимальное число значений. В таблице 6 представлен пример фрагмента отношения этих свойств атрибутов.

- 4) в таблице атрибутов класса определяются: имя атрибута, идентификатор концепта, тип значения, мощность, значение (для каждого класса из словаря концептов с аналогичными характеристиками). Соответствующий пример приведён в таблице 7.

Таблица 4 – Фрагмент словаря концептов онтологии правового объекта

Имя концепта	Экземпляры	Атрибуты класса	Атрибуты экземпляра	Отношения
Суд	Конституционный суд, Гражданский суд, Верховный суд, Административный суд	-	количество членов, место, территориальная юрис- дикция	слушается
Предприятия	-	вид контроля	имя	-
Судебный процесс	-	-	-	имеет ответчика, имеет истца, заслушан
Лицо	-	-	-	ответчик, истец
Физическое лицо	-	-	возраст, дата рождения, дата смерти, первая фамилия, имя, национальность вторая фамилия	мать имеет мать отец имеет отца

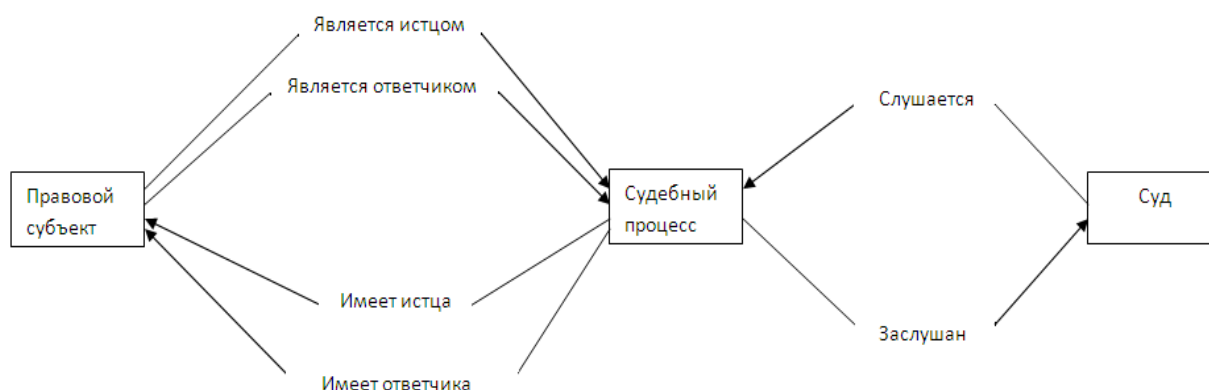


Рисунок 3 – Фрагмент диаграммы специальных бинарных отношений правового объекта

Таблица 5 – Фрагмент бинарного отношения онтологии правового объекта

Имя отношения	Исходный концепт	Исходное количество элементов	Целевой концепт	Обратное отношение
ответчик	Лицо	N	судебный процесс	имеет ответчика
истец	Лицо	N	судебный процесс	имеет истца
слушается	Суд	N	судебный процесс	заслушан
имеет ответчика	Судебный процесс	N	лицо	ответчик
имеет истца	Судебный процесс	N	лицо	истец
заслушан	Судебный процесс	N	суд	слушается

Таблица 6 – Пример фрагмента таблицы атрибутов экземпляра онтологии правового объекта

Имя атрибута экземпляра	Имя концепта	Тип значения	Диапазон значения	Мощность
количество членов	суд	целое	1 ..	(1,1)
место	суд	строка	-	(1,1)
территориальная юрисдикция	суд	строка	-	(1,1)

- 5) в *таблице логических аксиом* вводятся определения концептов через всегда истинные логические выражения. Пример такой аксиомы приведён в таблице 8. Определение каждой аксиомы включает: имя; ЕЯ-описание; концепт, к которому аксиома относится; атрибуты, используемые в аксиоме; логическое выражение, формально описывающее аксиому.
- 6) *таблица констант* включает для каждой константы: её имя, ЕЯ-описание, тип значения, само значение, единицу измерения, атрибуты, которые могут быть выведены с использованием данной константы. В таблице 9 приведён пример кортежа этого отношения.
- 7) в *таблице формул* для каждой формулы, включённой в таблицу атрибутов экземпляра (помимо собственно формулы), определяются её имя, атрибут, выводимый с помощью этой формулы, ЕЯ-описание, точность, ограничения, при которых возможно использовать формулу. Соответствующий пример приведён в таблице 10.

