Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шебзухова Татьяна Александровна
Должность: Директор Пятигорского института (филиал) Северо-Кавказского федерального университета
Дата подписания: 05.10.2023 16:50:12
Уникальный программный ключ:
d74ce93cd40e39275c3ba2f58486412a1c8ef96f

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Пятигорский институт (филиал) СКФУ

Методические указания

по выполнению лабораторных работ
по дисциплине
«Искусственный интеллект в профессиональной сфере»
для студентов специальности **38.05.02 Таможенное дело**
Направленность (профиль) **Таможенный контроль**

Пятигорск
2023

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3
СТРУКТУРА И СОДЕРЖАНИЕ ЛАБОРАТОРНЫХ ЗАНЯТИЙ 6
Лабораторная работа 1. Анализ современных программных средств с применением ИИ. ...6
Лабораторная работа 2 Формализация знаний Использование семантических сетей для представления знаний 7
Лабораторные работы 3-5 Создание онтологии в системе PROTÉGÉ. Основы RDF и OWL 12
Лабораторные работы 6 Разработка учебной экспертной системы. 57
Лабораторная работа 7 Анализ существующих образовательных платформенных решений. 67
Список литературы 73

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

ВВЕДЕНИЕ

Цели и задачи освоения дисциплины

Целью освоения дисциплины является ознакомление студентов с кругом задач в области искусственного интеллекта, а также знакомство с основными понятиями и методами машинного обучения и их применением к задачам, относящимся к профессиональной области.

Задачи освоения дисциплины – ознакомление студентов с возможностями использования технологий искусственного интеллекта в профессиональной сфере.

В результате освоения дисциплины студенты должны:

– использовать принципы работы современных информационных технологий для решения задач профессиональной деятельности;

– выбирать цифровые технологии, программные средства для решения профессиональных задач;

– применять при решении задач профессиональной деятельности специализированное программное обеспечение и методы искусственного интеллекта;

– разрабатывать основные модули интеллектуальных систем, владеть приемами решения лабораторных задач в предметной области.

Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	ИД-1.УК-1. выделяет проблемную ситуацию, осуществляет ее анализ и диагностику на основе системного подхода	Применяет основы поиска и критического анализа информации; использует методы системного подхода для решения поставленных задач с помощью цифровых и информационных технологий
	ИД-2. УК-1. осуществляет поиск, отбор и систематизацию информации для определения альтернативных вариантов стратегических решений в проблемной ситуации	Использует ресурсы Интернет и его сервисы, включая облачные хранилища и другие инструменты организации проектной, в том числе совместной, работы
	ИД-3. УК-1. определяет и оценивает риски возможных вариантов решений проблемной ситуации, выбирает оптимальный вариант её решения	Применяет цифровые технологии для решения поставленных задач, выбирая оптимальный способ их решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений
ОПК-6 Способен понимать принципы работы современных	ИД-1. ОПК-6. Знает принципы работы современных информационных технологий	Понимает принципы работы различных современных информационных технологий, программного

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: [illegible] Александровна
Действителен: с 19.08.2022 по 19.08.2023

информационных технологий и использовать их для решения задач профессиональной деятельности		обеспечения ПК и знает его возможности, а также применяет основные методы, способы и средства получения, хранения, поиска, систематизации, обработки и передачи информации при решении профессиональных задач
	ИД-2. ОПК-6. Применяет информационные технологии и средства обеспечения их функционирования для решения задач профессиональной деятельности	Применяет информационные технологии для решения задач профессиональной деятельности; работает с текстовыми и числовыми данными, проводить простейшую аналитику текстовых и числовых данных с помощью специального программного обеспечения; обрабатывает графические изображения
ПК-14 Способен адаптировать и модифицировать специализированное программное обеспечение, методы и алгоритмы систем искусственного интеллекта и машинного обучения в профессиональной деятельности	ИД-1ПК-14 Ориентируется в современных тенденциях развития цифровых технологий, выбирает технологии или программные средства для решения поставленных задач.	Владеет навыками применения информационных технологий для решения профессиональных задач, может выбрать технологии или программные средства для решения поставленных задач
	ИД-2ПК-14 Применяет при решении задач профессиональной деятельности специализированное программное обеспечение, методы искусственного интеллекта и машинного обучения	Готовность применять при решении задач профессиональной деятельности специализированное программное обеспечение и методы искусственного интеллекта
	ИД-3ПК-14 Способен обосновать применение того или иного алгоритма машинного обучения для решения конкретной задачи	Применяет тот или иной алгоритм машинного обучения для решения поставленных задач, выбирая оптимальный способ их решения, исходя из конкретной задачи
	ИД-4ПК-14 Разрабатывает оригинальные модели и алгоритмы искусственного интеллекта и машинного обучения в профессиональной деятельности	Готовность разрабатывать основные модули интеллектуальных систем, владеть приемами решения практических задач в предметной области

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

1. **Наименование лабораторных занятий**

№	Наименование тем дисциплины, их краткое содержание	Объем часов	Из них практическая подготовка, часов
6 семестр			
1.	Анализ современных программных средств с применением ИИ	3	
2.	Формализация знаний Использование семантических сетей для представления знаний	3	
3.	Создание онтологии в системе Protégé	4,5	
4.	Разработка учебной экспертной системы.	0,75	
4	Анализ существующих образовательных платформенных решений.	0,75	
	Итого за 6 семестр	12	
	Итого	12	

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

СТРУКТУРА И СОДЕРЖАНИЕ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Лабораторная работа 1.

Анализ современных программных средств с применением ИИ.

Вопросы (компетенции, навыки) для освоения:

1. Изучить современные программные продукты, основанные на ИИ.
2. Научиться анализировать возможности программных средств с применением ИИ.

Задания для выполнения и методические рекомендации:

Задание 1

Используя приведенный ниже список интернет-ссылок, а также другие интернет-источники, изучите интеллектуальные системы, предлагаемые на рынке.

1. https://www.zeluslugi.ru/info-czentr/stati/primery-iskusstvennogo-intellekta-programmy
2. https://3dnews.ru/1058498/intellektualnoe-prevoshodstvo-10-zaslugivayushchih-vnimaniya-aiservisov-i-prilogeniy
3. https://soware.ru/categories/artificial-intelligence-platforms

Определите их направление:
- экспертные системы (ЭС);
- системы с базами знаний;
- интеллектуальное обучение;
- нейронные сети.

Задание 2

Проанализируйте возможности, области применения, достоинства и недостатки изученных интеллектуальных систем. На основе анализа составьте отчет в табличной форме:

Название ПО или сервиса, версия	Возможности программы	Область применения	Стоимость	Источник информации (www.)

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Сделайте вывод о рынке ПО. Рассчитайте удельный вес по области применения ПО. Изобразите результат графически (постройте диаграмму).

Лабораторная работа 2
Формализация знаний
Использование семантических сетей для представления знаний

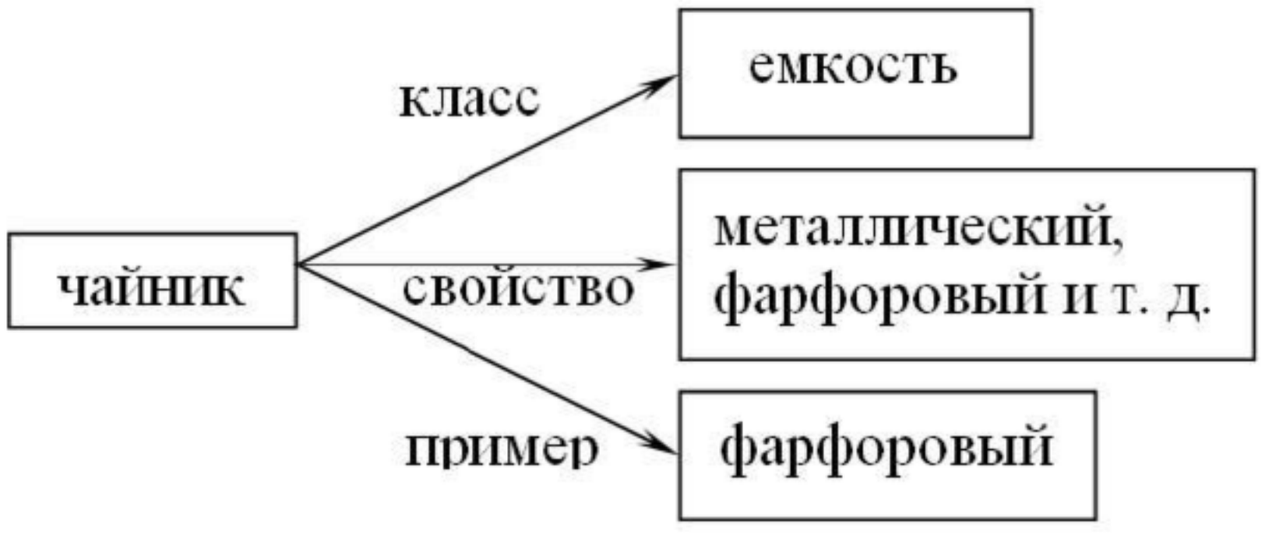
Вопросы (компетенции, навыки) для освоения:

1. Изучить общее представление о семантической сети и фреймах для представления знаний.
2. Научиться строить семантическую сеть и используя фреймы реализовать структуру отношений.

Задания для выполнения и методические рекомендации:

Семантическая сеть – это один из способов представления знаний. Изначально семантическая сеть была задумана как модель представления долговременной памяти в психологии, но впоследствии стала одним из способов представления знаний в ЭС.

Семантика – означает общие отношения между символами и объектами из этих символов.



Простейший образец семантической сети

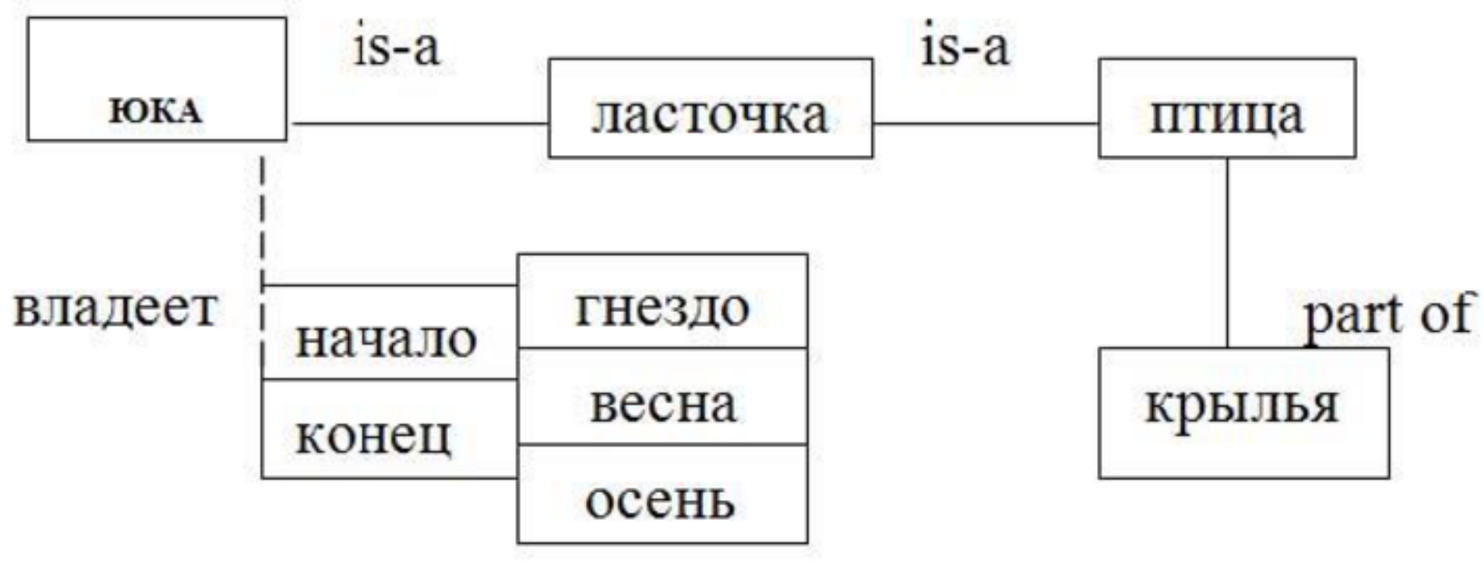
Вершины – это объекты, дуги – это отношения. Семантическая модель не раскрывает сама по себе каким образом осуществляется представление знаний. Поэтому семантическая сеть рассматривается как метод представления знаний и структурирования знаний. При расширении семантической сети в ней возникают другие отношения:

IS –A (принадлежит) и PART OF (является частью) отношение:

целое → часть.

Ласточка IS – A птица, «нос» PART OF «тело». Например:

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023



Расширение семантической сети

Могут быть и другие отношения: владеет. Тогда семантическая сеть расширяется иерархически (вершина имеет две ветви). Кроме того, можно расширить сеть и другим отношением:

период → «весна – лето».

Получается иерархическая структура понятия ЮКО. Можно разбить на подсхемы. Большой проблемой для семантических сетей является то, что результат вывода не гарантирует достоверности, так как вывод есть просто наследование свойств ветви is-a.

Для отображения иерархических отношений между объектами и введения единой семантики в семантические сети было предложено использовать процедурные сети. Сеть строится на основе класса (понятия); вершины, дуги и процедуры представлены как объекты.

Задание 1

Используя соответствующие дуги построить семантическую сеть, касающуюся:

1. географии какого-либо региона. Дуги: государство, страна, континент, широта.

2. диагностики глазных заболеваний. Дуги: категории болезней, патофизиологическое состояние, наблюдения, симптомы.

3. распознавания химических структур. Дуги: формула вещества, свойства вещества, область применения, меры предосторожности.

4. процедуры поиска полезных ископаемых. Дуги: наименование ископаемого, расположение месторождения, глубина залегания, методы добычи.

5. судебной процедуры. Дуги: юридическое лицо, событие, меры воздействия, способы расследования.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

6. распределения продуктов по магазинам. Дуги: источник снабжения, наименование продукта, способ транспортировки, конечный пункт транспортировки.

7. определения принадлежности животного к определенному виду, типу, семейству. Дуги: место обитания, строение, особенности поведения, вид питания.

8. классификации пищевых продуктов. Дуги: наименование продукта, составляющие части, способ приготовления, срок хранения.

9. распознавания типа компьютера. Дуги: страна изготовитель, стандартная конфигурация, область применения, используемое программное обеспечение.

10. иерархической структуры БД. Дуги: система, состояние, назначение, взаимодействие составляющих.

Использование фреймов для представления знаний

Фреймы - один из распространенных формализмов представления знаний в ЭС. Фрейм можно представить себе как структуру, состоящую из набора ячеек - слотов. Каждый слот состоит из имени и ассоциируемых с ним значений. Значения могут представлять собой данные, процедуры, ссылки на другие фреймы или быть пустыми. Такое построение оказывается очень удобным для моделирования аналогий, описания областей с родовидовыми связями понятий и т.п.

Любой фрейм состоит из некоторых составляющих, имена и содержание которых описано ниже:

1. Имя фрейма. Это идентификатор, присваиваемый фрейму, фрейм должен иметь имя уникальное в данной фреймовой системе.

2. Имя слота. Это идентификатор, присваиваемый слоту; слот должен иметь уникальное имя во фрейме, к которому он принадлежит. Обычно имя слота не несет никакой смысловой нагрузки и является лишь идентификатором данного слота.

3. Указатели наследования. Эти указатели касаются только фреймовых систем иерархического типа, основанные на отношениях “абстрактное-конкретное”, они показывают, какую информацию об атрибутах слотов во фрейме верхнего уровня наследуют слоты с такими же именами во фрейме нижнего уровня. Типичные указатели наследования Unique (U: - уникальный), Same (S: такой же), Range (R: установление границ), Override (O: игнорировать) и т.п. U показывает, что фрейм может иметь слоты с разными значениями: S - все слоты должны иметь одинаковые значения, R - значение слотов фрейма нижнего уровня должны находиться в пределах, указанных значениями слотов фрейма верхнего уровня, O - при отсутствии указания значение слота фрейма верхнего уровня становится значением слота фрейма нижнего уровня, но в случае определения нового значения слотов фреймов нижних уровней указываются в качестве значений слотов.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB6B952205E7BA900080000042E
Владелец: Щебзукова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

4. Указание типа данных. указывается, что слот имеет численное значение, либо служит указателем другого фрейма. К типам данных относятся:

FRAME (указатель), INTEGER (целый), REAL (действительный), BOOL (булев), LISP (присоединенная процедура), TEXT (текст), LIST (список), TABLE (таблица), EXPRESSION (выражение) и др.

5. Значение слота. Пункт ввода значения слота. Значение слота должно совпадать с указанным типом данных этого слота, кроме того должно выполняться условие наследования.

6. Демон. Здесь дается определение демонов типа IF-NEEDED, IF-ADDED, IF-REMOVED и т.д. Демоном называется процедура, автоматически запускаемая при выполнении некоторого условия. демоны запускаются при

обращении к соответствующему слоту. Кроме того, демон является разновидностью присоединенной процедуры.

7. Присоединенная процедура. В качестве значения слота можно использовать программу процедурного типа. Когда мы говорим, что в моделях представления знаний фреймами объединяются процедурные и декларативные знания, то считаем демоны и присоединенные процедуры процедурными знаниями.

