

Методы распределенного репозитория системы моментальных платежей на основе онтологии

Р.А. Гайдуков
Факультет информатики робототехники
Уфимский государственный авиационный
технический университет
Уфа, Россия

В.А. Котельников
Факультет информатики робототехники
Уфимский государственный авиационный
технический университет
Уфа, Россия
e-mail: vit_kot@mail.ru

Н.И. Юсупова
Факультет информатики робототехники
Уфимский государственный авиационный технический университет
Уфа, Россия
e-mail: yussupova@ugatu.ac.ru

Аннотация¹

Для полноценного функционирования систем моментальных платежей следует иметь отлично построенную архитектуру системы, которая должна отвечать всем требованиям. В работе разработана одна из необходимых ее частей – репозиторий на основе онтологии, проведена оценка ее эффективности с точки зрения когнитивной эргономики.

1. Введение

Система моментального платежа это сети электронных платежных терминалов, принимающие наличные деньги и позволяющие оплачивать услуги операторов мобильной связи, доступа в интернет спутникового телевидения, услуги ЖКХ и пр. Система моментальных платежей позволяет сократить время подключения, резко упростить взаиморасчеты, упростить подготовку менеджеров за счет единого интерфейса и заметно расширить спектр услуг, например за счет оплаты роуминговых абонентов.

Провайдеры услуг предоставляют сетевые сервисы финансовых операций. Существуют платежные системы, которые предоставляют свои сетевые сервисы для работы с провайдерами услуг и их число постоянно растет. В условиях децентрализации для обеспечения расширения перечня услуг без изменения программного обеспечения необходимо объединение в единое информационное

пространство– репозитарий, и возможность всех участников системы помещать в него данные и получать информацию. При решении поставленной задачи встает вопрос представления предметной области на основе единого описания услуг в виде онтологии.

2. Постановка задачи

Необходимо разработать концепцию функционирования репозитория на основе онтологии, обеспечивающую возможность оказания услуги как напрямую поставщику услуги или через посредника, предоставляющего доступ к услуге.

Репозиторий должен выполнять следующие функции:

- содержание релевантных классов объектов и их связи
- конфиденциальность данных
- редактирование данных
- полное функционирование с системой моментальных платежей

Для организации приема платежей в репозитории должны храниться разрабатываемые поставщиками протоколы информационного обмена для взаимодействия с ними, которые можно представить, как сетевой сервис. Создание и описание функциональности сервиса и опубликование его в сети - задача владельца. Использовать эту информацию могут быть все, от обычного пользователя, до компьютера, выполняющего определенные задачи в автоматизированном процессе. Назовем данное представление - описание Предложения Сервиса (ОПС).

Труды третьей международной конференции "Интеллектуальные технологии обработки информации и управления", 10 - 12 ноября, Уфа, Россия, 2015

ОПС попадает к интересующемуся пользователю (ИП) двумя вариантами Первый вариант - ОПС хранится в централизованном репозитории, и ИП сам ищет там подходящий ему сервис. В данном случае, описание предложения сервиса должно, в основном, содержать формализованную информацию о том, как вызывается описываемый сервис. Описание функциональности сервиса в подобном описании предложения сервиса может быть представлено в достаточно свободной форме, например, как текстовое описание.

Второй вариант – это частичная или полная автоматизация процесса поиска необходимого сервиса. В этом случае репозиторий для хранения

описания предложения сервиса может быть, как распределенным, так и централизованным, в зависимости от инфраструктуры сети. Поиск же выполняется не конечным пользователем, а клиентским приложением.

Процесс сравнения различных описаний предложения сервиса с Описанием Запроса Пользователя (ОЗП) может происходить как на стороне пользователя, так и на стороне репозитория. Первый случай подходит для распределенного варианта хранения ОПС, а второй для централизованного варианта хранения ОПС. На рисунке 1 представлена общая схема использования сетевого сервиса.