Таблица 7 – Фрагмент таблицы атрибутов класса онтологии правового объекта

Имя атрибута класса	Определенный концепт	Тип значения	Мощность	Значение
вид контроля	Частное предприятие	[частное, акционерное]	(1,2)	частное
вид контроля	Акционерное предприятие	[частное, акционерное]	(1,2)	акционерное

Таблица 8 – Пример логической аксиомы онтологии судебного процесса

Имя аксиомы	Описание	Выражение	Ссылочные концепты	Ссылочные отношения	Переменные
Несовместимость истец-ответчик	Человек не может быть истцом и ответчиком в одном судебном процессе	не (существует (X,Y) (человек(X) и судебный процесс (Y) и [истец] (X,Y) и [ответчик] (X,Y)))	Человек Судебный процесс	истец ответчик	X Y

Таблица 9 – Пример кортежа таблицы констант онтологии правового объекта

Имя	Тип значения	Значение	Единица измерения
возраст совершеннолетия	количественное числительное	18	годы

Таблица 10 – Пример фрагмента таблицы формул

Имя формулы	Либеральное поведение
Выводимый атрибут	Варианты поведения
Формула	Варианты поведения = Разрешено $\cup$ Прямо не запрещено
Описание	«Разрешено всё то, что прямо не запрещено»: субъекту предоставляется возможность выбирать любой вариант поведения, кроме тех, что прямо и строго сформулированы в виде запретов
Основной атрибут экземпляра	Разрешено Прямо не запрещено
Основной атрибут класс	-
Константы	-
Точность	-
Ограничения	Варианты поведения, строго запрещенные законом

- 8) *деревья классификации атрибутов* графически показывают соответствующие атрибуты и константы, используемые для вывода значения корневого атрибута и формулы, применяемые для этого. Пример фрагмента представлен на рисунке 4. Фактически, эти деревья используются для проверки полноты: все атрибуты, представленные в формуле, должны иметь описания, и ни один из атрибутов не должен быть пропущен.

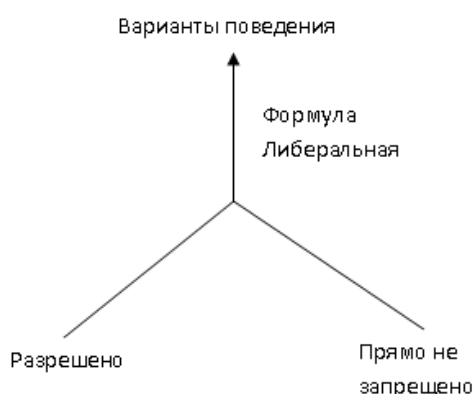


Рисунок 4 – Фрагмент дерева классификации атрибутов

9) в *таблице экземпляров* для каждого входа в словарь концептов специфицируются: имя экземпляра, имя концепта (соответствующего), его атрибуты и их значения. Фрагмент этого отношения представлен в таблице 11.

Таблица 11 – Фрагмент таблицы экземпляров онтологии правового объекта

Имя экземпляра	Имя концепта	Атрибут	Значение
Гражданский суд	суд	место	Киев
		территориальная юрисдикция	Украина
Верховный суд	суд	территориальная юрисдикция	Украина
Конституционный суд	суд	количество членов	12
		территориальная юрисдикция	Украина

## Заключение

Направление онтологического инжиниринга является актуальной задачей, в рамках которой большой интерес представляет полученный вариант базовой онтологии для системы права. С нашей точки зрения, представленный здесь результат адаптации является базовым уровнем разработки соответствующей компьютерной технологии в правовой области.

Традиционно для анализа текстов используются методологии обработки документов с использованием знаний ключевых фраз. Каждая фраза может представлять множество значений, и много различных фраз могут иметь одинаковые значения. Предлагаемый метод анализирует грамматику предложения и строит онтологию документов. Затем отношения между документами сравниваются с вычислением уровней их сходства и различия. Представленная методология, с нашей точки зрения, более эффективна по сравнению с часто используемым подходом К-средних ключевых фраз.

Описан способ введения нечёткой логики непосредственно в онтологию для заданной ПрО. Предлагаемые решения позволяют представлять неточную информацию в системах и реализовать нечёткие рассуждения. Представлен метод, основанный на использовании модификаторов понятий для автоматического их обновления. Актуальность решения этих проблем для правовой области знаний не вызывает сомнений.

В настоящее время ведутся работы по расширению полученной русскоязычной версии правовой онтологии и использованию её для решения прикладных задач правовых систем, основанных на знаниях.