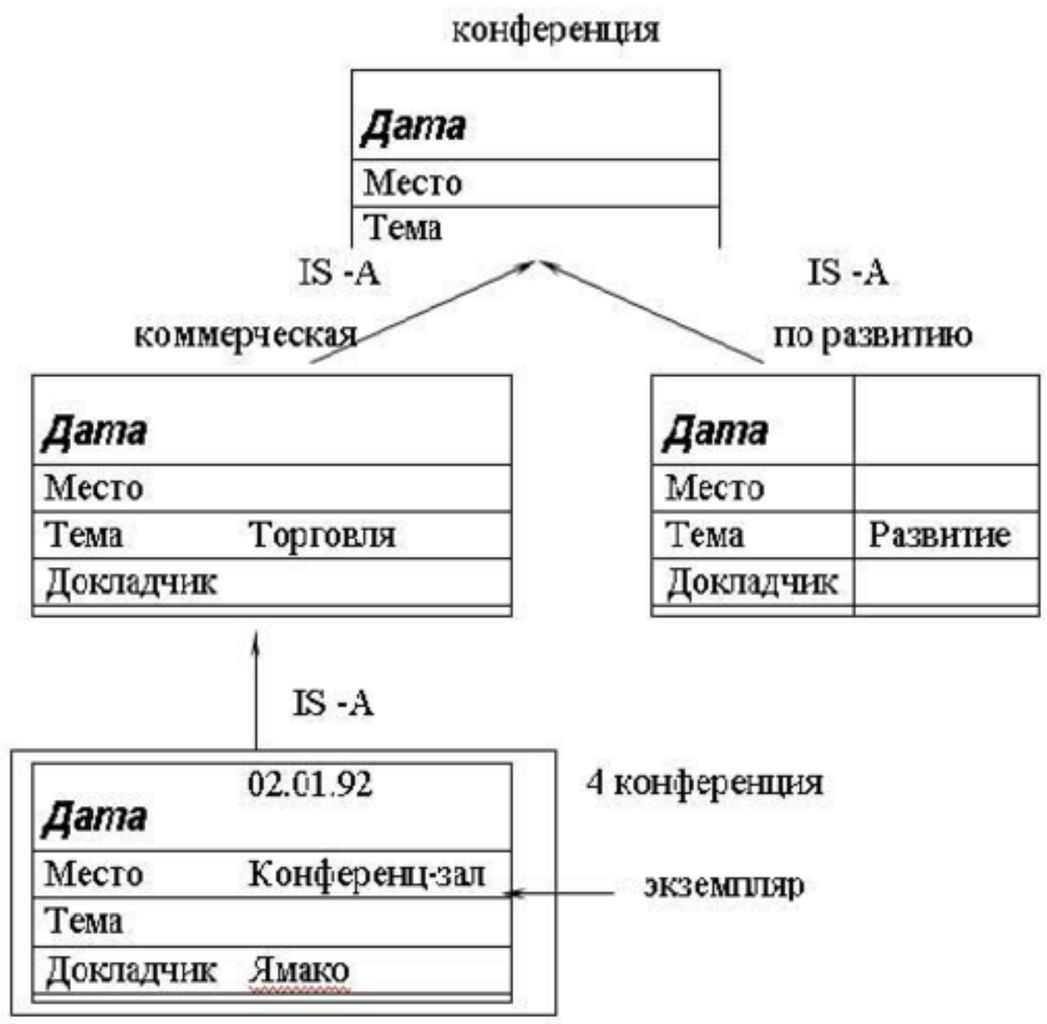
Особенностью иерархической структуры является то, что информация об атрибутах фрейма на верхнем уровне совместно используется всеми фреймами нижних уровней, связанных с ним.

Например: Фреймовое представление конференции.

Иерархические фреймовые структуры базируются на отношениях IS – A между фреймами, описывающими некоторую конференцию. Все фреймы должны содержать информацию о ДАТЕ, МЕСТЕ, НАЗВАНИИ ТЕМЫ, ДОКЛАДЧИКЕ. Таким образом, на самом верхнем уровне определен фрейм КОНФЕРЕНЦИЯ.

Конференции разделяются на коммерческие и по развитию. Они составляют дочерние фреймы. В них могут быть добавлены слоты: объем торговли и бюджет.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023



Пример фреймовой модели

Задание 2.

Используя фреймовую модель представления знаний реализовать структуру отношений, описывающие следующие ситуации:

1. экзамен по дисциплине за семестр у преподавателя при составляющих: семестр, экзамен, преподаватель, оценка, студент, получать.

2. ведомость при составляющих: дисциплина, студент, экзамен, семестр, преподаватель, оценка.

3. конференция по коммерческим вопросам при составляющих: дата, место проведения, тема, цель выступающие.

4. получение оценки при составляющих: преподаватель, студент, оценка, получать.

5. использования изделия при составляющих: организация, разработка технологического решения, исследование «физического эффекта», методы создания изделия.

6. информационная структура БД в машиностроении при составляющих: физические эффекты, технические решения, изделия, объект поставки изделия, приборы и стенды, нормативы.

7. классификация продукта при составляющих: название, область применения, способ хранения, способ транспортировки.

8. аудитория (описание) при составляющих: вместимость, назначение, составляющие, местонахождение.

9. животный мир при составляющих: вид, тип, среда обитания, особенности поведения.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Лабораторные работы 3-5

Создание онтологии в системе PROTÉGÉ. Основы RDF и OWL

Вопросы (компетенции, навыки) для освоения:

3. Определить предметную область, создать классы, отношения с использованием системы Protégé, а также на примере созданной БЗ рассмотреть модель RDF, язык OWL и синтаксис Turtle.
4. Научиться реализовывать понятия Defined Classes, DL Query в системе Protégé
5. Изучить возможности языка запросов SPARQL
6. Научиться анализировать создавать правила для проверки информации о товарах с помощью языков SWRL и SQWRL .

Задания для выполнения и методические рекомендации:

В качестве примера будет рассмотрена информация о конкретном товаре (конкретном смартфоне[1]) и на ее основании составлена первая версия БЗ.

Перед началом работы с системой Protégé [6] необходимо составить хотя бы некоторую упрощенную концептуальную модель предметной области. И, для того чтобы потом не было сложностей с ее формализацией, необходимо определить какими сущностями можно оперировать. В данном случае это сущности модели RDF [1, 2] и языка OWL [4], т.к. многие системы и библиотеки, в том числе Protégé поддерживает эти стандарты, и они являются наиболее распространенными.

Стоит сразу отметить, что модель RDF является абстрактной и существует множество языков с разными синтаксисами, реализующих ее [1]. OWL также может быть преобразован в несколько нотаций и в любой из языков RDF [4]. Конкретные синтаксисы будут обсуждаться позднее и все

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен с 19.08.2022 по 19.08.2023

[1] https://www.dns-shop.ru/product/052629b3621f3330/62-smartfon-samsung-galaxy-s9-256-gb-fioletovyj/characteristics/

примеры, описанные ниже могут их не придерживаться.

Онтология в рамках модели RDF предполагает определение любых понятий предметной области в виде троек: субъект – предикат – объект. Множество таких троек образует граф понятий. Субъекты и объекты интерпретируются как вершины ориентированного графа, а предикаты как его дуги (и определяют отношения между сущностями). Вершины могут быть следующих типов: ресурс, пустая вершина.

Ресурс описывает некоторое понятие реального мира и разделяется на ссылочные ресурсы и скалярные. Ссылочные ресурсы – это некоторые понятия, которые могут иметь характеристики (например, смартфон, смартфон SMG965FZPHZER, экран). Таким понятиям присваивается глобальный уникальный идентификатор IRI (Internationalized Resource Identifier) (например, http://myontololgy.org/goods#phone1). Соответственно, такие ресурсы могут находиться как на месте субъекта, так и на месте объекта в зависимости от предиката.

Скалярные ресурсы указывают некоторые конкретные характеристики и являются просто значением определенного типа. Соответственно, такие ресурсы могут находится только на месте объекта. Но, при этом для них может определяться тип и для строковых значений может определяться язык. Например, для нашей предметной области такими сущностями могут являться: 6.2 (конкретная диагональ экрана; float), «Samsung Galaxy S9+¬ (оригинальное название; string); «черный¬@ru (цвет; string; русский).

Также необходимо отметить, что «предикат¬ также может быть «субъектом¬ или «объектом¬ в рамках других троек. Это необходимо чтобы, например, описать ограничения данного отношения.

Пустая вершина в основном необходима для «имитация¬ n-арных связей. Как видно из концепции модели, предикат может связывать только 2 ресурса. Но, это решается с помощью введения промежуточной вершины и разделением связи на несколько бинарных. При этом вершина не будет представлять понятие реального мира и ей нет смысла давать уникальный

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: [illegible]
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

глобальный идентификатор, достаточно локального для составления троек. Именно для таких вершин и введен обсуждаемый тип. При этом не всегда очевидно стоит ли вводить для той или иной сущности глобальной идентификатор. Например, конкретный смартфон обладает рядом характеристик дисплея и было решено выделить дисплей как отдельную сущность. При этом для нас он является неотъемлемой частью смартфона, а производитель смартфона может ставить разные модели дисплея от разных производителей при условии сохранения характеристик. Тогда может быть целесообразно использовать пустую вершину. С другой стороны, если производитель присвоил данному дисплею определенный код, то может быть целесообразно выделить его как ресурс. Т.к. он может быть установлен и на другой смартфон, а также будет проще реализовывать, например, запросы для определения сервисов, где есть в наличии данный дисплей. Для примера пустой вершины рассмотрим характеристику «разрешение¬ у конкретного дисплея:

SM-D-G965FZPHSER разрешение _x

_x горизонталь 2960

_x вертикаль 1440

Данные тройки описывают граф, изображенный на рис. 1.

Рис. 1 Граф, описывающий разрешение дисплея

Стоит отметить, что данные понятия также можно было описать с использование 2 связей «разрешение_гор¬, «разрешение_вер¬ и без промежуточной вершины. Что показывает, что создание онтологии остается вариативным процессом, зависящим от конкретных целей и предпочтений разработчика.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Дасбухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Как правило IRI ресурса состоит из идентификатора БЗ или

пространства имен и идентификатора понятия в нем. Например, «http://myontololgy.org/goods#phone1¬, может означать, что описывается понятие «phone1¬ в БЗ товаров с идентификатором «http://myontololgy.org/goods#¬. Чтобы не писать каждый раз полный идентификатор БЗ или пространства имен, его заменяют на префикс. Например, можно записывать «mo:phone1¬, где под «mo:¬ будет подразумеваться полный IRI БЗ.

Стандарт RDF вводит не только концепцию, но и 3 пространства имен с общими понятиями: rdf, rdfs, xsd [3]. Где xsd описывает типы, которые можно использовать в онтологиях на базе RDF (например, xsd:string, xsd:integer, xsd:double, xsd:boolean). Например, разрешение дисплея по горизонтали могло быть описано следующим образом: _x горизонталь “2960”^^xsd:integer. Основные понятия из rdf, rdfs, которые будут использоваться в рамках данной работы: rdfs:Resource (понятие ресурса, описанного ранее), rdfs:Class (понятие класса), rdf:Property (понятие отношения), rdfs:subClassOf (отношение, которое определяет «субъект¬ как подкласс «объекта¬), rdfs:subPropertyOf (отношение, которое определяет «субъект¬ как уточнение «объекта¬, являющегося отношением), rdfs:domain (отношение, определяющее область определения другого отношения), rdfs:range (отношение, определяющее область значения другого отношения), rdf:type (отношение, которое определяет «субъект¬ как экземпляр «объекта¬).

OWL является стандартизированным языком создания онтологий. Он имеет несколько вариантов синтаксиса (в том числе функциональный), а также может быть преобразован в любой из синтаксисов RDF. Т.к. OWL совместим с RDF, то в нашем случае он будет интерпретироваться как надстройка над RDF, которая уточняет подход к созданию онтологий, расширяет возможности модели и вводит новые сущности (из пространства имен owl).

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

БЗ, созданные с использованием OWL предполагают, что на них

производится вывод, который позволяет проверить непротиворечивость БЗ, вывести новые отношения.

В связи с этим описание онтологии можно разделить на:

1 Сущности. Все понятия, описанные ранее в рамках RDF, а также некоторые новые из OWL;

2 Выражения. Описание понятия через комбинацию множества других сущностей. Одним из основных примеров применения является определение класса через выражение. Например, «Камерофон¬ – это телефон, который имеет не менее 2 основных камер с разрешением не менее 20мегапикселей; новый тип «число¬ – это объединение множеств целых чисел и чисел с дробной частью.

3 Аксиомы. Точно известные положения и положения, которые всегда должны оставаться истинными. Например, на отношение «плотностьПикселей¬ могут быть наложены ограничения на область определения – экземпляр класс «Дисплей¬ и область значения – «целое число¬. При этом если в базе знаний будет тройка «G плотностьПикселей 529¬, то это может привести как к ошибке (если G точно не может являться экземпляром класса «Дисплей¬), так и к выводу нового отношения «G rdf:type Дисплей¬.

Как уже было сказано выше, OWL добавляет новые понятия и аксиомы. Для данной работы необходимы следующие понятия:

1 owl:Class. Аналогично rdfs:Class;

2 owl:Thing. Класс, являющийся предком всех классов. Аналогичен Object в некоторых языках программирования;

3 owl:ObjectProperty. Подкласс отношений (rdf:Property), связывающих два ресурса с IRI;

4 owl:DataProperty. Подкласс отношений (rdf:Property), связывающих ресурс с IRI и литерал;

5 owl:topObjectProperty/owl:topDataProperty. Верхнее в иерархии object/data property, являющееся предком всех object/data property;

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C00000435A9A8B9[illegible]
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

6 owl:FunctionalProperty. Класс отношений, применяемых к одному «субъекту¬ только один раз. Т.е. в онтологии не может существовать два триплета «x c y1¬, «x c y2¬, где y1 и y2 разные сущности и «c rdf:type owl:FunctionalProperty¬;

7 owl:disjointWith. Отношение, декларирующее, что экземпляр субъекта-класса не может быть одновременно экземпляром объекта-класса. Т.е. в онтологии не может существовать два триплета «x rdf:type y1¬, «x rdf:type y2¬, где y1 и y2 разные классы и «y1 owl:disjointWith y2¬.

8 Сущности, необходимые для составления выражений. Будут описывать при появлении (owl:members,, owl:unionOf и т.д.)

Теперь, когда сущности, которые могут быть использованы, обозначены, можно спланировать общее описание предметной области. Оно разделено на описание классов, экземпляров и отношений между ними (рис. 2) и описание самих отношений (рис. 3).

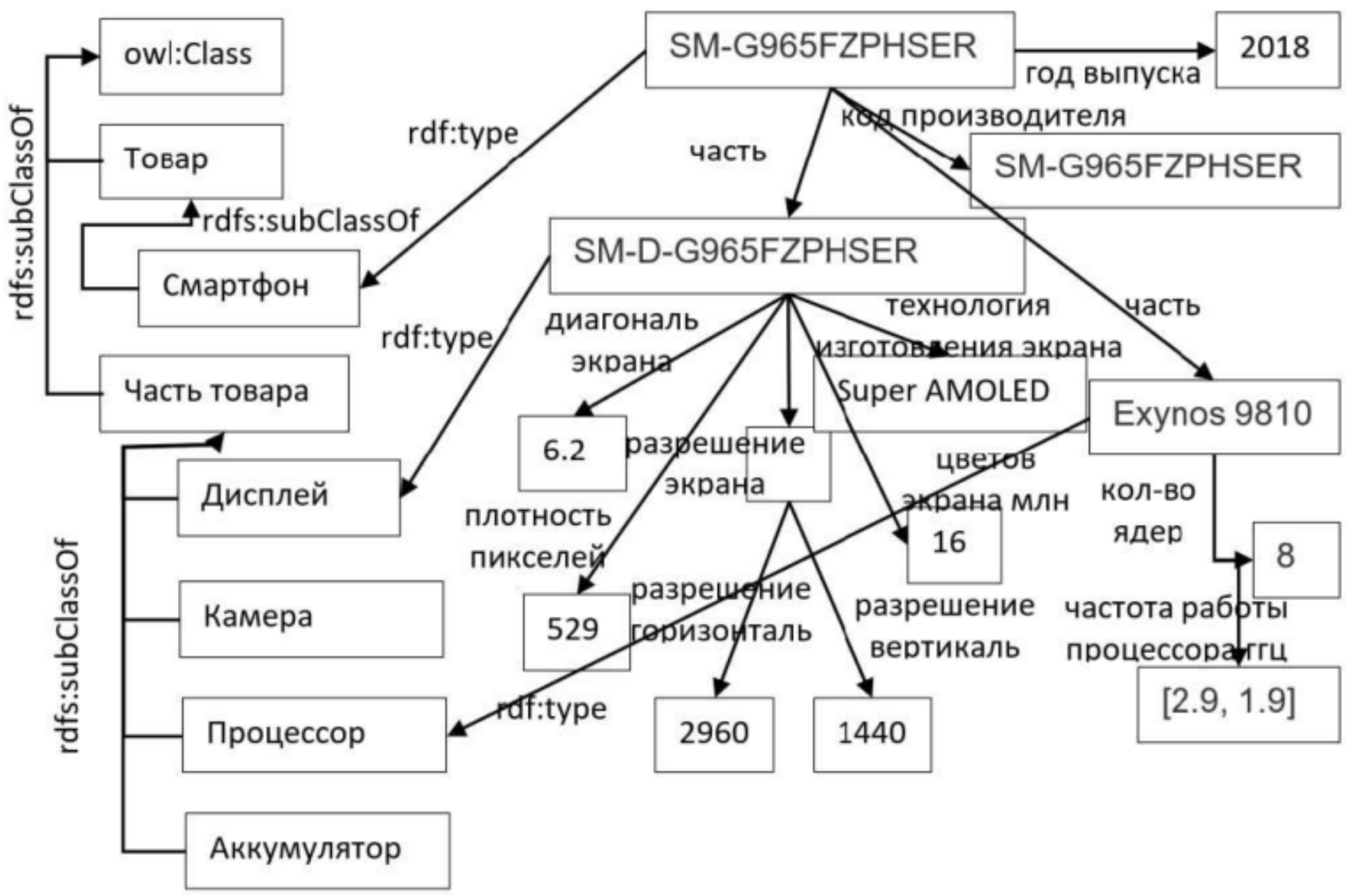


Рис. 2 Фрагмент графа, описывающего классы, экземпляры и их отношения в рамках описания смартфонов

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Рис. 3 Фрагмент графа, описывающего свойства в рамках описания смартфонов

В рамках приведенного фрагмента классов выделены очевидные классы:

«Товар», «Смартфон». Также выделены некоторые отдельные составляющие смартфона как подклассы класса «Часть товара»: «Дисплей», «Камера», «Процессор», «Аккумулятор». Сюда выделены только самые важные части, которые больше всего интересуют покупателей и наиболее ценны при обработке информации о товаре. Остальные характеристики будут задаваться отношениями между конкретным смартфоном и конкретными значениями (на изображении в качестве примера таких отношений можно видеть: «год выпуска», «код производителя»). Стоит отметить, что некоторые узлы (например, «[2.9, 1.9]») заданы в абстрактном виде и их конкретное задание будет зависеть от используемого синтаксиса и/или возможностей инструментальной системы.

В рамках приведенного фрагмента свойств не сделано разделение на ObjectProperty и DataProperty во избежание излишней конкретизации (при реализации разделение делается очевидным образом). В рамках прямых

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: [illegible]
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

наследников «owl:Property» выделены типы возможных отношений: «часть» (для связи товара с его частью), «числовое/текстовое/свободное/разрешение» (данные свойства определяют группу свойств, объединенных общей областью определения. Разделение на текстовое и свободное сделано для упрощения дальнейшей обработки. Под текстовыми подразумеваются конкретные значения, которые потом могут быть преобразованы в перечисляемый тип. В рамках свободных же описания могут содержать несколько характеристик, одну характеристику, записанную разным образом и т.д.), «unknown» (группа характеристик, добавляемых автоматически в процессе сбора информации, для которых невозможно определить принадлежность к ранее описанным группам).

Необходимо понимать, что в рамках рис. 2, рис. 3 приведены только фрагменты графа для упрощения. Оставшиеся фрагменты описываются аналогично и не обладают дополнительными особенностями.