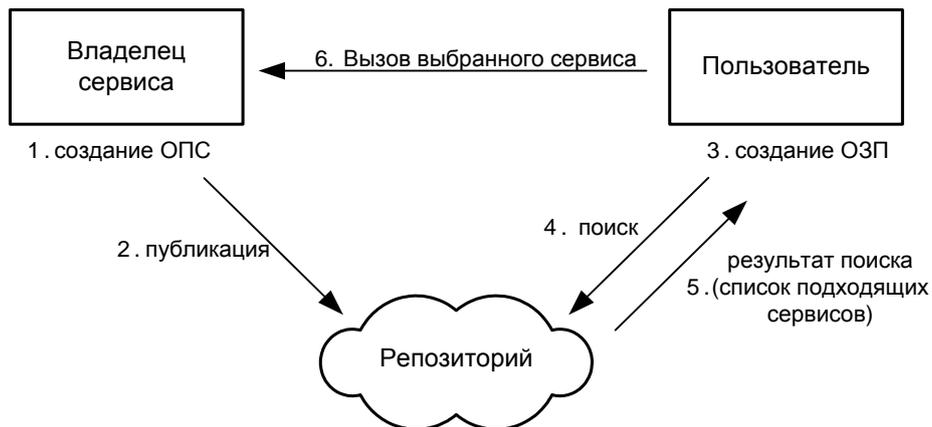


Рис. 1. Общая схема использования сетевого сервиса

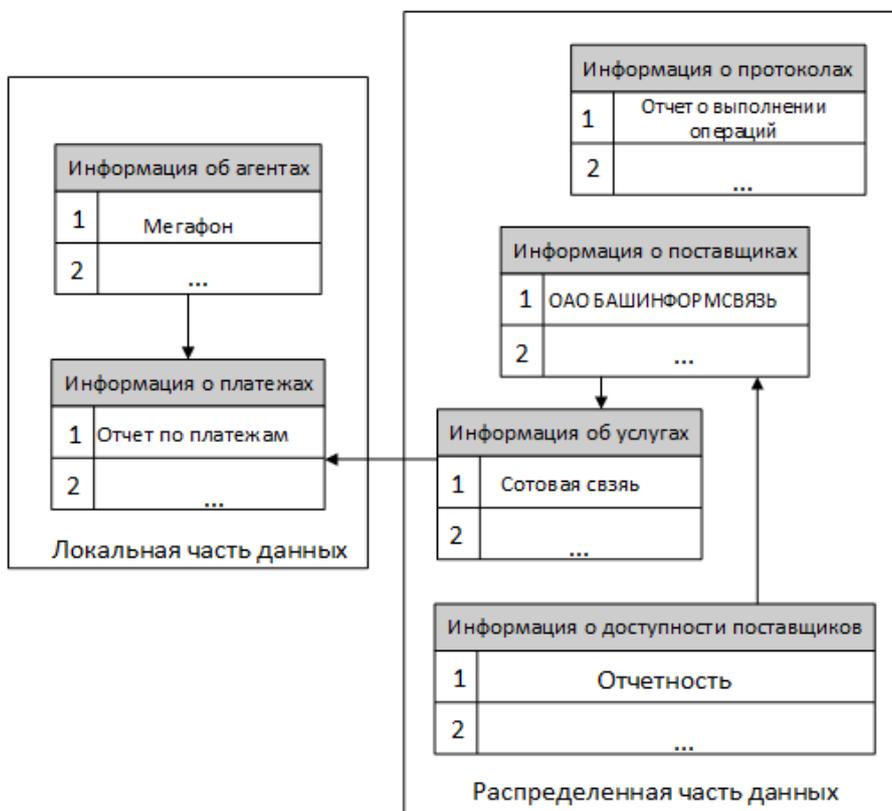


Рис. 2. Структура репозитория платежной системы

Процесс вызова сервиса могут осуществляться как пользователем, так и быть автоматизированными. Для автоматизации необходимо, чтобы запросы имели машино-понятную формализованную форму, достаточную для выполнения машинно-зависимых операций. Выразительная способность формализованных описаний сервиса определяет степень возможной автоматизации процесса использования сервиса. Другими словами, выразительная способность описания определяет, насколько точно и полно можно отобразить реальную функциональность сервиса в ОЗП или ОПС.

3. Разработка структуры репозитория

Рассмотрим структуру репозитория изображенную на рисунке 2. Она состоит из двух логических частей: локальной и распределенной.

Локальная часть содержит информацию о платежах, проведенных через местную часть платежной системы и информацию о локальных агентах принявших платежи в пользу операторов услуг.

Распределенная часть синхронизируется между репозиториями распределенной платежной системы и содержит информацию об услугах, поставщиках, протоколах передачи информации и доступна всем участникам распределенной системы и используется

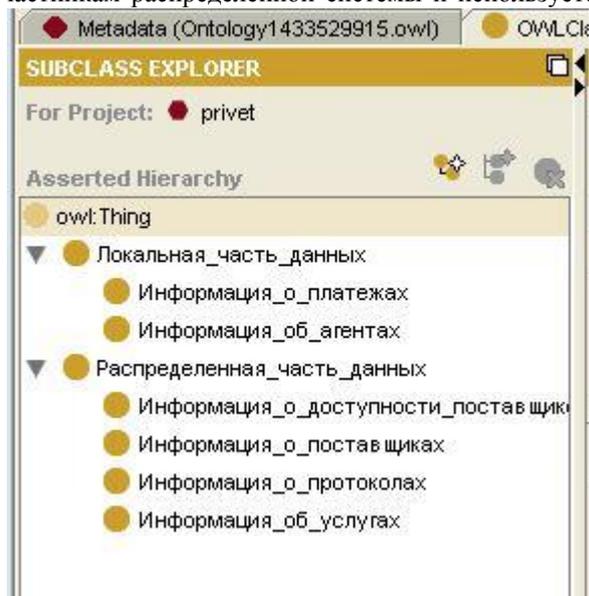


Рис. 3. Окно создание классов

для определения оптимального между посредниками или поставщиком услуги способа проведения платежа.