## СПИСОК ИСТОЧНИКОВ

- [1] *Соловьев, В.Д.* Онтологии и тезаурусы / В.Д. Соловьев, Б.В. Добров, В.В. Иванов, Н.В. Лукашевич – Казань: Казанский государственный университет, 2006. – 157 с.
- [2] *Noy, N.* Ontology Development 101 / N. Noy, D. McGuinness. – Stanford Knowledge Systems Laboratory Technical Report KSL-01-05. – 2001. – P. 217-228
- [3] IDEF5 Method Report. – Knowledge Based Systems, Inc. for Information Integration for Concurrent Engineering. – 1994.
- [4] *Fernandez-Lopez, M.,* Overview and Analysis of methodologies for building ontologies / M Fernandez-Lopez, A. Gumez-Parez // Knowledge Engineering Review (KER). - Vol. 17[2]. – 2002.
- [5] DILIGENT Ontology Engineering. – (<http://www.aifb.uni-karlsruhe.de/WBS/cte/ontologyengineering/diligent.htm>).
- [6] NeOn Project. – (<http://www.neon-project.org>).
- [7] *Деринг, Ю.* Онтологии в правовой сфере. / Ю. Деринг, М. Хербергер // Проблемы правовой информатизации. – 2006. – № 2(12).
- [8] *Hou, J.L.* A document content extraction model using keyword correlation analysis / J.L. Hou, C.A. Chan // Electron. Bus. Manag. - 2003. - Vol. I. - No. I. - P. 54-62.
- [9] *Wu, Z.* Verb semantics and lexical selection / Z. Wu, M. Palmer // Proc. 12nd Anna. Meeting Assoc. Comput. Linguist (Las Cruces. NM. Jun. 27-30, 1994). - P. 133-138.
- [10] *Witten, I.H.* Adaptive text mining: Interring structure from sequences / I.H. Witten // Discret Algorithms. – 2004. - Vol. 2. - No. 2. Jun. - P. 137-159.
- [11] *Sanchez, S.N.* A feature mining based approach for the classification of text documents into disjoint classes. / S.N. Sanchez, E. Triantaphyllou, O. Kraft // Inf. Process. Manag. – 2002. - Vol. 38. - No. 4. Jul. - P. 283-604.
- [12] *Mamdani, E.H.* Application of fuzzy logic to approximate reasoning using linguistic synthesis / E.H. Mamdani // Proc. 6th Int. Symp. Multiple-Valued Logic. Logan. UT. 1976. - P. 196-202.
- [13] *Mamdani, E.H.* Application of fuzzy algorithm for control of simple dynamic plant. / E.H. Mamdani // Proc. Inst. Elect. Eng. – 1974. - Vol. 121. - No. 12. - P. 1585-1588.
- [14] *Асаки, К.* Прикладные нечёткие системы / К. Асаки, Д. Ватада, С. Иваи и др. // Под ред. Тэрано Т., Сугэно М. – М.: Мир, 1993. - 368 с.
- [15] *Заде, Л.А.* Основы нового подхода к анализу сложных систем и процессов принятия решений / Л.А. Заде // Математика сегодня: Сборник статей; пер. с англ. – М.: Знание, 1974. – С.5-49.
- [16] *Заде, Л.* Понятие лингвистической переменной и ее применение к принятию приближенных решений / Л. Заде. — М.: Мир, 1976. -165 с.
- [17] *Заде, Л.А.* Роль мягких вычислений и нечёткой логики в понимании, конструировании и развитии информационных/ интеллектуальных систем / Л.А. Заде // Новости Искусственного Интеллекта. 2001. - №2-3. - С. 7-11.
- [18] *Лихтарников, Л.М.* Математическая логика: Курс лекций. Задачник-практикум и решения / Л.М. Лихтарников, Т.Г. Сукачева.— СПб.: Лань, 1998.- 285 с.
- [19] *Новак, В.* Математические принципы нечёткой логики / В. Новак, И. Перфильева, И. Мочкорж – М.: ФИЗМАТЛИТ, 2006. – 352 с.
- [20] *Орловский, С.А.* Проблемы принятия решений при нечёткой исходной информации / С.А. Орловский. — М.: Радио и связь, 1981. — 286 с.
- [21] *Штовба, С.Д.* Введение в теорию нечётких множеств и нечёткую логику - <http://matlab.exponenta.ru/fuzzylogic/book1/index.php>
- [22] LKIF Ontology. A core ontology of basic legal concepts. – <http://www.estrellaproject.org/lkif-core/>.