Т.к. описание должно подразумевать возможность сбора информации с использованием программных средств, необходимо обговорить как при этом будет происходить добавление новых сущностей:

1. Получить с детальной страницы товара список характеристик в виде «название
   – текст» и других необходимых отдельных полей (ссылка, код товара, категория и т.д.);

2 Произвести поиск подкласса класса «Товар», соответствующего категории рассматриваемого товара. Если не найден, то создать;

3 Создать экземпляр полученного в п.2 класса (локальная часть IRI формируется как «Код производителя» или как «Код товара» при его отсутствии)

4 Для каждой характеристики

4.1. Проверить относится ли характеристика к описания какого-то из подклассов класса «Часть товара». Подклассы класса «Часть товара», а также отношения, связанные с ними заданы заранее и не изменяются в

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C00[illegible]435
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

процессе сбора информации. Для каждого из таких классов задаются списки полей и правила их преобразования. Конкретные реализации этой части не будут рассматриваться.

4.2. Если относится, то получить экземпляр (а при отсутствии создать) соответствующей части (формирования локальной части IRI для таких частей также прописано в правилах из п. 4.1.). Для полученного экземпляра задать отношение, связывающего его с рассматриваемой характеристикой.

4.3. Если не относится, то найти отношение по имени характеристики (при отсутствии создать отношение с таким именем как дочернее свойство для свойства «unknown» или типизированного свойства). Связать экземпляр товара с рассматриваемым значением характеристики полученным отношением.

Здесь проявляется гибкость онтологического подхода. В отличии от реляционной модели, в любой момент легко могут быть добавлены новые отношения, произведен перенос их в другую группу, произведено задание им новых ограничений. Также работа с описанием предметной области в целом ведется в терминах объектной или любой интересующей нас модели. При этом изменения, как правило, не ведут к долгому ожидания (т.к. чаще всего достаточно изменения нескольких троек).

Получив общее неформальное описание модели, можно приступать к ее формализации и реализации БЗ с использованием конкретного синтаксиса и/или инструментария. Эти процессы будут делаться параллельно с использованием системы Protégé.

Первоначально после запуска системы необходимо задать идентификатор онтологии на вкладке «Active ontology» в поле «Ontology IRI» (рис. 3б). Данный идентификатор также определяет и базовый префикс. Для изображенной онтологии IRI будет «http://myontologies.com/goods», а базовый префикс для всех создаваемых сущностей «http://myontologies.com/goods#»

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: [illegible]
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

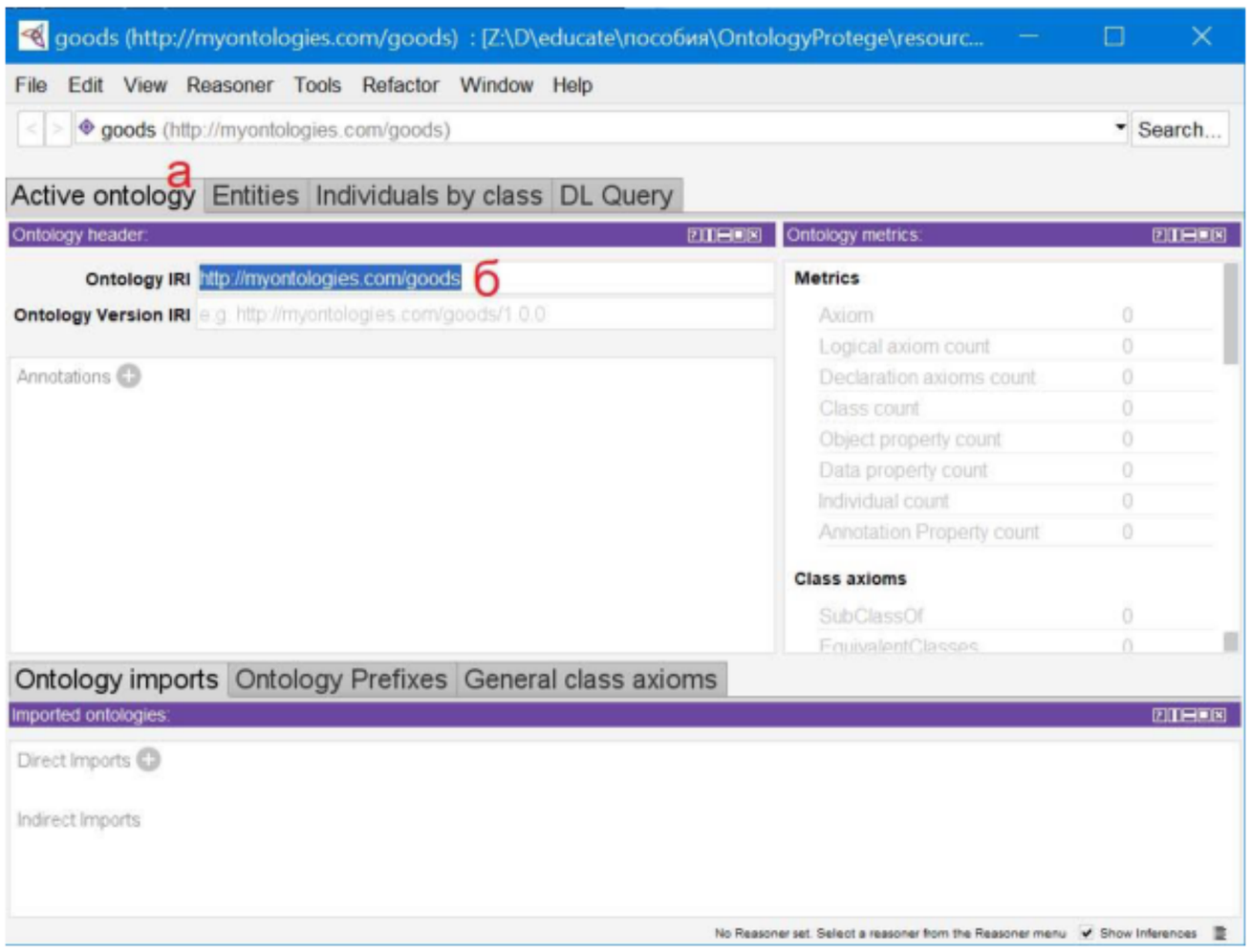


Рис. 4 Вкладка «Active ontology» в системе Protégé (а – панель в вкладок с активной вкладкой «Active ontology»; б – идентификатор онтологии)

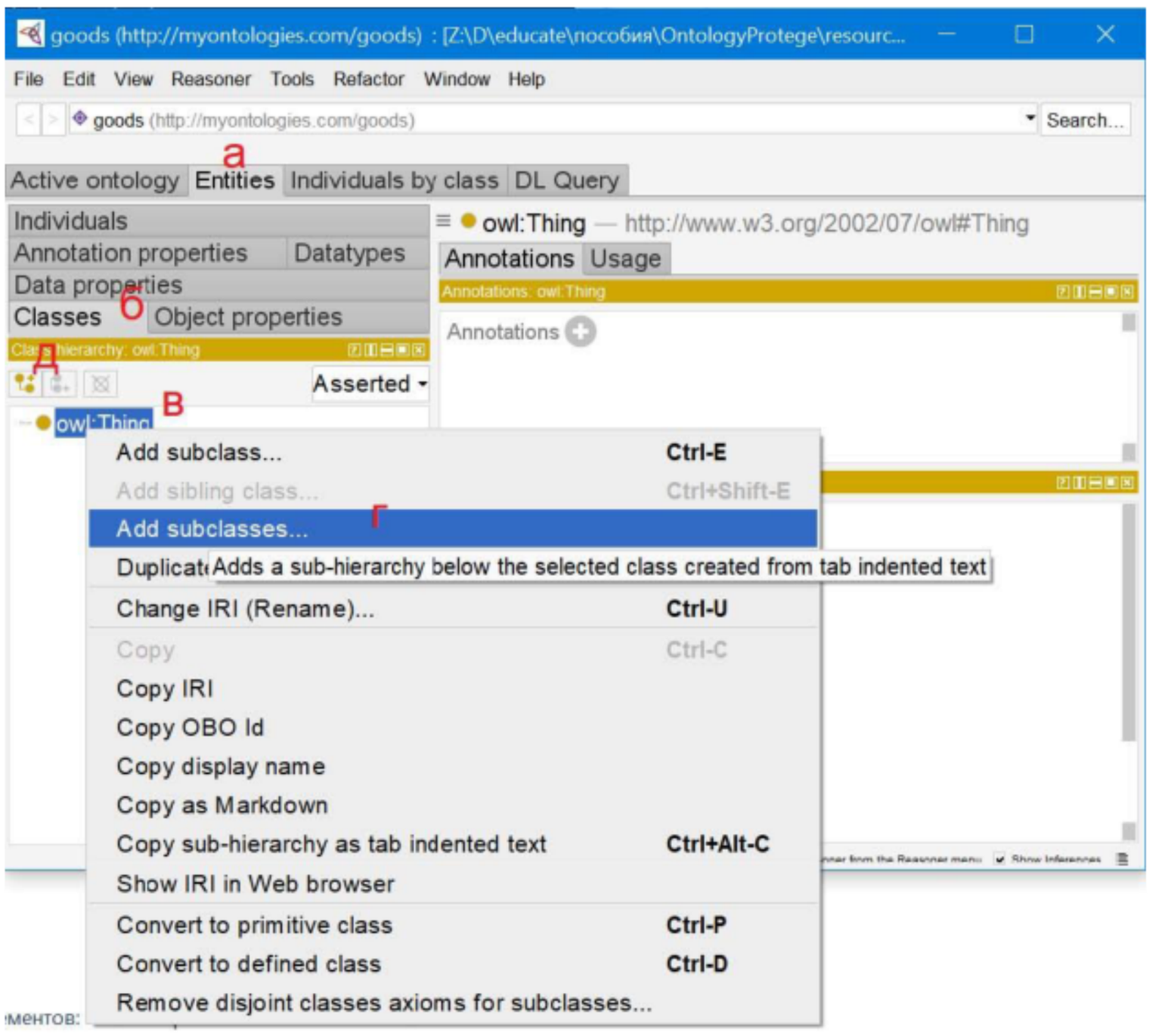


Рис. 5 Вкладка «Entities» в системе Protégé (а – панель в вкладок с активной вкладкой «Enitites»; б – панель вложенных вкладок вкладки «Entities» с активной вкладкой «Classes»; в – суперкласс owl:Thing; г, д – кнопки добавления иерархии подклассов/подкласса для выделенного класса)

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Далее необходимо создать иерархию классов. Для этого необходимо выделить класс, подклассами которого будут верхние классы нашей

иерархии (в нашем случае owl:Thing, рис. 5в). После этого вызывать контекстное меню (правой кнопкой мыши – ПКМ) и выбрать пункт «Add subclasses» (рис. 5г). Классы также могут добавляться по одному (рис. 5д).

При нажатии на кнопку «Add subclasses…» появляется окно добавления иерархии подклассов (рис. 6). Каждая введенная строка в данном окне создает отдельный класс (путем добавления в онтологию троек «x rdf:type owl:Class . x rdfs:subClassOf y»). Где x – IRI создаваемого класса, задаваемый как «{IRI онтологии}#{обработанное_имя_класса}» (в рамках обработки имени класса, происходит, например, замена пробела на «_»);y – IRI родительского класса, который задается как owl:Thing для классов, заданных без отступа и как IRI ближайшего верхнего класса с меньшим отступом для остальных (отступы задается как «Tab»).

После нажатия на «Continue» появляется окно, предлагающее создать для всех создаваемых классов одного уровня отношения «owl:disjointWith». Для этой задачи соглашаемся.

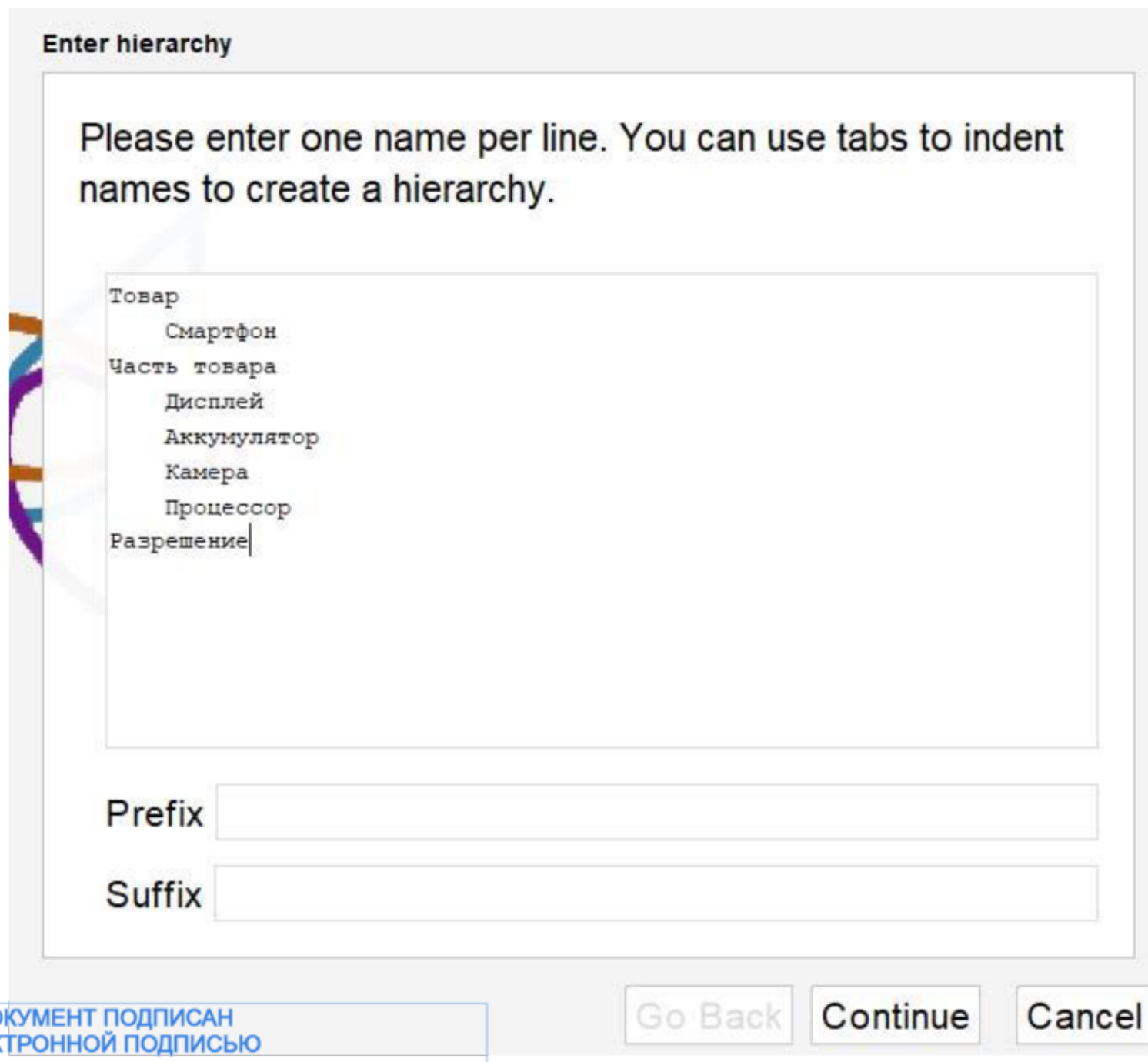


Рис. 6 Окно создания иерархии подклассов в системе Protégé

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Пока БЗ является достаточно простой, рассмотрим ее графическое представление и один из синтаксисов OWL/RDF. Для вывода графического представления необходимо активировать вкладку «Window -> Tabs -> OntoGraf» и перейти на нее (рис. 7). Здесь будут видны классы, экземпляры и object property. Рассмотрение текстового представления позволит нам увидеть какие именно тройки инструментальная система создала на данный момент и лучше понять как она работает (в некоторых случаях поведение системы может быть настроено, здесь рассматривается поведение по умолчанию). Для этого необходимо сохранить онтологию (File -> Save As). При сохранении на данном этапе необходимо выбрать синтаксис «Turtle Syntax».

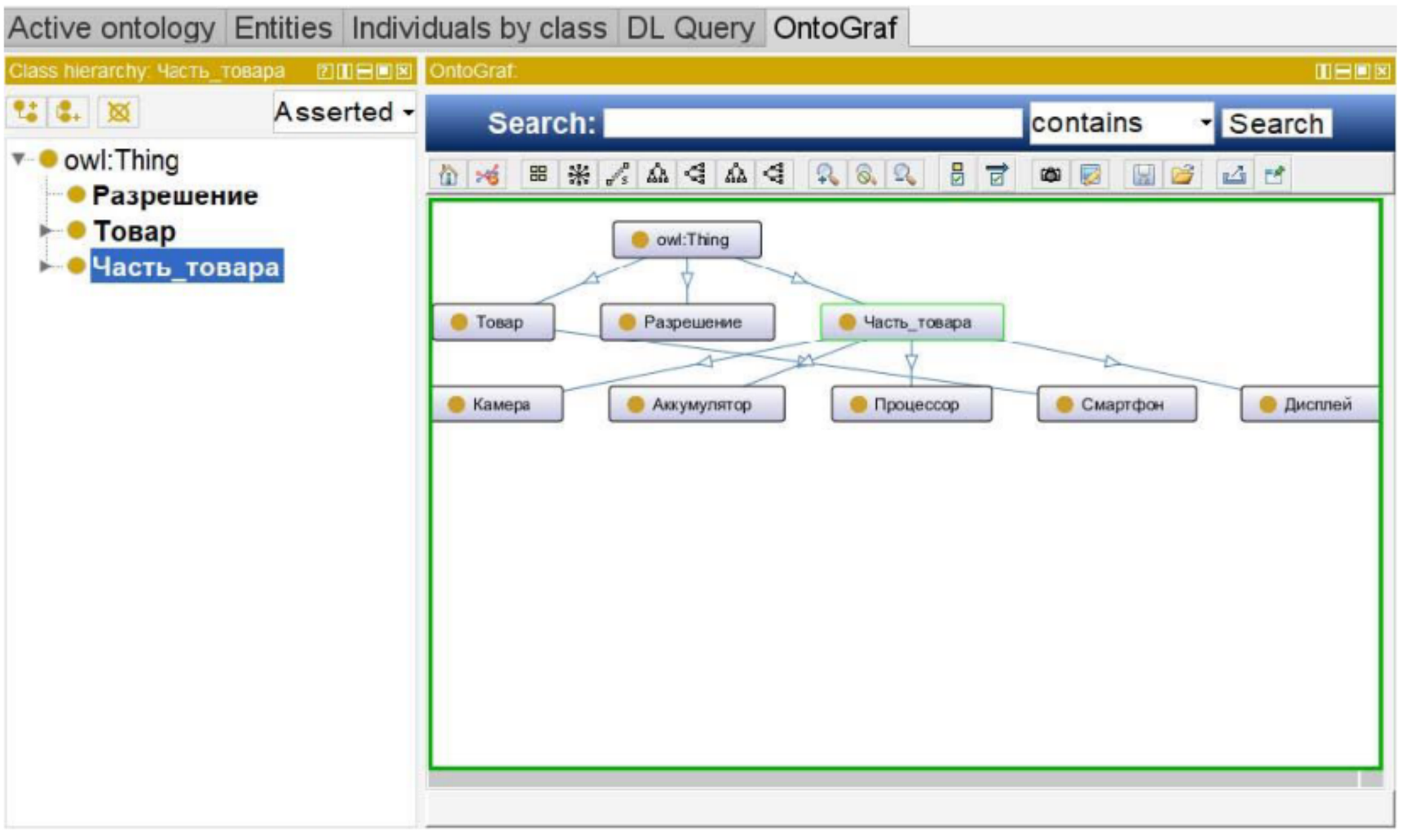


Рис. 7 Вкладка «OntoGraf» в системе Protégé

Открыв получившийся файл можно увидеть приведенный далее текст (приведены фрагменты с комментариями, объясняющими синтаксис и общую структуру онтологии. Комментарии начинаются с #).

#Описание начинается с объявления префиксов

#Определение общего префикса всех создаваемых сущностей (далее можно вместо него писать «:»

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0040013E8AB8B05220 5E7BA500B60000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

@prefix : <http://myontologies.com/goods#> .

#Определение префиксов owl, rdf, xml, xsd, rdfs

«.» является символом разделителем троек

@prefix owl: <http://www.w3.org/2002/07/owl#> .

...

#далее IRI нашей онтологии определяется как экземпляр онтологии

<http://myontologies.com/goods> rdf:type owl:Ontology .

#Далее следует описание созданных классов

#«;» является специальным разделителем троек и после него идет двойка «предикат – объект», а «субъект» берется из предыдущей тройки

:Аккумулятор rdf:type owl:Class ;

rdfs:subClassOf :Часть_товара .

#Т.е. это эквивалентно записи «:Аккумулятор rdf:type owl:Class . :Аккумулятор rdfs:subClassOf :Часть_товара»

#Для подклассов owl:Thing отношение rdfs:subClassOf может быть опущено при генерации файла

:Разрешение rdf:type owl:Class .

...

#Далее описываются общие аксиомы. В нашем случае это несколько условий на группы классов одного уровня, что никакие сущности не могут быть экземплярами одновременно нескольких классов из группы

[rdf:type owl:AllDisjointClasses ;

owl:members (:Аккумулятор :Дисплей :Камера :Процессор)] .

Преобразуем последнюю запись в вид троек для ее понимания, учитывая что: «[]» – позволяет задавать пустые узлы; «()» – позволяет задавать списки (для задания списков в RDF есть специальный класс rdf:List и отношения rdf:first, rdf:rest):

_x rdf:type owl:AllDisjointClasses .

_x owl:members _y .

_y rdf:type rdf:List . _y rdf:first :Аккумулятор . _y rdf:rest _e .

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C000004[illegible]000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

_e rdf:type rdf:List . _e rdf:first :Дисплей . _y rdf:rest _l .

_l rdf:type rdf:List . _l rdf:first :Камера . _y rdf:rest _k .

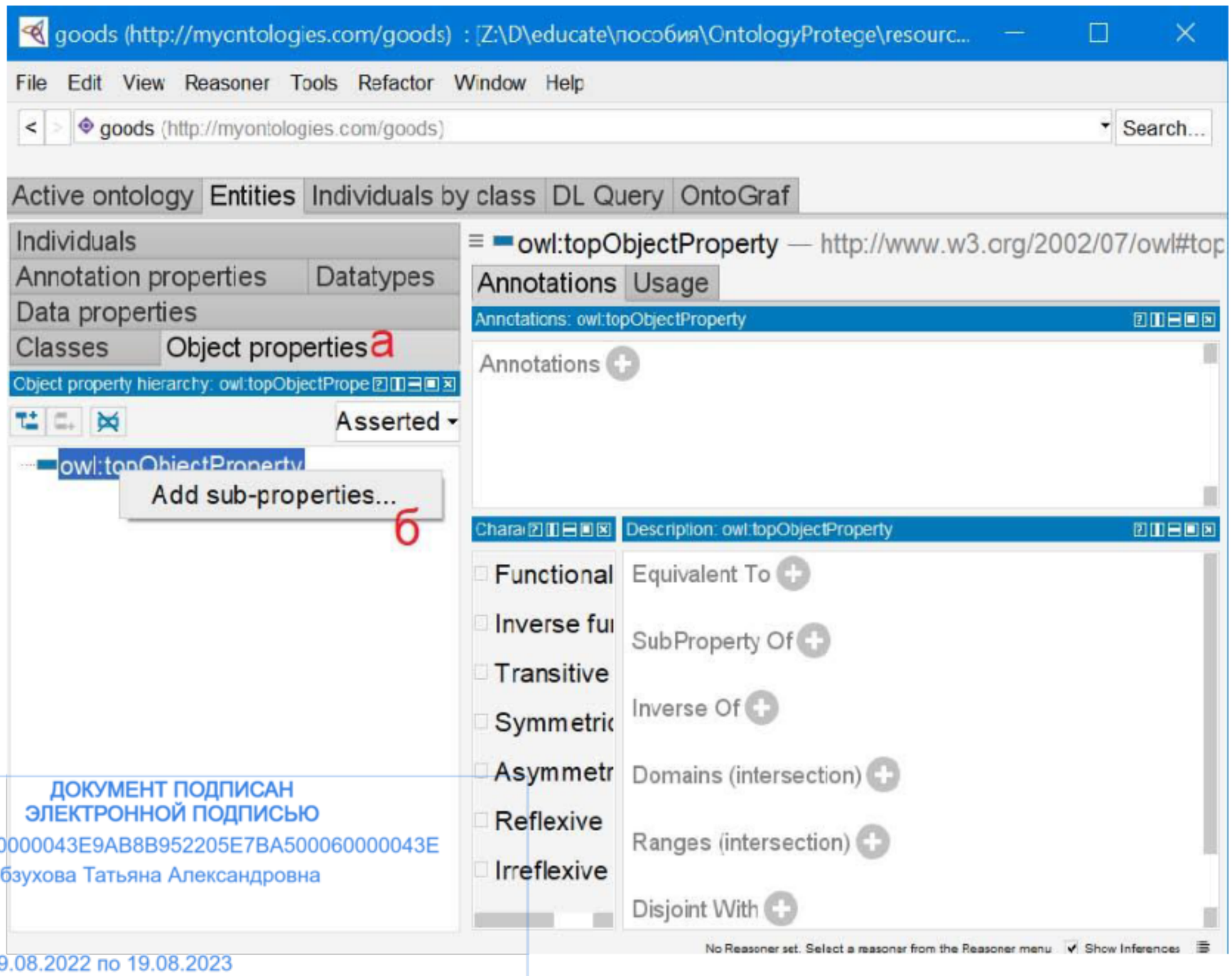
_k rdf:type rdf:List . _k rdf:first :Процессор . _y rdf:rest rdf:nill .

Это также может быть записано в функционально стиле: DisjointClasses(

:Аккумулятор :Дисплей :Камера :Процессор)

Понимание стандартов и синтаксиса облегчает дальнейшую работу с онтологией (в особенности работу с запросами и редактирование онтологии из программного кода). Кроме того, система Protégé реализует на уровне интерфейса не все операции (в частности, нет возможности явного создания пустых узлов, но при этом система в состоянии их понять в большинстве случаев). Также стоит отметить, что многие описания так или иначе придется писать самостоятельно в соответствии с определенным синтаксисом (defined classes, сложные ограничения на поля и т.д. реализуются в системе с использованием Manchester Syntax [7]).

Далее необходимо задать Object Properties. Они задаются аналогично иерархии классов на вкладке «Object Properties». (рис. 8)



ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Рис. 8 Добавление Object Properties (а – активная вкладка Object Properties, б– кнопка добавления иерархии отношений)

На данном этапе необходимо задать свойства «часть» и «разрешение». Далее опишем эти свойства подробнее, а именно зададим область определения и область значения.

Для свойства «часть» область определения «Товар», область значения «Часть_товара». Для того, чтобы задать эту информацию, необходимо выбрать свойство (рис. 9а), найти раздел «Domains» в появившемся описании и добавить новую запись (рис. 9б), выбрав соответствующий класс сущностей (рис. 9г).

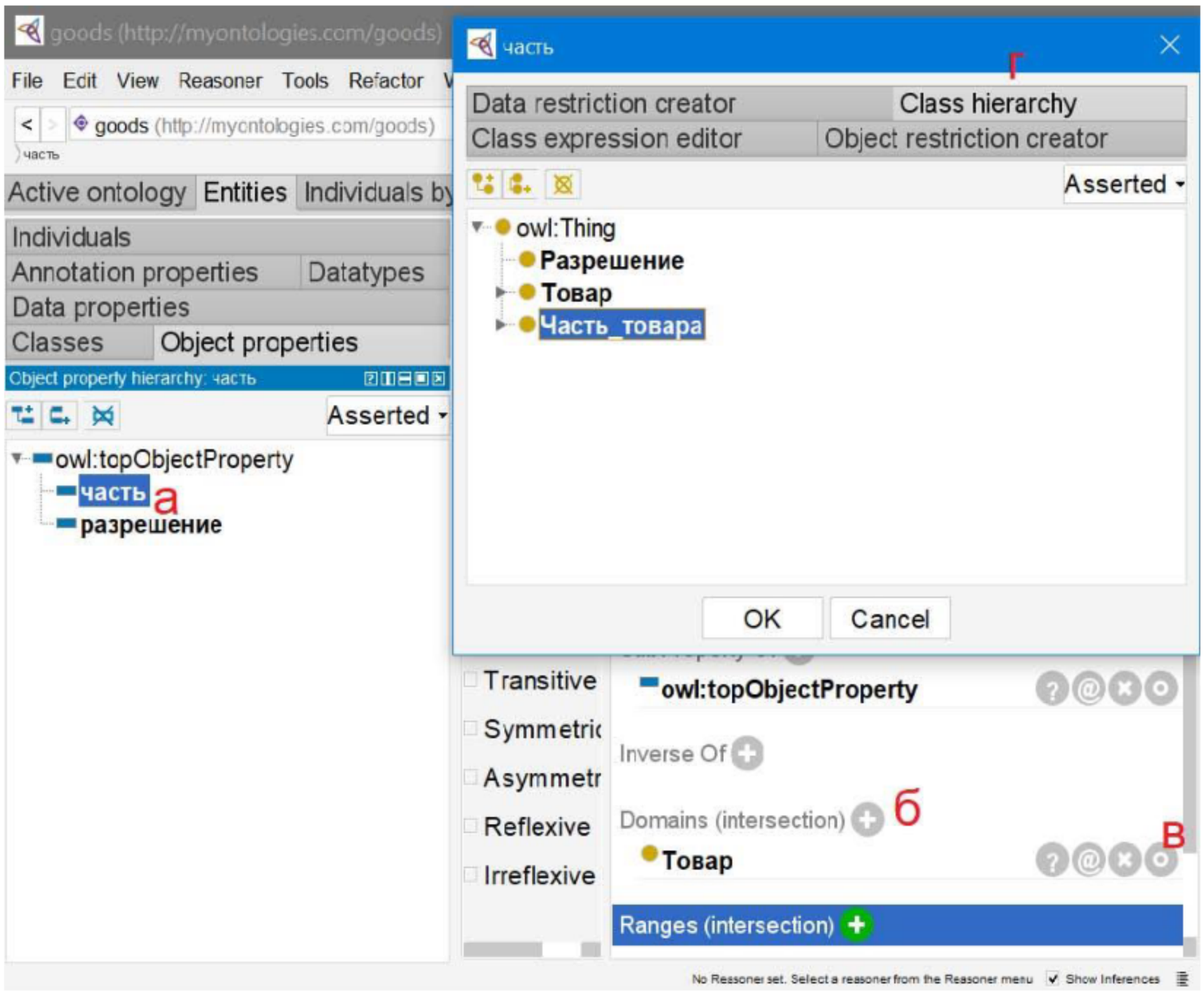


Рис. 9 Задание информации о Object Property в системе Protégé (а – выделенное свойство «часть», б – кнопка добавления новой области определения, в – кнопка редактирования существующей области определения, г – активная вкладка иерархии классов при задании области определения свойства)

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

При этом возможен не только выбор из существующих классов, но и задание через выражения на вкладках «Class expression editor», «Object restriction creator» (они являются одним из примеров, где придется использовать Manchester Syntax, а не просто графический интерфейс).

По аналогии создаются Data Properties. Полный список свойств, создаваемых в данной БЗ, приведен ниже (при создании необходимо заменить «;» на переносы строк и отступ)

текстовое

код производителя; цвет заявленный производителем; сеть 2g; сеть 3g; частота lte; формат sim карты; технология изготовления экрана; материал корпуса; защита корпуса; защита экрана; степень защиты; операционная система; производитель процессора; модель процессора; конфигурация процессора; графический ускоритель; тип вспышки основной камеры

числовое

год выпуска; число sim карт; диагональ экрана; разрешение горизонталь; разрешение вертикаль; разрешение кадр/сек; плотность пикселей; количество цветов экрана млн; количество ядер; частота процессора ггц; объем оперативной памяти гб; объем встроенной памяти гб; максимальный объем карты памяти гб; количество основных камер; количество мегапикселей основной камеры; апертура основной камеры; количество мегапикселей фронтальной камеры; количество фронтальных камер; емкость аккумулятора мА*ч; время работы в режиме разговора; время работы в режиме ожидания; ширина; высота; толщина; вес

логическое

автофокусировка основной камеры; оптическая стабилизация основной камеры; наличие вспышки фронтальной камеры; наличие беспроводной зарядки

свободное

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: [illegible]0060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

конфигурация процессора; датчики; особенности основной камеры; комплектация; биометрическая защита; особенности

Для них также необходимо задать область определения и область значения. Стоит отметить, что для верхних свойств в иерархии будет задаваться только область значения, т.к. такие отношения как «числовое» могут связывать любой «субъект», но только с числом. Окно задание области значения для data properties отличается от аналогичного для object property (рис.10). Для свойств «числовое», «свободное», «текстовое», «логическое» область значения задается с использованием вкладки «Built in datatypes», на которой выбираются типы «xsd:double», «xsd:string», «xsd:boolean».

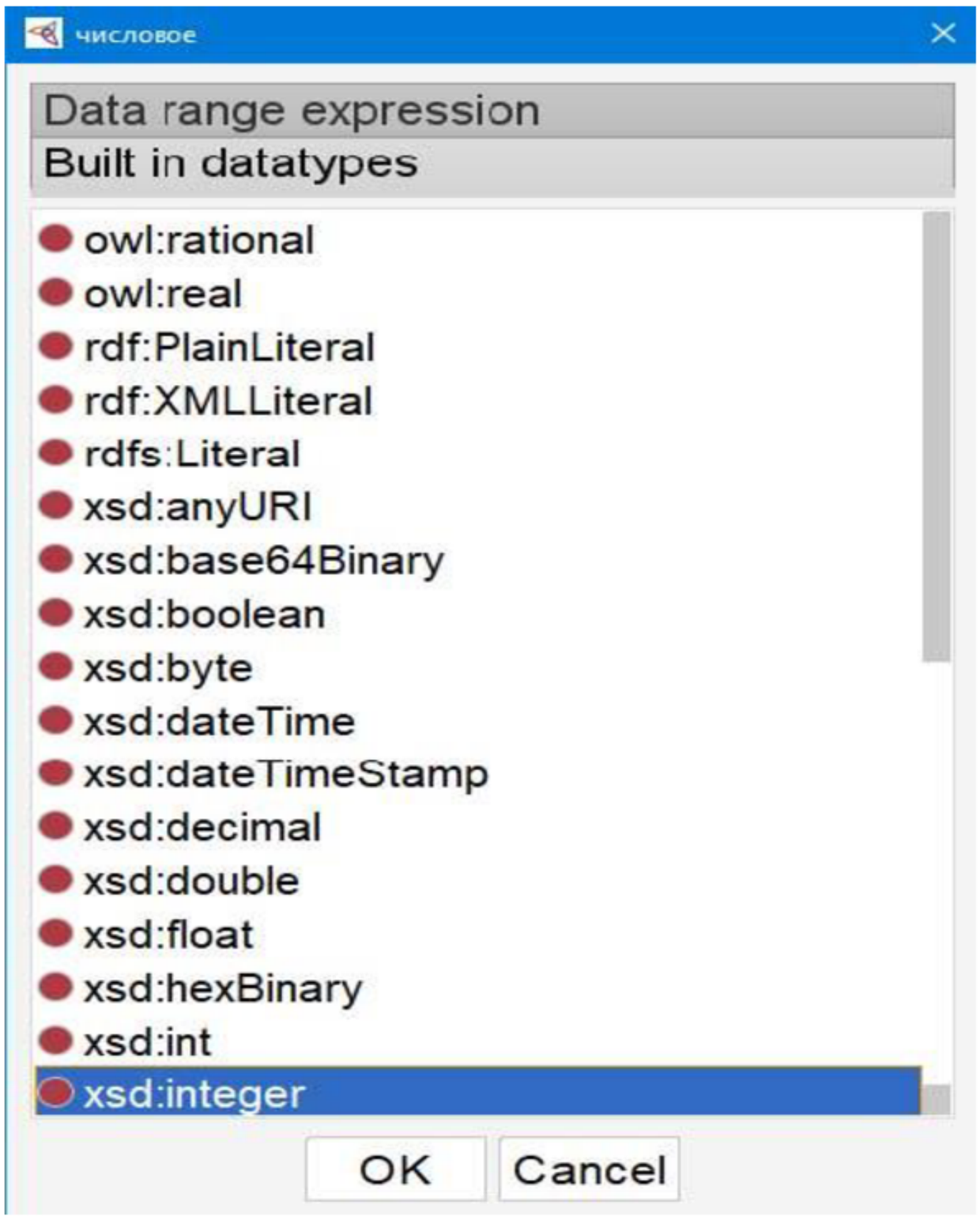


Рис. 10 Окно задания области значения для data property в системе Protégé

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB6B952265EBA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Для некоторых свойств необходимо задать дополнительные ограничения. Для этого используется вкладка «Data range expression» и

Manchester Syntax. Разберем некоторые из свойств. Форматы сим карт стандартизированы: Полноразмерные (1FF), Mini-SIM (2FF), Micro-SIM (3FF), Nano-SIM (4FF), Встроенные (E-SIM). Потому можно задать дополнительное ограничение на свойство «формат sim карты» (рис. 11).