## ONTOLOGY DESIGN BASED ON THE TEXT CONTENT WITH FUZZY LOGIC FOR LEGAL DOMAIN

I.R. Valkman<sup>1</sup>, C.A. Hala<sup>2</sup>

*International Research and Training Center of Information Technologies and Systems NAS and MES of Ukraine, Kiev, Ukraine*

<sup>1</sup>[yur@valkman.kiev.ua](mailto:yur@valkman.kiev.ua), <sup>2</sup>[cecerongreat@ukr.net](mailto:cecerongreat@ukr.net)

## Abstract

The paper is dedicated to the problems of ontological engineering in the legal domain. Features of creation of legal ontology and problem of realization of this ontology are investigated. Systems, which provide an exchange of knowledge between existing legal systems, have to support the translation of the legal knowledge bases presented in various formats and forms. The formalization of representation of this knowledge has to provide compatibility within larger architecture for development of the legal knowledge systems. The method of introduction of fuzzy logic in legal ontologies is offered. Two opportunities by means of which the expert can add knowledge in ontology are described. Proposed solutions allow to describe inexact information in systems and to realize fuzzy reasoning.

**Key words:** *indistinct conclusion, fuzzy logic, ontology structure, text analysis, text content, representation of knowledge, ontology, Protégé, basic ontology, law.*

## References

- [1] *Solov'ev, V.D.* Ontologii i tezaury [Ontologies and thesauri] / V.D. Solov'ev, B.V. Dobrov, V.V. Ivanov, N.V. Lukashovich – Kazan: Kazan State University, 2006. – 157 p. (In Russian)
- [2] *Noy, N.* Ontology Development 101 / N. Noy, D. McGuinness. – Stanford Knowledge Systems Laboratory Technical Report KSL-01-05. - 2001. - P. 217-228
- [3] IDEF5 Method Report. – Knowledge Based Systems, Inc. for Information Integration for Concurrent Engineering. – 1994.
- [4] *Fernandez-Lopez, M.*, Overview and Analysis of methodologies for building ontologies / M Fernandez-Lopez, A. Gymeze-Parez // Knowledge Engineering Review (KER). - Vol. 17[2]. – 2002.
- [5] DILIGENT Ontology Engineering. – <http://www.aifbuni-karlsruhe.de/WBS/cte/ontologyengineering/diligent.htm>.
- [6] NeOn Project. – <http://www.neon-project.org>.
- [7] *Dering, Yu.* Ontologii v pravovoj sfere [Ontologies in the legal domain]/ Yu. Dering, M. Herberber // Problemy pravovoj informatizatsii [Problems of legal automation]. – 2006. – No. 2(12). (In Russian)
- [8] *Hou, J.L.* A document content extraction model using keyword correlation analysis / J.L. Hou, C.A. Chan // Electron. Bus. Manag. - 2003. - Vol. I. - No. I. - P. 54-62.
- [9] *Wu, Z.* Verb semantics and lexical selection / Z. Wu, M. Palmer // Proc. 12nd Anna. Meeting Assoc. Comput. Linguist (Las Cruces. NM. Jun. 27-30, 1994). - P. 133-138.
- [10] *Witten, I.H.* Adaptive text mining: Interring structure from sequences / I.H. Witten // Discret Algorithms. – 2004. - Vol. 2. - No. 2. Jun. - P. 137-159.
- [11] *Sanchez, S.N.* A feature mining based approach for the classification of text documents into disjoint classes. / S.N. Sanchez, E. Triantaphyllou, O. Kraft // Inf. Process. Manag. – 2002. - Vol. 38. - No. 4. Jul. - P. 283-604.
- [12] *Mamdani, E.H.* Application of fuzzy logic to approximate reasoning using linguistic synthesis / E.H. Mamdani // Proc. 6th Int. Symp. Multiple-Valued Logic. Logan. UT. 1976. - P. 196-202.
- [13] *Mamdani, E.H.* Application of fuzzy algorithm for control of simple dynamic plant. / E.H. Mamdani // Proc. Inst. Elect. Eng. – 1974. - Vol. 121. - No. 12. - P. 1585-1588.
- [14] *Asaki, K.* Prikladnye nechyotkie sistemy [Applied fuzzy systems]/ K. Asaki, D. Vatada, S. Ivai and others // under ed. of Tehrano T., Sugehno M. – Moscow: Mir, 1993. - 368 p. (in Russian)
- [15] *Zade, L.A.* Osnovy novogo podkhoda k analizu slozhnykh sistem i protsessov prinyatiya reshenij [Foundations for a new approach to the analysis of complex systems and decision-making processes] / L.A. Zade // Matematika segodnya: Sbornik statej; per. s angl. [Mathematics today, collection of articles, translated from English] – Moscow: Znanie, 1974. – pp.5-49. (In Russian)
- [16] *Zade, L.A.* Ponyatie lingvisticheskoy peremennoj i ee primenenie k prinyatiyu priblizhennykh reshenij [The concept of linguistic variable and its application to the adoption of approximate solutions] / L. Zade. — Moscow: Mir, 1976. -165 p. (In Russian)
- [17] *Zade, L.A.* Rol' myagkikh vychislenij i nechyotkoj logiki v ponimani, konstruirovani i razvitii informatsionnykh/intellektual'nykh sistem [The role of soft computing and fuzzy logic in understanding the design and development of information / intelligent systems]/ L.A. Zade // Novosti Iskusstvennogo Intellekta. 2001. – No. 2-3. - pp. 7-11. (In Russian)
- [18] *Likhtarnikov, L.M.* Matematicheskaya logika: Kurs lektzij. Zadachnik-praktikum i resheniya [Mathematical Logic: Lectures. Problem book and solutions]/ L.M. Likhtarnikov, T.G. Sukacheva. — Saint-Petersburg: Lan', 1998.- 285 p. (In Russian)
- [19] *Novak, V.* Matematicheskie printsipy nechyotkoj logiki [Mathematical principles of fuzzy logic] / V. Novak, I. Perfil'eva, I.Mochkorzh – Moscow: FIZMATLIT,2006. – 352 p. (In Russian)
- [20] *Orlovskij, S.A.* Problemy prinyatiya reshenij pri nechyotkoj iskhodnoj informatsii [Decision making with fuzzy initial information]/ S.A. Orlovskij. — Moscow: Radio I svyaz, 1981. — 286 p. (In Russian)