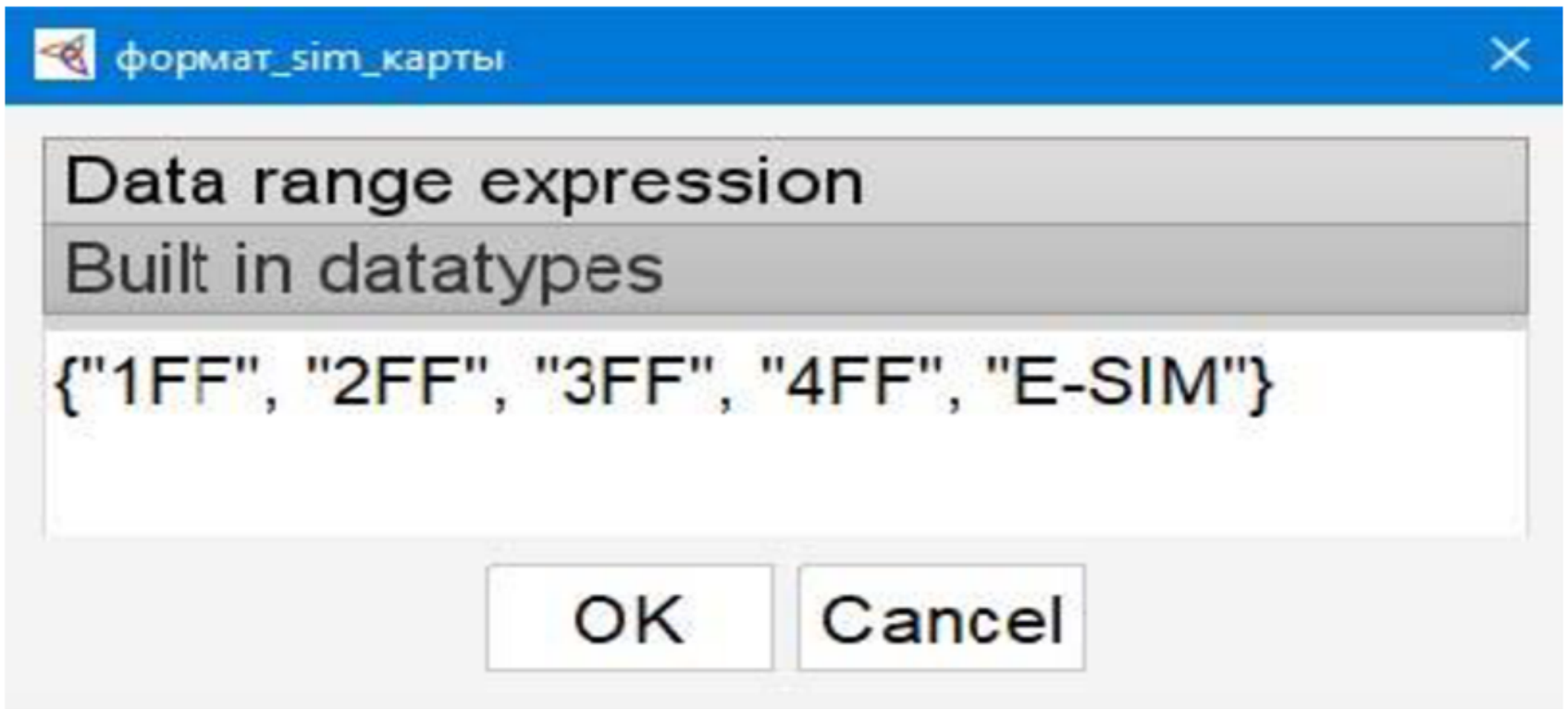


Рис. 11 Задание дополнительных ограничений на свойство «формат sim карты

Рассмотрим, что при этом создается в файле онтологии:

:формат_sim_карты rdf:type owl:DatatypeProperty ;

rdfs:subPropertyOf :текстовое ;

rdfs:range [rdf:type rdfs:Datatype ; owl:oneOf ("1FF" "2FF" "3FF" "4FF" "E-SIM")] .

Как видно, для описания области значений создается пустой узел, который является типом данных (экземпляром rdfs:Datatype) и его экземпляры могут принимать значения из заданного списка (за счет использования отношения owl:oneOf). При использовании ограничений на data property пустой узел rdfs:Datatype создается всегда, дальнейшие конструкции зависят от введенного разработчиком. Рассмотрим возможные конструкции для этого поля подробнее:

1 { } – задает список значений, в рамках которых могут быть только скалярные значения (возможно использовать задание типа данных через «^^» и теги языков через «@»)

2 Любой тип данных (xsd:integer, xsd:string и т.д.)

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

3 Любое скалярное значение

4 [] – ограничение на значение, которые идут после типа значения. Например, xsd:interger[> 0] соответствует целым строго положительным числам. Ограничения прописываются в неограниченном количестве через запятую, например: xsd:integer[> 0 , < 100]. Ограничения, которые могут понадобиться в работах: все знаки сравнения (>, <, >=, <=), проверка строки по регулярному выражению (xsd:string[pattern “[0-9]{4}”]. Необходимо отметить, что регулярное выражение в таком виде проверяет соответствие всей строки, а не ищет подобные вхождения. Т.е. аналогично выражению «^[0-9]{4}$» в обычной ситуации).

5 Логические операторы: or, and. Например, «xsd:integer or xsd:double».

Рассмотрим свойство «год выпуска». Он должен быть целым четырехзначным числом не меньшим 1973 или целым не более чем двузначным числом (в случае, если учитываем возможность внесения сокращенной записи «20», «21», а не делаем преобразование заранее). Пользуясь новыми возможностями, изменим область значений свойства «числовое» на «xsd:integer or xsd:double or xsd:decimal». Далее зададим свойству «год выпуска» ограничение «xsd:integer[>1973] or xsd:integer[>=0 , <= 99]». Также рассмотрим, как это выражение преобразуется в RDF синтаксис (на примере Turtle):

```
:год_выпуска rdf:type owl:DatatypeProperty ;
rdfs:subPropertyOf :числовое ;
rdfs:range [ rdf:type rdfs:Datatype ;
        owl:unionOf ( [ rdf:type rdfs:Datatype ; owl:onDatatype xsd:integer ;
          owl:withRestrictions ( [ xsd:minInclusive 0 ] [ xsd:maxInclusive 99 ] )
]
        [ rdf:type rdfs:Datatype ; owl:onDatatype xsd:integer ;
          owl:withRestrictions ( [ xsd:minExclusive 1973] ) ] ) ] .
```

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043[illegible]
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Приведенное выражение несколько сложнее Manchester Syntax, но позволяет получить понимание как работают выражения в терминах RDF и как они, соответственно, потом обрабатываются.

Создаются 2 пустых узла с отношениями xsd:minInclusive 0, xsd:maxInclusive 99, которые объединяются в RDF список (rdf:List через сокращенное описание «()»). Далее создается новый пустой узел – тип данных, основанный на xsd:integer (owl:onDatatype), которые соединяется отношением owl:withRestrictions с созданным ранее списком. Полностью аналогично создается еще один тип данных, основанный на integer, но с другими ограничениями. После этого создается еще один новый тип данных, которые определяется как объединение (owl:unionOf) созданных ранее. И именно он является областью определения нашего свойства. При этом по факту, созданные типы интересуют нас только с точки зрения возможных значений и могут рассматриваться как множества.

На остальные свойства ограничения могут быть наложены аналогично. Как минимум необходимо заполнить области определения и значения для свойств, связывающих части товара и конкретные характеристики («материал корпуса», «количество ядер» и т.д.). Остальное на данном этапе можно оставить как есть и переходить к экземплярам. Они создаются на вкладке «Individuals». (рис. 12)

На данной вкладке экземпляры создаются по одному. Для этого необходимо нажать на кнопку добавления (рис. 12б) и ввести локальное имя (на основании которого с использование IRI онтологии будет сгенерирован полный IRI), полный IRI или идентификатор с использованием префиксов. Создадим экземпляр рассматриваемого смартфона с идентификатором «SM-G965FZPHSER». После того как экземпляр создан, появляется окно его описания (рис. 13).

Нам необходимо задать экземпляром какого класса является созданная сущность (построить отношение rdf:type). Для этого необходимо добавить запись в раздел «Types» (рис 13а), где, использовав вкладку «Class

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: [illegible]
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Hierarchy», выбрать класс «Смартфон».

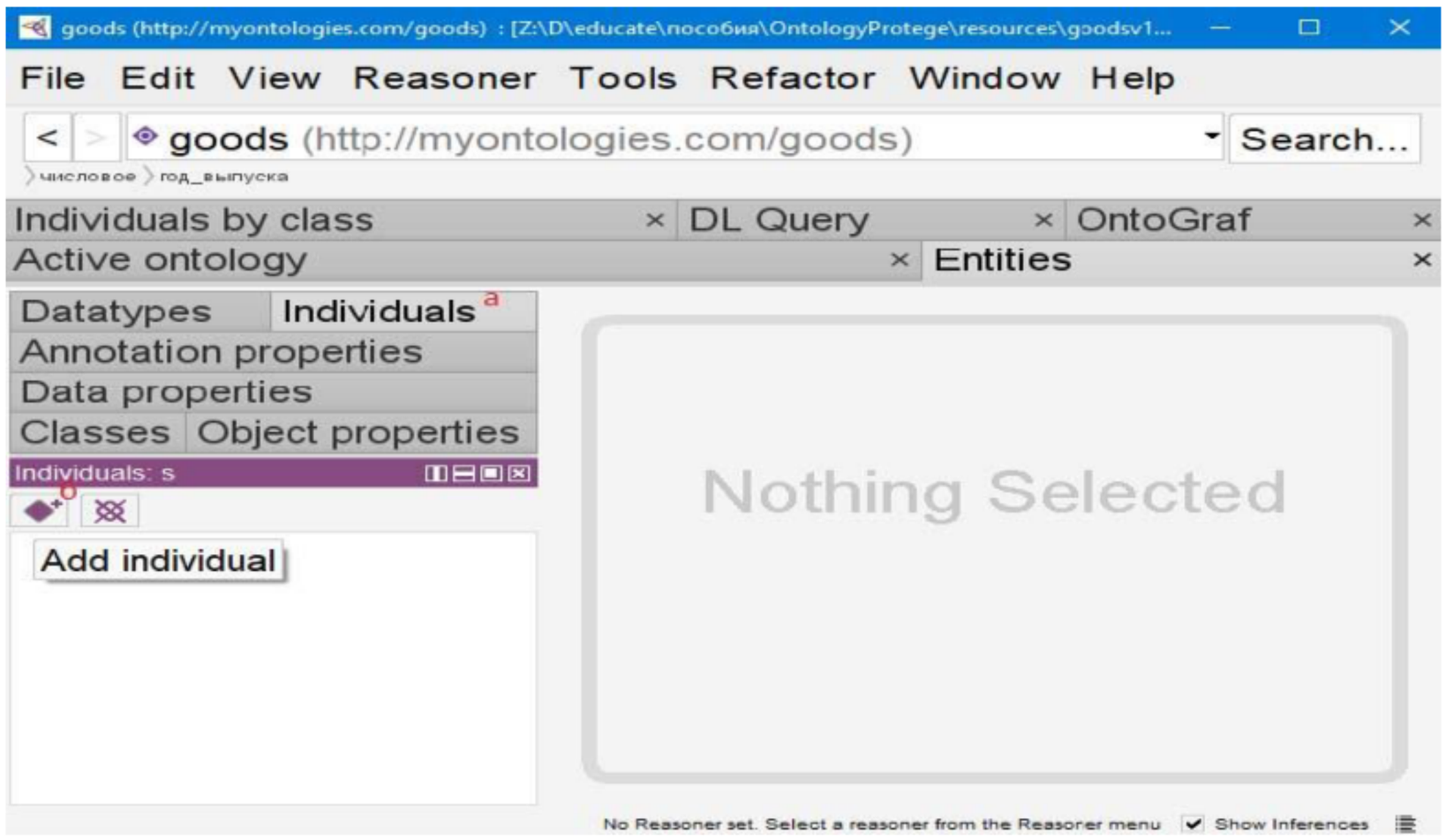


Рис. 12 Вкладка «Individuals» в системе Protégé (а – активная вкладка Individuals, б

– кнопка создания нового экземпляра)

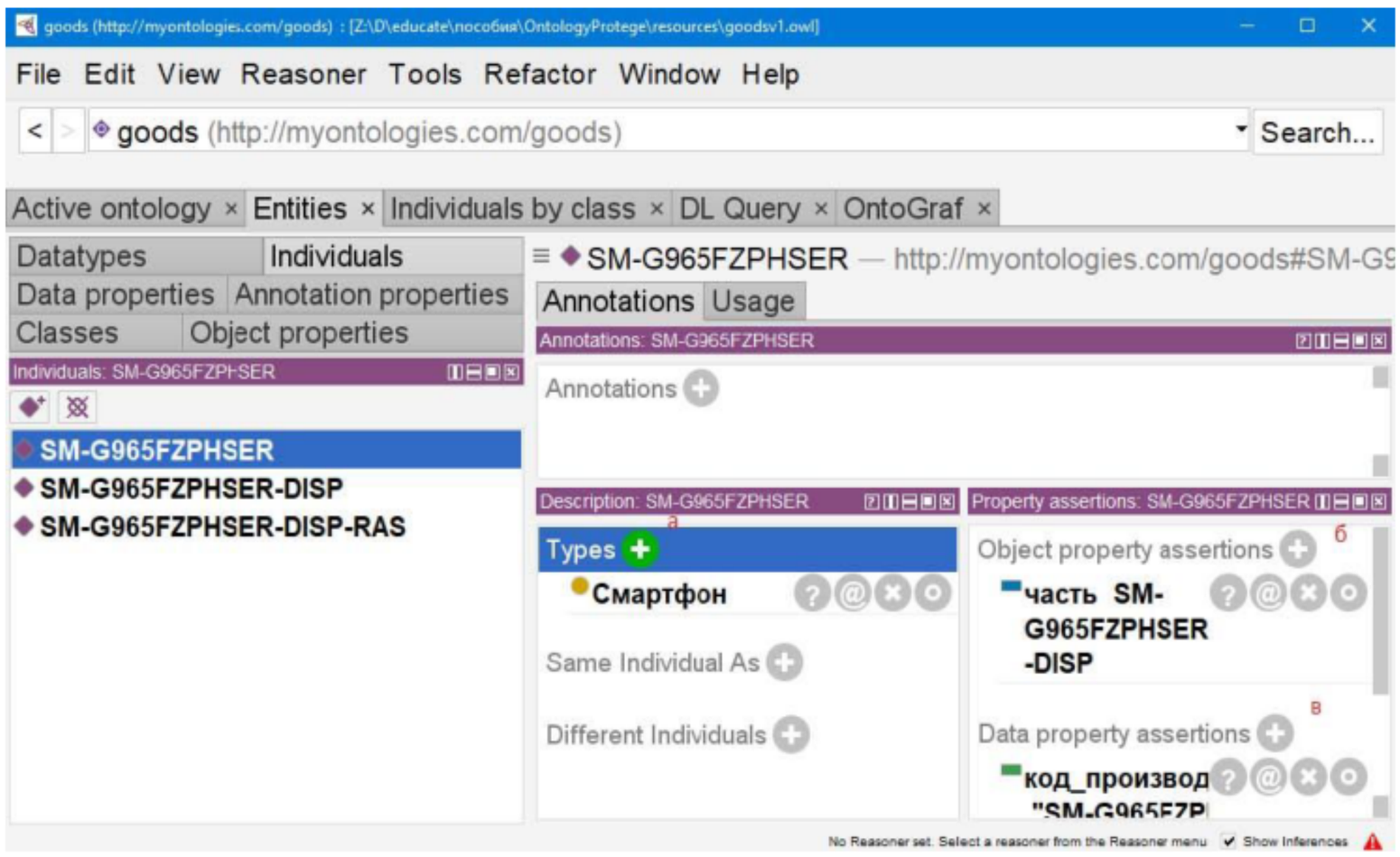


Рис. 13 Окно описания экземпляра в системе Protégé (а – задание отношения rdf:type, б – задание отношений с использование object property, в – задание отношений с использованием data property)

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: [illegible]
Действителен: с 16.08.2022 по 19.08.2023

Далее необходимо задать отношения для рассматриваемого смартфона. Первым создадим отношение «часть» с его дисплеем. Для этого по аналогии создадим новый экземпляр класса «Дисплей». После чего для экземпляра

смартфона добавим новую запись в раздел «Object property assertions» (рис. 13б). При добавлении необходимо ввести имя отношения и имя экземпляра, с которым строится
отношение (рис. 14).

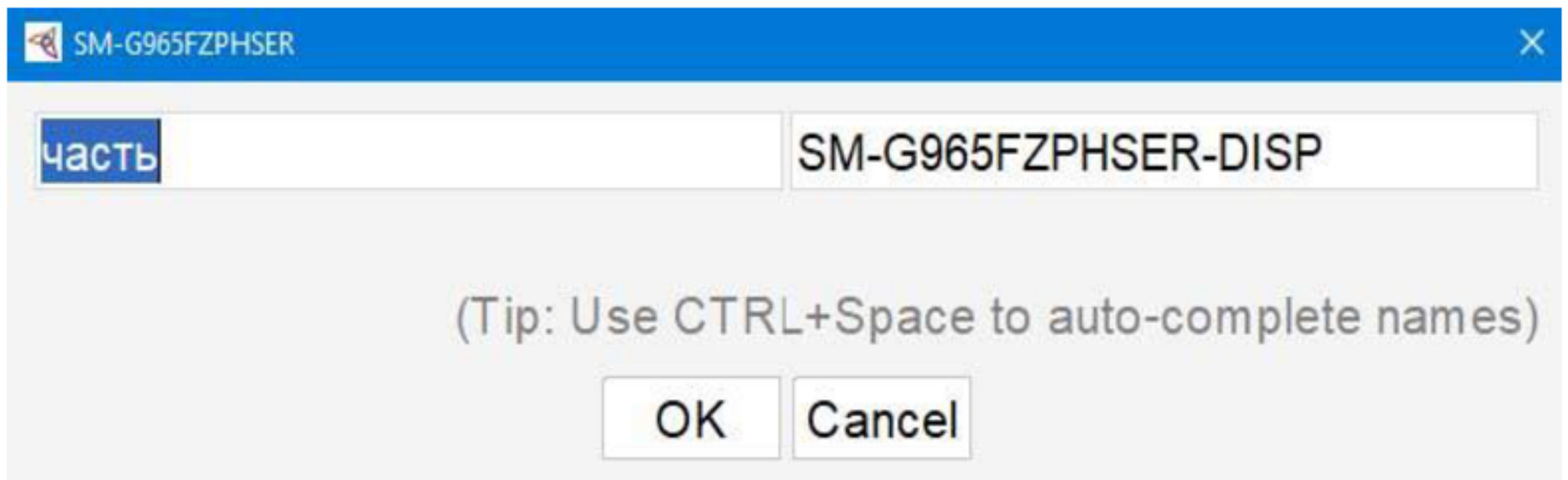


Рис. 14 Окно задания object property для экземпляра

После этого зададим отношение в разделе data properties на примере «код производителя» (рис. 15).