- [21] *Shtovba, S.D.* Vvedenie v teoriyu nechyotkikh mnozhestv i nechyotkuyu logiku [Introduction to the theory of fuzzy sets and fuzzy logic] - <http://matlab.exponenta.ru/fuzzylogic/book1/index.php> (In Russian)
- [22] LKIF Ontology. A core ontology of basic legal concepts. – <http://www.estrellaproject.org/lkif-core/>.

### Сведения об авторах



**Валькман Юрий Роландович**, 1948 г. рождения. Окончил Киевский национальный университет им. Т.Г. Шевченко в 1971 г., д.т.н. (1996). Заведующий отделом распределенных интеллектуальных систем Международного научно-учебного центра информационных технологий и систем НАН и МОН Украины, профессор кафедры математических методов системного анализа Института прикладного системного анализа Киевского политехнического института и кафедры математической информатики факультета кибернетики Киевского национального университета им. Т.Г. Шевченко, вице-председатель Совета Международной ассоциации

создателей и пользователей интеллектуальных систем, член Российской ассоциации искусственного интеллекта, член редколлегии журналов «Программные продукты и системы», «Онтология проектирования», «Информатика, вычислительная техника и инженерное образование», «Радиоэлектроника, информатика, управление». В списке научных трудов более 300 работ в области разработки компьютерных и интеллектуальных технологий и систем.

**Valkman Iuri Rolandovich** (b.1948) graduated from Kiev National University named after Taras Shevchenko in 1971, Dr. of science (1996). He is head of the department of distributed intelligent systems. at International Research and Training Center of Information Technologies and Systems NAS and MES of Ukraine. He is professor of the Department of Mathematical Methods of System Analysis at the Institute for Applied Systems Analysis Kiev Polytechnic Institute and professor of the Department of Mathematical Informatics Faculty of Cybernetics Taras Shevchenko National University of Kiev. He is vice-chairman of the board of Association of developers and users of intelligent systems, member of Russian association of artificial intelligence, member of the editorial board of such journals as «Program products and systems», «Ontology of designing», «Informatics, computing and engineering education», «Electronics, computer science, management». He is co-author of more then 300 scientific articles and abstracts in the field of computer and intelligent technologies and systems.



**Хала Екатерина Александровна**, 1988 г. рождения. Окончила Национальный технический университет Украины «Киевский политехнический институт» в 2011 г. Аспирантка Международного научно-учебного центра информационных технологий и систем НАН и МОН Украины. В списке научных трудов около 10 работ в области компьютерного моделирования сложных систем; систем, основанных на знаниях.

**Catherine Aleksandrovna Hala** (b.1988) graduated from the National Technical University of Ukraine "Kiev Polytechnic institute" in 2011. She is Postgraduate Student at International Research and Training Center of Information Technologies and Systems NAS and MES of Ukraine. She is co-author of about 10 publications in the field of computer simulation of complex systems based on knowledge.