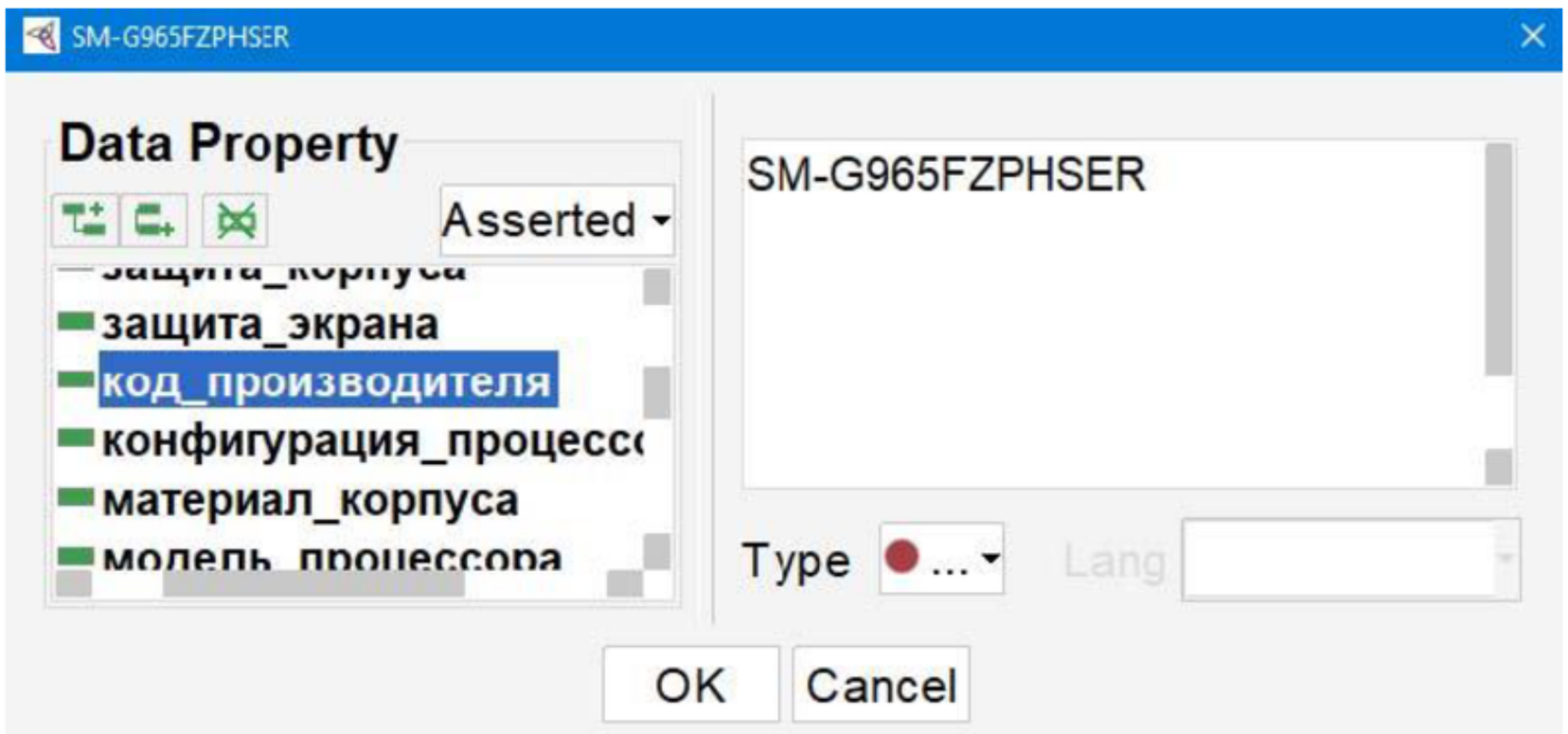


Рис. 15 Окно задания data property для экземпляра

Для этого в появившемся окне необходимо выбрать само свойство, ввести его значение и выбрать тип.

Для определенных типов также активируется поле выбора языка.

По аналогии создаем экземпляр «Разрешение» и задаем дисплею разрешение.

Рассмотрим текстовое представление созданных сущностей:

:SM-G965FZPHSER rdf:type owl:NamedIndividual , :Смартфон ;

:часть :SM-G965FZPHSER-DISP ;

:код_производителя "SM-G965FZPHSER"^^xsd:string .

:SM-G965FZPHSER-DISP rdf:type owl:NamedIndividual , :Дисплей ;

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E0A0D00B000000043
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

:разрешение :SM-G965FZPHSER-DISP-RAS .

:SM-G965FZPHSER-DISP-RAS rdf:type owl:NamedIndividual , :Разрешение ;

:разрешение_вертикаль 1920 ;

:разрешение_горизонталь 1080 .

Новая синтаксическая конструкция «,» означает, что после нее будет записан только «объект» из тройки, а «субъект – предикат» будут взяты из предыдущей. «owl:NamedIndividual» указывает, что сущность является экземпляром и не пустым узлом. Однако, на данном этапе, при удалении этого отношения, в системе Protégé не будет видна разница. Дальнейшее описание достаточно большого количества экземпляров и их отношений проще сделать напрямую в файле онтологии с использованием любого текстового редактора. При этом можно было бы заменить некоторые экземпляры на пустые узлы:

:SM-G965FZPHSER-DISP rdf:type owl:NamedIndividual ,

:Дисплей ; :разрешение [rdf:type :Разрешение ;

:разрешение_вертикаль 1920 ; :разрешение_горизонталь

1080] .

Система будет понимать такое описание, однако в интерфейсе не будет указываться сам экземпляр класса «Разрешение» и, как следствие, не будет возможности редактирования его описания. При этом все возможности обработки, в том числе вывод в рамках системы Protégé остаются возможными. Тем не менее, для большей наглядности при открытии онтологии в системе, далее будет приведен текст экземпляров и их отношений без использования таких пустых узлов.

:SM-G965FZPHSER rdf:type owl:NamedIndividual , :Смартфон ;

:часть :SM-G965FZPHSER-DISP;

:часть :Exynos_9810;

:часть :SM-G965FZPHSER-CAM-BACK;

:часть :SM-G965FZPHSER-CAM-FRONT;

:часть :SM-G965FZPHSER-BAT;

:код_производителя "SM-G965FZPHSER";

:цвет_заявленный_производителем "ультрафиолет";

:сеть_2g "GSM 1900", "GSM 1800", "GSM 900";

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B53205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 18.08.2023

:сеть_3g "WCDMA 900", "WCDMA 2100";
:частота_lte "LTE 1800 (B3)", "LTE 2100 (B1)", "LTE 2300 (B40)", "LTE 2600 (B7)", "LTE 1900 (B39)", "LTE 2000 (B34)";
:формат_sim_карты "4FF";
:степень_защиты "IP68";
:операционная_система "Android 8.0 Oreo";
:год_выпуска 18;
:число_sim_карт 2;
:объем_оперативной_памяти_гб 6;
:объем_встроенной_памяти_гб 256;
:максимальный_объем_карты_памяти_гб 400;
:материал_корпуса "металл", "стекло";
:защита_корпуса "защита от пыли и влаги";
:ширина 73.8;
:высота 158.1;
:толщина 8.5;
:вес 189;
:датчики "гироскоп, датчик Холла, акселерометр, барометр, компас, датчик приближения, датчик освещения, датчик сердечного ритма";
:комплектация "комплект амбушюр, комплект переходников, документация, сетевой адаптер, скрепка для извлечения слота SIM-карты, наушники";
:биометрическая_защита "сканер лица, сканер радужной оболочки глаза, сканер отпечатков пальцев";
:особенности "ANT+, технология Dolby Atmos, камера Bixby, камераперевoдчик, селфимоджи, стереодинамики от AKG, несъемный аккумулятор" .

:SM-G965FZPHSER-DISP rdf:type owl:NamedIndividual , :Дисплей ;
:разрешение :SM-G965FZPHSER-DISP-RAS;
:технология_изготовления_экрана "Super AMOLED";
:защита_экрана "Corning Gorilla Glass";
:диагональ_экрана 6.2;
:плотность_пикселей 529;
:количество_цветов_экрана_млн 16 . :SM-G965FZPHSER-DISP-RAS rdf:type owl:NamedIndividual ;
:разрешение_вертикаль 1920 ;
:разрешение_горизонталь 1080 .
:Exynos_9810 rdf:type owl:NamedIndividual , :Процессор ;
:производитель_процессора "Samsung" ;
:модель_процессора "Exynos 9810" ;
:конфигурация_процессора "4x Custom 2.9 ГГц, 4x Cortex-A55 1.9 ГГц" ;
:количество_ядер 8 ; :частота_процессора_ггц 2.9, 1.9 ;
:графический_ускоритель "Mali-G72 MP18" .
:SM-G965FZPHSER-CAM-BACK rdf:type owl:NamedIndividual , :Камера ;

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E0A8BB5B2C5E7BA500030000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

:количество_основных_камер 2 ;
:количество_мегапикселей_основной_камеры 12, 12 ;
:апертура_основной_камеры 1.5, 2.4 ;
:тип_вспышки_основной_камеры "светодиодная" ;
:разрешение :SM-G965FZPHSER-DISP-CAM-BACK-RAS-1, :SM G965FZPHSER-DISP-CAM-BACK-RAS-2, :SM-G965FZPHSER-DISP-CAM-BACKRAS-3;
:автофокусировка_основной_камеры 1 ;
:оптическая_стабилизация_основной_камеры 1 ;
:особенности_основной_камеры "сверхбыстрый датчик изображения, двойная апертура объектива" .
:SM-G965FZPHSER-DISP-CAM-BACK-RAS-1 rdf:type owl:NamedIndividual ;
:разрешение_вертикаль 3840 ;
:разрешение_горизонталь 2160 ;
#полный IRI необходим из-за спец символа "/"
<http://myontologies.com/goods#разрешение_кадр/сек> 60 .
:SM-G965FZPHSER-DISP-CAM-BACK-RAS-2 rdf:type owl:NamedIndividual ;
:разрешение_вертикаль 1280 ;
:разрешение_горизонталь 720 ;
#полный IRI необходим из-за спец символа "/"
<http://myontologies.com/goods#разрешение_кадр/сек> 960 .
:SM-G965FZPHSER-DISP-CAM-BACK-RAS-3 rdf:type owl:NamedIndividual ;
:разрешение_вертикаль 1920 ;
:разрешение_горизонталь 1080 ;
#полный IRI необходим из-за спец символа "/"
<http://myontologies.com/goods#разрешение_кадр/сек> 240 .
:SM-G965FZPHSER-CAM-FRONT rdf:type owl:NamedIndividual , :Камера ;
:количество_мегапикселей_фронтальной_камеры 8;
:количество_фронтальных_камер 1 ; :разрешение :SM-G965FZPHSER-DISP-CAM-FRONT-RAS-1 ; :наличие_вспышки_фронтальной_камеры 0 .
:SM-G965FZPHSER-DISP-CAM-FRONT-RAS-1 rdf:type owl:NamedIndividual , :Разрешение ;
:разрешение_вертикаль 2560 ;
:разрешение_горизонталь 1440 ;
#полный IRI необходим из-за спец символа "/"
<http://myontologies.com/goods#разрешение_кадр/сек> 30 .
:SM-G965FZPHSER-BAT rdf:type owl:NamedIndividual , :Аккумулятор ;
#полный IRI необходим из-за спец символа "*"
<http://myontologies.com/goods#емкость_аккумулятора_мА*ч> 3500 ;
:время_работы_в_режиме_разговора 25 ;
:наличие_беспроводной_зарядки 1 .

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0060043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Теперь основные данные внесены и можно проверить онтологию на непротиворечивость. Для этого необходимо выбрать механизм вывода HermiT Reasoner
[5] (рис. 16б) и запустить его (рис. 16в).

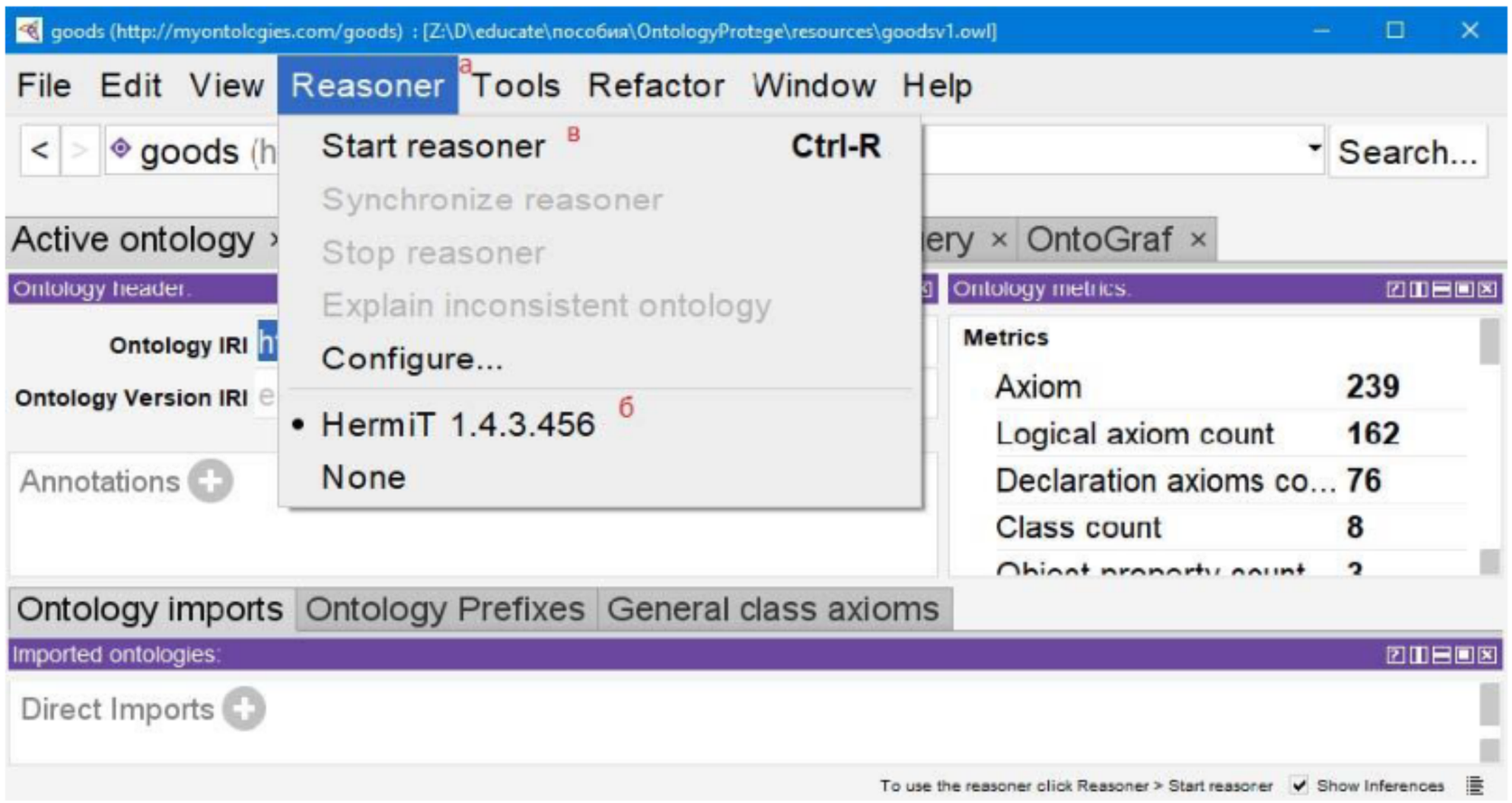


Рис. 16 Запуск механизма вывода в системе Protégé (а – пункт меню с действиями вывода, б – выбранный механизм HermiT, в – кнопка запуска вывода).

Индикацией того, что вывод запущен в данном случае будет то, что кнопка «Start reasoner» станет неактивной, а «Stop reasoner» наоборот активной. Если в процессе вывода были найдены противоречия, то система выведет сообщения об ошибках с их объяснением (рис. 17а) или выведет сообщение об ошибке запуска вывода (рис. 17в). При выполнении всех приведенных выше действий и рекомендаций ошибок быть не должно.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

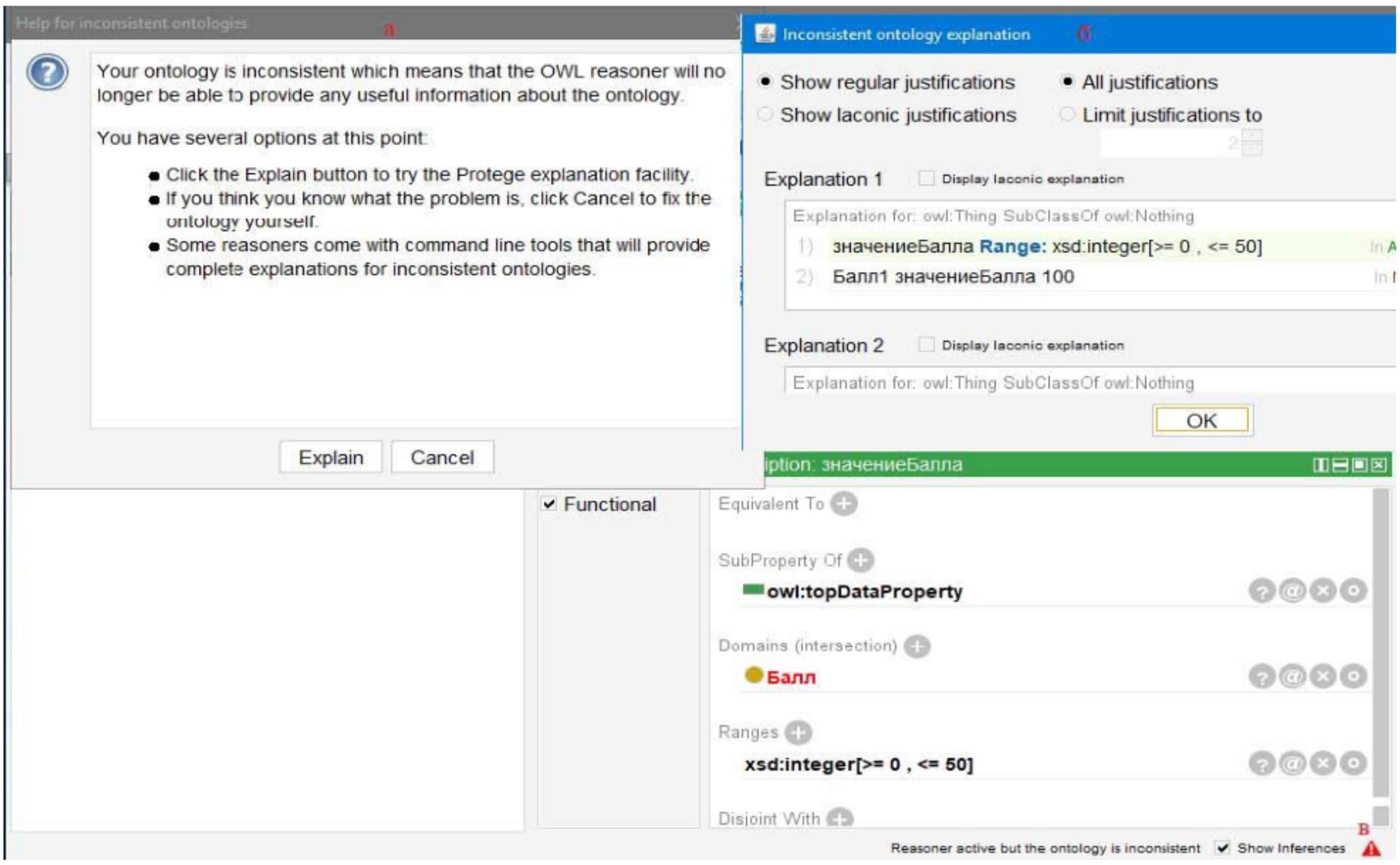


Рис. 17 Пример ошибки при запуске механизма рассуждений (а – сообщение о проблемах в онтологии с предложением объяснения; б – окно объяснений; в – индикатор ошибок в логах системы)

Также, если внимательно изучить приведенный выше фрагмент с экземплярами онтологии, то мы намеренно не задали тип для таких сущностей как «G965FZPHSER-DISP-CAM-BACK-RAS-2». Если при этом читатель выполнял все рекомендации и заполнил область определения для таких отношений как «разрешение_горизонталь», то можно увидеть, что тип был выведен автоматически (рис. 18). Выведенные (Inferred) понятия отмечаются в системе Protégé желтым цветом.

Для удобства дальнейшей обработки дополним онтологию новым object property: «является_частью». Это свойство является обратным к «часть», что необходимо отметить в его описании (рис. 19а). Прописывать его к созданным сущностям нет необходимости, т.к. оно будет выведено (рис. 19б).

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

В процессе этой работы было создана первая версия онтологии, декларативно описывающей предметную область «товар смартфон». В следующих работах в онтологию будут добавлены новые аксиомы, выражения и правила для вывода отношений и проверки непротиворечивости БЗ, а также написаны запросы для выбора информации из нее по определенным критериям.

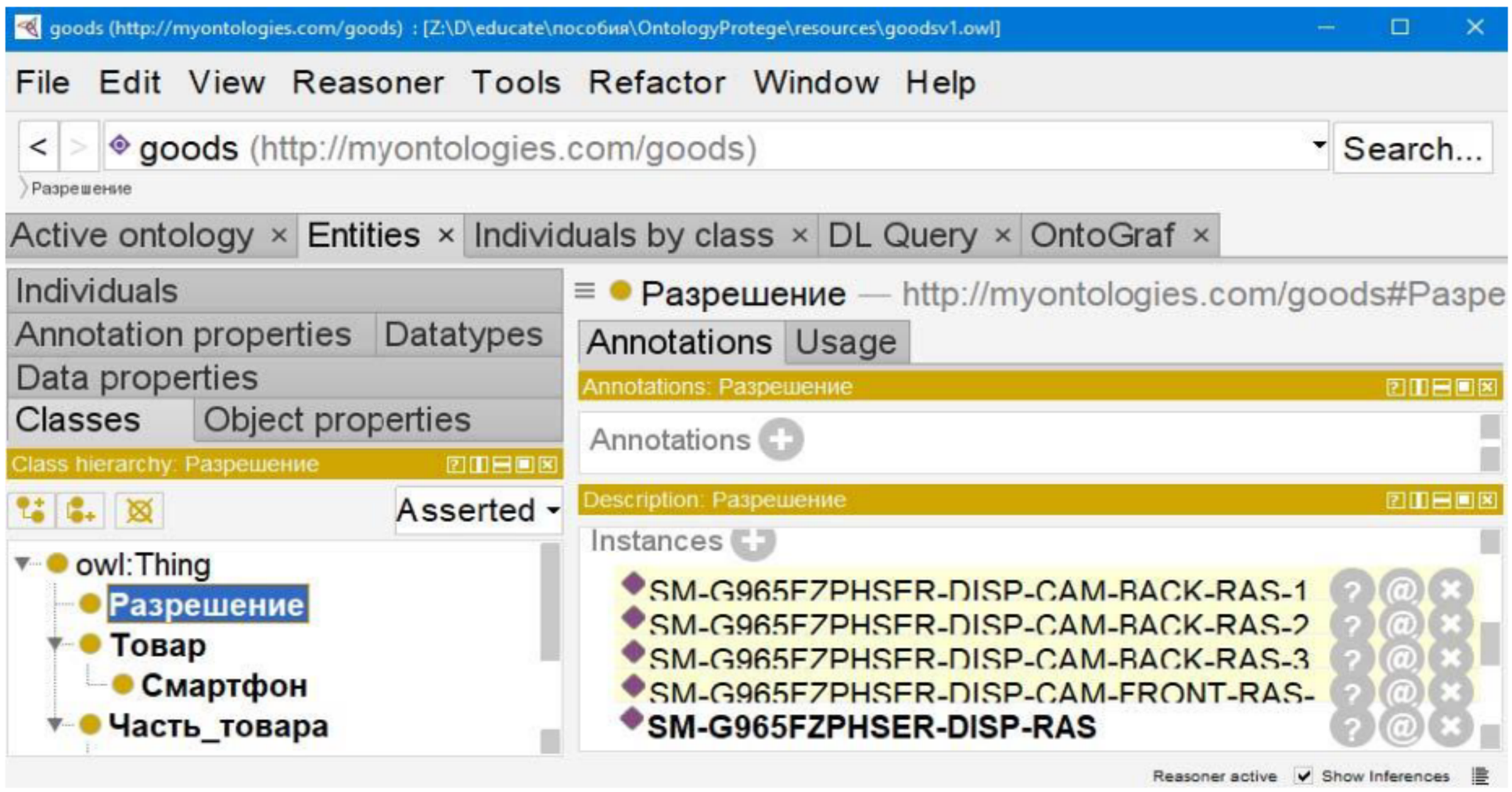


Рис. 18 Выведенные экземпляры у класса «Разрешение» в системе Protégé

Рис. 19 Демонстрация вывода в соответствии с описанием inverseOf

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000C3E9A98B9E0000000C3E9A98B9
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Задания

1 Дополнить созданную базу знаний минимум 10 экземплярами других смартфонов

2 По аналогии, дополнить базу знаний для категорий «ноутбук» и «холодильник»

Создание defined classes в системе PROTÉGÉ

В этой работе будут созданы классы, отношения «rdf:type» с которыми не будут указываться в явном виде, а будут строиться автоматически в процессе вывода. Такие классы в системе Protégé называются «Defined Classes» («обычные» классы называются «Primitive Classes»). Для примера будут реализованы следующие классы: «Новый смартфон», «Камерофон», «Игровой смартфон», «Смартфон для фильмов», «Смартфон для походов». Это позволит образовать в онтологии группы экземпляров в соответствии с заранее заданными критериями, которые будут доступны для дальнейшей обработки.

Первоначально сформируем критерии, по которым экземпляры будут относится к заявленным классам:

1 Новый смартфон. Выпущен после 18 года.

2 Камерофон. Минимум 2 основные камеры. Минимум 12 мегапикселей. Есть оптическая стабилизация. Съемка видео минимум Ultra HD 4K 60кадр./сек.

3 Игровой смартфон. Процессор Snapdragon или Exynos. Минимум 6 ядер. Частота работы процессора минимум 2.9ГГц. Объем оперативной памяти минимум 6гб. Графический ускоритель Mali или Adreno.

4 Смартфон для фильмов. Игровой смартфон. Соотношение сторон 21:9. Разрешение экрана от 4K.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

5 Смартфон для походов. Емкость аккумулятора от 6000 мА*ч. Степень защиты выше чем IP68. Защита экрана «Gorilla Glass». Защита корпуса

«укрепленный корпус». Материал корпуса включает металл.

Далее отладим запросы, в соответствии с которым будут формироваться экземпляры обозначенных классов. Такие запросы формируется с использованием Manchester Syntax и называются в системе Protégé «DL Query». Для отладки таких запросов выделена отдельная вкладка (рис. 20).

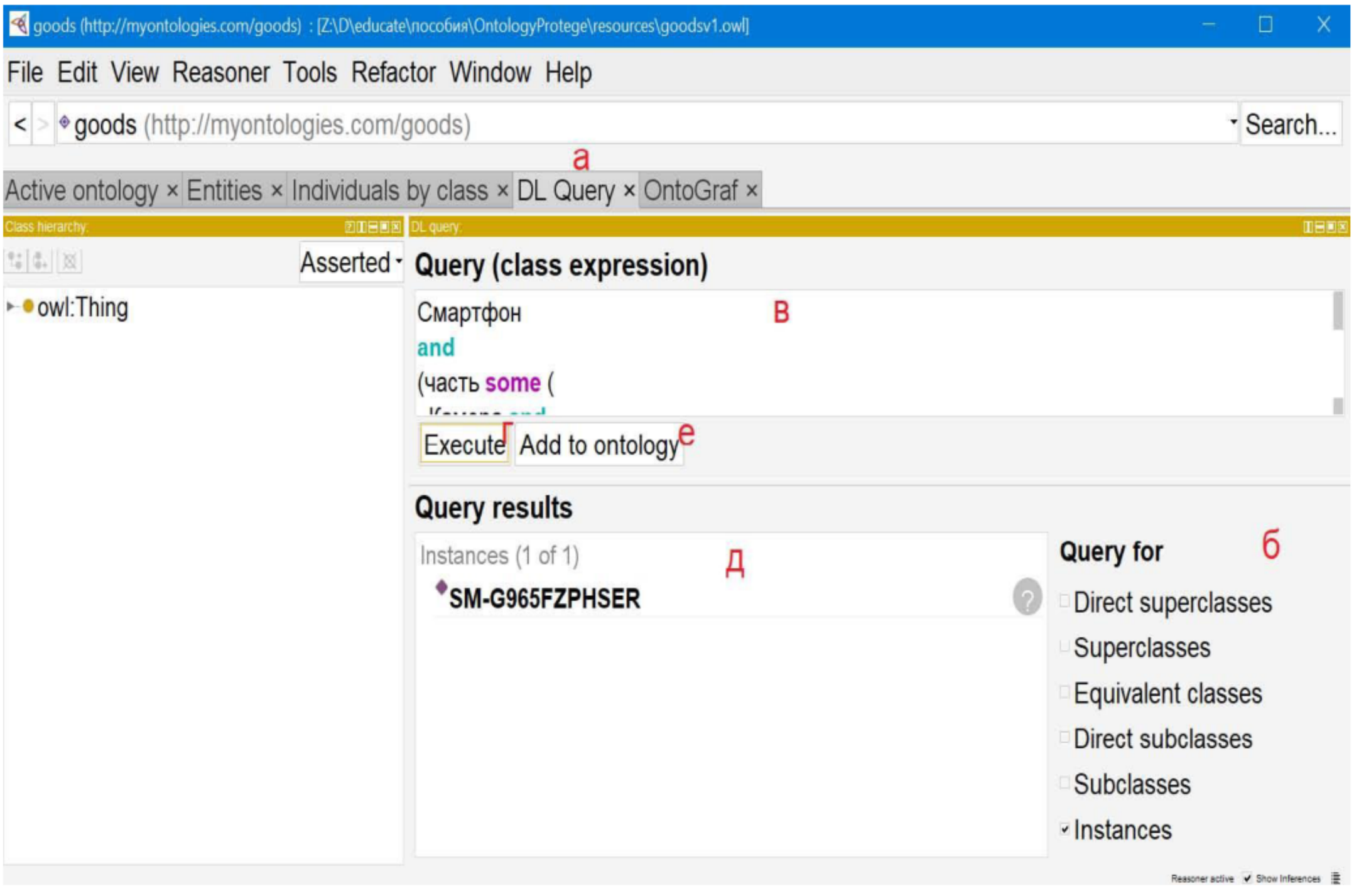


Рис. 20 Вкладка «DLQuery» в системе Protégé (а – панель вкладок с активной вкладкой «DLQuery»; б – панель выбора типа запрашиваемых сущностей; в – поле ввода запроса; г – кнопка выполнения запроса; д – результаты запроса; е – кнопка добавления запроса в онтологию)

Для освоения интерфейса выберем все экземпляры класса «Смартфон». Для этого необходимо в поле ввода запроса (рис. 20в) ввести запрос «Смартфон», выбрать ожидаемый тип результата – экземпляры (рис. 20б), запустить запрос (рис. 20г). После этого можно увидеть результаты (рис. 20д).

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Перед написанием запросов необходимо определиться с возможными для использования конструкциями (объяснения будут приводится применительно к типу выбираемых сущностей – экземпляр):

1. Локальное имя класса. Позволяет выбрать экземпляры данного класса. Как явно заданные, так и выведенные. Например: «Смартфон».

2. <локальное_имя_свойства> some <выражение>. Позволяет выбрать все экземпляры, для которых задано как минимум одно отношение <локальное_имя_свойства> с сущностью, подходящей под <выражение>. В качестве выражения могут выступать: вложенный «DL Query»; любое вариант ограничений, рассмотренный ранее при задании области значения data property. Например: «год_выпуска some xsd:integer»

3. <локальное_имя_свойства> only <выражение>. Все экземпляры, для которых отсутствуют заданные отношения или они все удовлетворяют выражению. Например: «год_выпуска only xsd:integer[>= 18].

4. <локальное_имя_свойства> exactly <количество>. Все экземпляры, для которых задано указанное количество указанных отношений. Например: «год_выпуска exactly 1»

5. <локальное_имя_свойства> exactly <количество> <выражение>. Все экземпляры, для которых задано указанное количество указанных отношений, удовлетворяющих выражению. При этом может быть сколько угодно таких же отношений, не удовлетворяющих выражению. Например: «год_выпуска exactly 1 xsd:integer[>= 18]»

6. <локальное_имя_свойства> min/max <количество> [<выражение>]. Аналогично exactly, но должно быть не менее/не более, чем указанное количество указанных отношений.

7. <локальное_имя_свойства> value <литерал>. Все экземпляры, которые связаны указанным отношением с указанным скалярным значением.

8. <выражение1> and/or <выражение2>. Пересечение/объединение множеств выбранных экземпляров в рамках выражения1 и выражения2. Например: «Дисплей or Процессор»

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 3C0000013F9ABBB52385E7BA500680000435
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

9. not <выражение>. Можно рассматривать как операция дополнения над множеством экземпляров, выбранных в рамках выражения. Например: «not Смартфон»

Определившись с множеством возможных конструкций, составим запрос для первого класса «Новый смартфон». У экземпляров такого класса не должно быть отношений «год выпуска», связывающих их с числом меньше 18 (для упрощения в первом варианте запроса рассмотрим год как двузначное число). Но, при этом должно быть хоть одно такое отношение (если таких отношений нет, то недостаточно информации для принятия решения). Также экземпляр должен быть типа «Смартфон». Соответственно, необходимый нам запрос должен описываться как пересечение 3 множеств: «Смартфон and (год_выпуска min 1) and (год_выпуска only xsd:integer[>= 18])».

Выполнив этот запрос на вкладке «DL Query», можно увидеть, что результатов не найдено. Вывод основывается на теории дескриптивных логик. Однако, для понимания вывода и дальнейших действий, рассмотрим причины пустого результата в упрощенном варианте.

В онтологии относительно года выпуска есть как минимум утверждения «:SM-G965FZPHSER rdf:type Смартфон; :год_выпуска 18». Для того, чтобы экземпляр был выбран в результатах запроса, необходимо, чтобы он относился к каждому из трех множеств. Учитывая приведенные утверждения, можно сделать вывод, что «:SM-G965FZPHSER» относится к классу «Смартфон» и «год_выпуска min 1». Но, нельзя сделать никакие выводы относительно «год_выпуска only xsd:integer[>= 18]». У нас информация о том, что существует такое x(«:SMG965FZPHSER :год_выпуска x»), которое удовлетворяет ограничениям (x >= 18), но из этого нельзя сделать вывод, что для любого x это ограничение всегда выполняется. При этом в результаты запроса «not (Смартфон and (год_выпуска min 1) and (год_выпуска only xsd:integer[>= 18]))» рассматриваемый экземпляр также не

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952265E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

войдет. Т.е. для «год_выпуска only xsd:integer[>= 18]» применительно к данного экземпляру не был сделан вывод ни о ложности, ни о истинности высказывания. В этом (предположение об открытости мира) состоит еще одна существенная разница с базами данных.

Одним из вариантов решения сложившейся проблемы является внесение в онтологию новых сведений. Так, если бы для рассматриваемого экземпляра выполнялось ограничение «:год_выпуска exactly 1» (для рассматриваемого экземпляра существует ровно одно отношение «год выпуска»), то с учетом существования подходящего под критерии отношения уже можно было бы сделать вывод о том, что существуют только такие отношения. Эти сведения могут быть добавлены в разделе «Types» экземпляра с использованием вкладки «Class expression editor» (рис. 21).

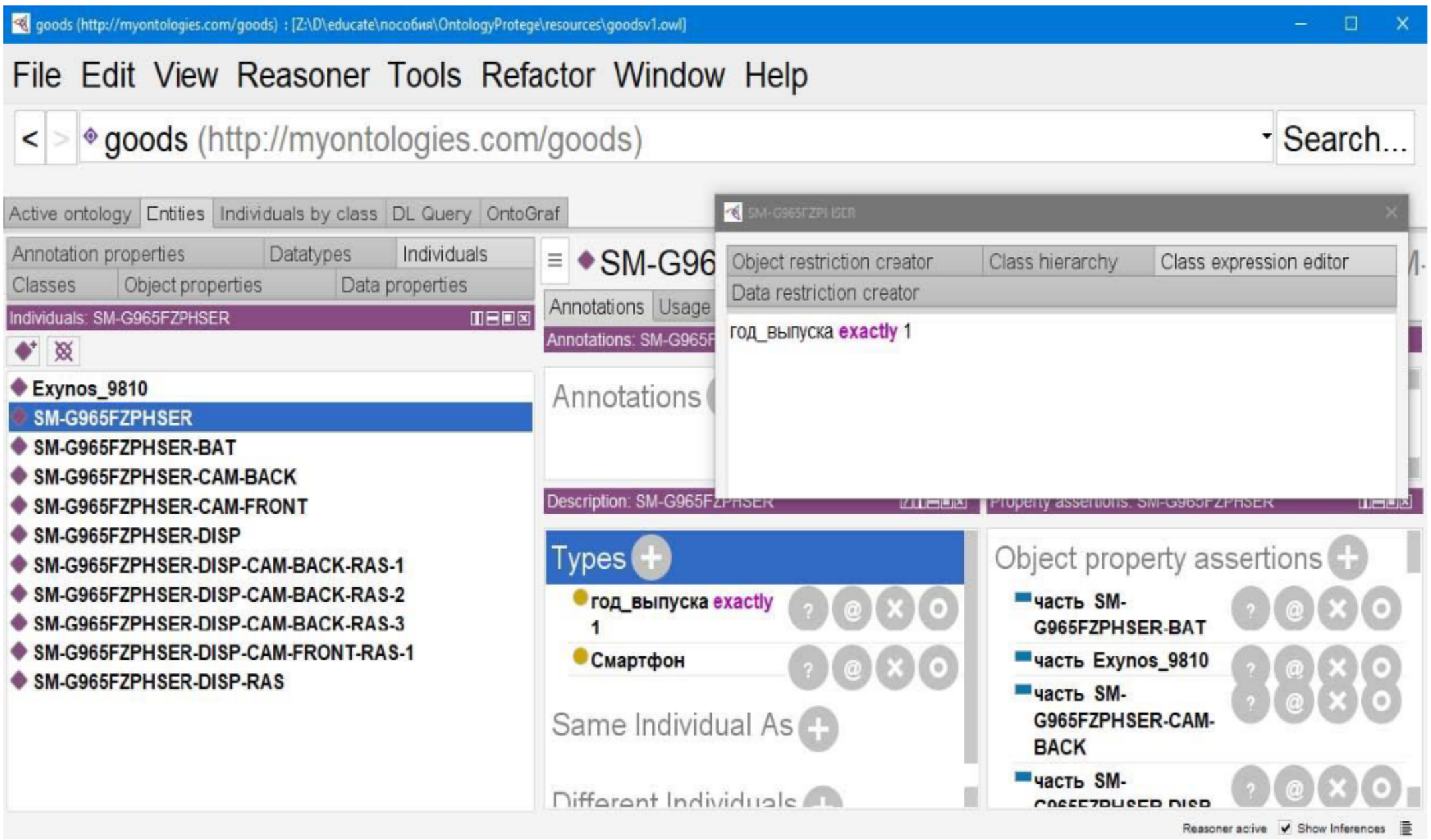


Рис. 21 Использование вкладки «Class expression editor» для добавления информации о экземпляре

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебукова Наталья Владимировна
Действителен: с 19.08.2022 по 19.08.2023

После отладки запроса необходимо добавить класс. Это можно сделать с использованием кнопки «Add to ontlogy» на вкладке «DL Query» (рис. 20е)

или создав класс и заполнив поле «Equivalent To» в его описании (рис. 22).

Рассмотрим, как данный класс описывается в тексте онтологии.

:Новый_смартфон rdf:type owl:Class ;

owl:equivalentClass [owl:intersectionOf (:Смартфон

[rdf:type owl:Restriction ;

owl:onProperty :год_выпуска ;

owl:allValuesFrom [rdf:type rdfs:Datatype ; owl:onDatatype

xsd:integer ; owl:withRestrictions ([xsd:minInclusive 18])]] [

rdf:type owl:Restriction ; owl:onProperty :год_выпуска ;

owl:minCardinality "1"^^xsd:nonNegativeInteger]); rdf:type

owl:Class]

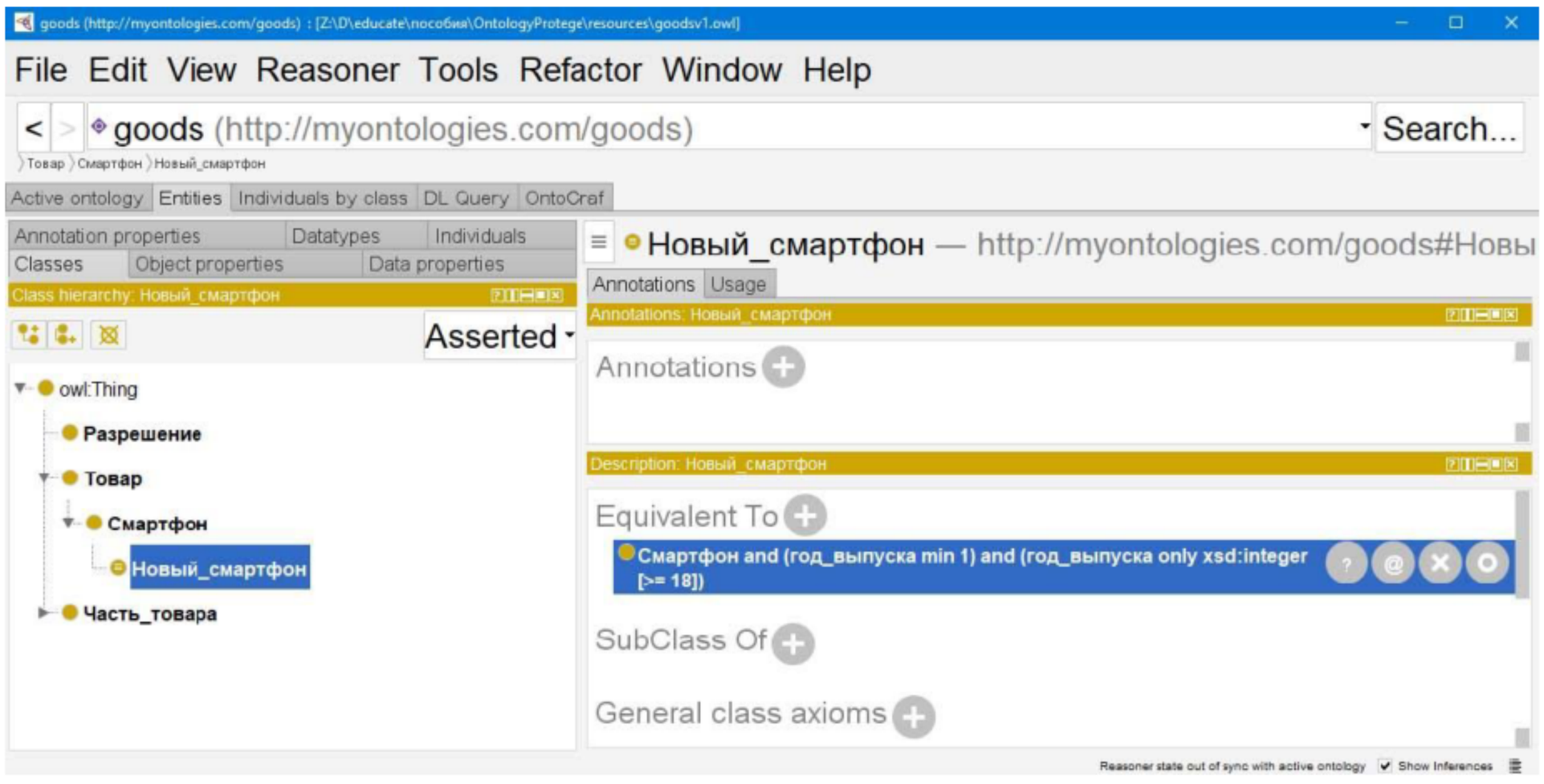


Рис. 22 Пример заполнения поля «Equivalent To» в описании класса

Как видно, используется отношение «owl:equivalentTo», которое позволяет сделать вывод, что если экземпляры относятся к «объекту» этого отношения, то относятся и к «субъекту». А в качестве «объекта» выступает пустой узел, являющийся пересечением 3 других классов, как и планировалось изначально.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: 13.02.2022 до 20.08.2023

Дополним написанный запрос для работы с годом, заданным в полном формате:

Смартфон and (год_выпуска min 1) and (год_выпуска only (xsd:integer[>= 18,

```
<= 25] or xsd:integer[>= 2018]))
```

Также составим запрос для класса «Камерофон»:

```
Смартфон
and
(часть some ( Камера and (количество_основных_камер some xsd:integer[ >= 2
  ])
  and (количество_мегапикселей_основной_камеры some xsd:integer[ >= 12 ])
  and
    (разрешение some (Разрешение and
      ((разрешение_горизонталь some xsd:integer[ >=
      3840 ]) or (разрешение_вертикаль some
      xsd:integer[ >= 3840 ])) and
      (разрешение_кадр/сек some xsd:integer[ >= 60 ])
  ))
))
```

Задания

1 Составить запросы для оставшихся групп: «Смартфон для фильмов», «Игровой смартфон», «Смартфон для походов». Создать defined class для каждой.

2 Сформировать минимум 5 групп (на основании имен подборок на ресурсе источнике) для категорий «Ноутбуки» и «Холодильники». Для каждой из групп сформировать критерии выбора экземпляров и создать defined class.

Написание запросов на языке SPARQL

В предыдущей работе было произведено отнесение экземпляров к группам по определенным критериям. Для этого использовались defined classes и язык DL Query. Данные инструменты сильно расширяют

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: [illegible]
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

возможности рассуждений в рамках онтологий. Однако, DL Query применяется именно в рамках рассуждений и имеет ряд недостатков, если использовать его только для получения доступа к явно заданным или уже выведенным сущностям онтологии: медленное выполнение (т.к. используется логический вывод); необходимость учитывать поведение при выводе (например, добавления в онтологию «exactly» в прошлой работы); невозможность описания некоторых условий (например, выбор отношений, а не экземпляров; выбор ряда характеристик смартфона). Поэтому для таких задач используется другой язык – SPARQL [8]. Данный язык позволяет выбирать и добавлять сущности в онтологию, формировать новый граф RDF. В данном работе будут рассмотрены только возможности выбора. Язык синтаксически схож с SQL (и не только синтаксически. В частности, он также основан на предположении о закрытости мира, в отличие от DL Query), применяемым для аналогичных задач в базах данных. Поэтому (и потому что SQL должен быть уже знаком обучаемым) с ним будут проводиться аналогии.

Для работы с SPARQL в системе Protégé необходимо активировать вкладку «Window -> Tabs -> SPARQL Query» (рис. 23). Необходимо понимать, что SPARQL и механизм рассуждений в Protégé работают независимо и не влияют на результаты работы друг друга. Это означает, что нет возможности выбрать экземпляры класса «Новый смартфон», которые являются выведенными. При этом это возможно в рамках, например, библиотеки Apache Jena для языка Java.

Первоначально выберем первые 2 смартфона и характеристики их камер (количество камера; мегапиксели; апертура) и рассмотрим на примере этого запроса основные синтаксические конструкции.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

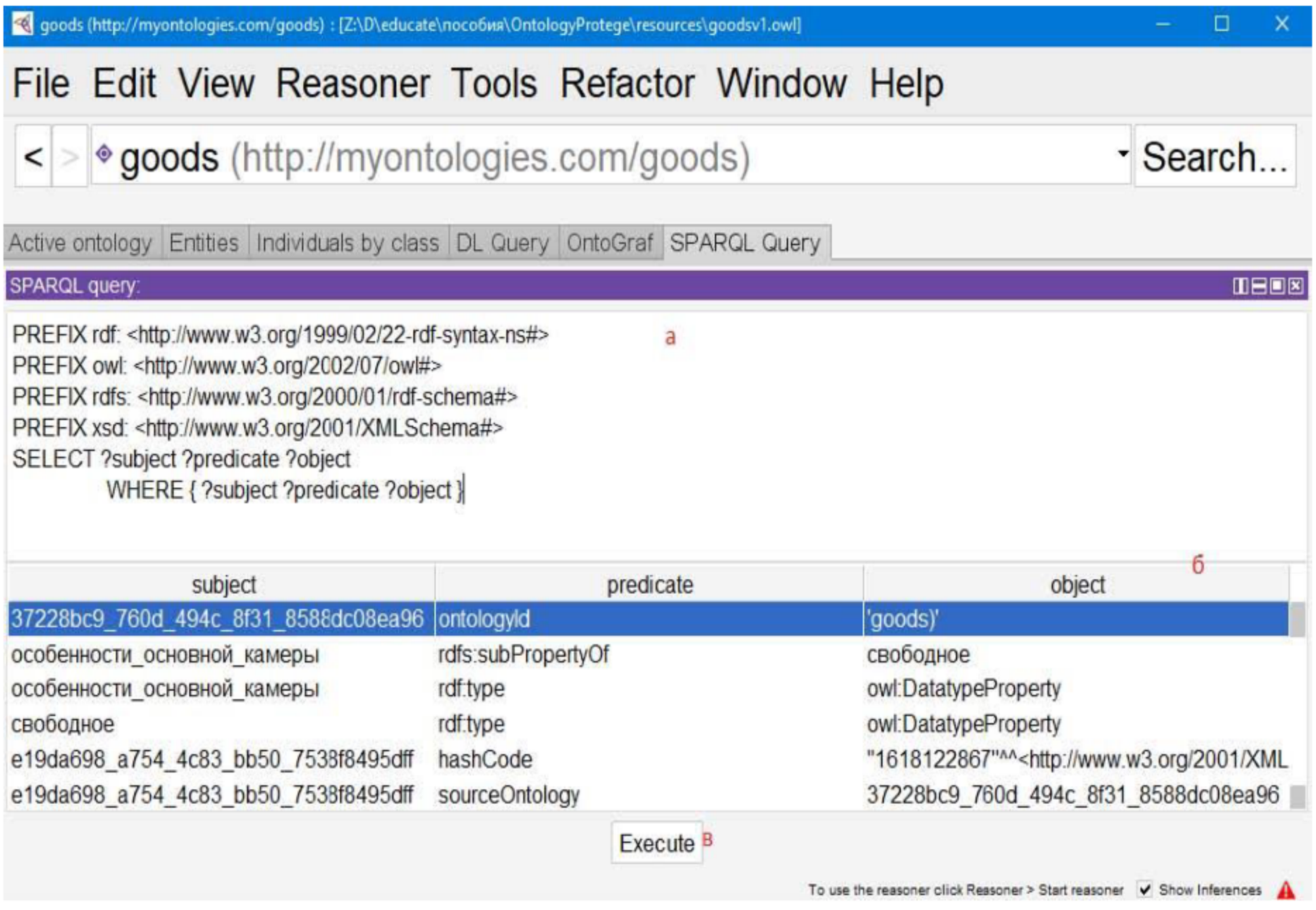


Рис. 23 Вкладка «SPARQL Query» в системе Protégé (а – поле ввода запроса; б – результаты запроса; в – кнопка выполнения запроса)

#Сначала приводится блок стандартных префиксов для того, чтобы не записывать громоздкие идентификаторы в рамках основной части запроса

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX owl: <http://www.w3.org/2002/07/owl#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

#Также необходимо добавить префикс для созданной онтологии (ее идентификатор берется с вкладки «Active Ontology»)

PREFIX : <http://myontologies.com/goods#>

#После префиксов идет конструкция SELECT{переменные}, которая описывает что будет выбрано в качестве результата запроса

#Все переменные начинаются с «?» и перечисляются через пробел

SELECT ?смартфон ?камера_число ?камера_мп ?камера_ап

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0[illegible]
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

#Далее идет конструкция WHERE { {тройки} }. Тройки записываются аналогично синтаксису Turtle, что позволяет также использовать разделители «.», «;», «,».

WHERE {

#Вместо отдельных сущностей могут ставится переменные, начинающиеся с «?». На них могут накладываться условия, они могут выбираться в рамках списка полей в «SELECT».

?смартфон :часть ?камера .

?камера rdf:type :Камера ;

:количество_основных_камер ?камера_число ;

:количество_мегапикселей_основной_камеры ?камера_мп ;

:апертура_основной_камеры ?камера_ап }

#Далее идет конструкция ORDER BY, позволяющая задать сортировку результатов

ORDER BY DESC(?камера_ап)

#Далее идет конструкция LIMIT, OFFSET позволяющая задать количество пропускаемых записей и количество выбираемых

LIMIT 2 OFFSET 0

Как видно, структура запроса схожа с языком SQL (в том числе доступны конструкции GROUP BY, HAVING), однако содержимое блоков различается. Основная часть – WHERE, разберем подробнее ее вложенные конструкции, которые будут использоваться в рамках работы:

1. Тройка «субъект предикат объект». На месте каждого из элементов тройки может стоять конкретный IRI или литерал или переменная. Позволяет выбрать все тройки онтологии, подходящие под заданную (заданная тройка используется как маска). При выборе: на месте переменных в заданной тройке может стоять любая сущность в выбираемой тройке; литералы должны совпадать по значению, типу, языку; IRI на соответствующих местах должны совпадать.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 3C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

2. Разделители «.» между тройками можно интерпретировать как декартового

произведение, но с учетом равенства соответствующих переменных. Например, «?смартфон :код_производителя ?код» позволяет выбрать все тройки, в которых присутствует отношение «код производителя». Если в онтологии 2 смартфона и только у одного из них есть это отношение, то будет выбрана 1 тройка.

«?смартфон :часть ?часть» позволяет выбрать все тройки, в которых присутствует отношение «часть». Если в онтологии 2 смартфона и у каждого из них по 5 частей, то будет выбрано 10 троек.

Если упрощенно рассмотреть порядок выбора в рамках «?смартфон :код_производителя ?код . ?смартфон :часть ?часть», то будет сначала выбрана 1 запись по первой тройке, 10 записей по второй. Далее будет сформировано 10 записей как все варианты совмещений данных записей. После чего будут удалены 5 записей, где в рамках одноименных переменных «смартфон» не совпадают значения.

3. В рамках троек также могут использоваться пустые узлы, описываемые аналогично синтаксису Turtle с использованием «[]».

4. OPTIONAL { {вложенная часть where} }. Позволяет задать вложенный запрос, который будет использован только при наличии результатов. Например, «?смартфон :код_производителя ?код . ?смартфон :часть ?часть» выберет 10 записей, если в онтологии будет 2 смартфона и 5 частей у каждого. Но, если будет 2 смартфона и 0 частей у каждого, то никаких результатов выбрано не будет. При этом «?смартфон :код_производителя ?код OPTIONAL { ?смартфон :часть ?часть }» выберет также 10 записей в первом случае, но 2 записи во втором (часть в рамках OPTIONAL учитываться не будет).

5. FILTER({условие}). Позволяет оставить в рамках выбранных записей только те, которые подходят под условие. Например, «?смартфон :год_выпуска ?год FILTER(?год > 18)» позволяет выбрать только те результаты, где год выпуска позднее 18 (стоит учитывать, что если у смартфона 2 таких отношения и у одного значение больше 18, а у другого меньше, то

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000013E9AB8B953305E7BA500060000013E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023