Использование единого семантического описания сервисов позволяет интегрировать в рамках одной платежной системы оказание клиентам оплаты услуг, как имеющихся провайдеров, так и будущих, при этом подключение нового провайдера является простой тривиальной операцией.

4. Разработка онтологии в Protégé

Во вкладке OWLClasses создаем классы. Каждый из этих классов обязательно должен быть подклассом системного класса THING. Для того, чтобы это сделать, можно выделить класс THING и щелкнуть на значок добавления подкласса на панели или же нажать правой кнопкой мыши на классе и выбрать пункт «create subclass». Таким образом создаем классы «Локальная часть данных» и «Распределенная часть данных». В свою очередь для класса «Локальная часть данных» создаем 2 подкласса «Информация о платежах» и «Информация об агентах», а для класса «Распределенная часть данных» « информация о доступности поставщиков», «Информация о поставщиках», «Информация о протоколах» и «Информация об услугах» (рис. 3)

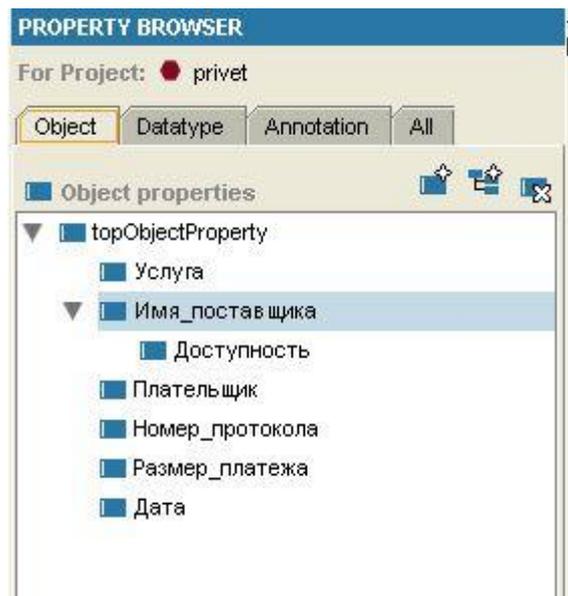


Рис. 4. Окно создания свойств классов

После создания нужных нам классов, приступаем к созданию свойств. Свойства объекта определяют отношения (предикаты) между двумя объектами (также называемые individuals) в OWL онтологии. Конструктивно, свойства объекта являются основой, соединяющие узлы в OWL ontology graph. Ролью свойств является определение допустимых типов данных и значений данных, связанных с объектами. Так же, как и при создании классов, создание свойств начинается с создания свойств-наследников для системного свойства «topObjectProperty». Создаем

свойства «Услуга», «Имя поставщика», «Плательщик», «Номер протокола», «Дата», так же у свойства «Имя поставщика» будет свойство-наследник «доступность» (рис. 4). После создания, для каждого свойства указываем «Domain» и «Range», для определения к какому классу привязано свойство (рис. 5).

Для создания объектов, которые будут определять класс, переходим во вкладку «Individuals». Так как класс «Информация о поставщиках» представляет из себя набор сущностей, его объекты будут состоять

только из реально существующих поставщиков услуг на данный момент времени (рис. 6).

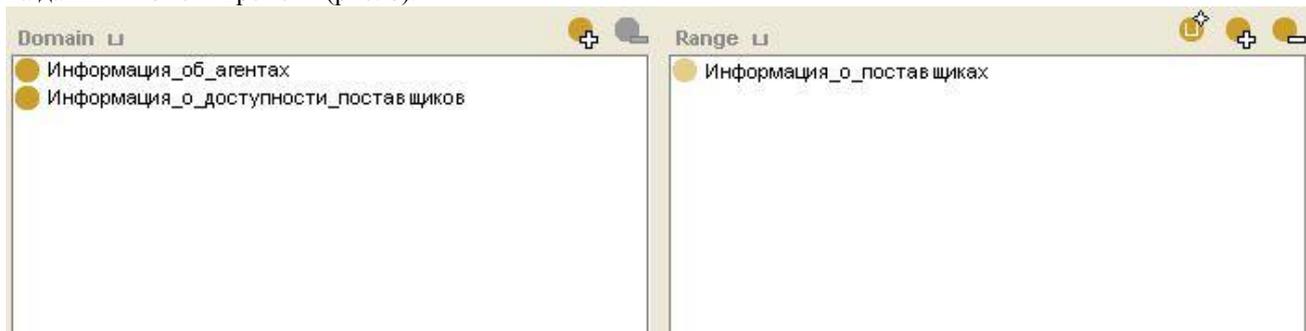


Рис. 5. Окно связывания свойств с классами (Доступность)

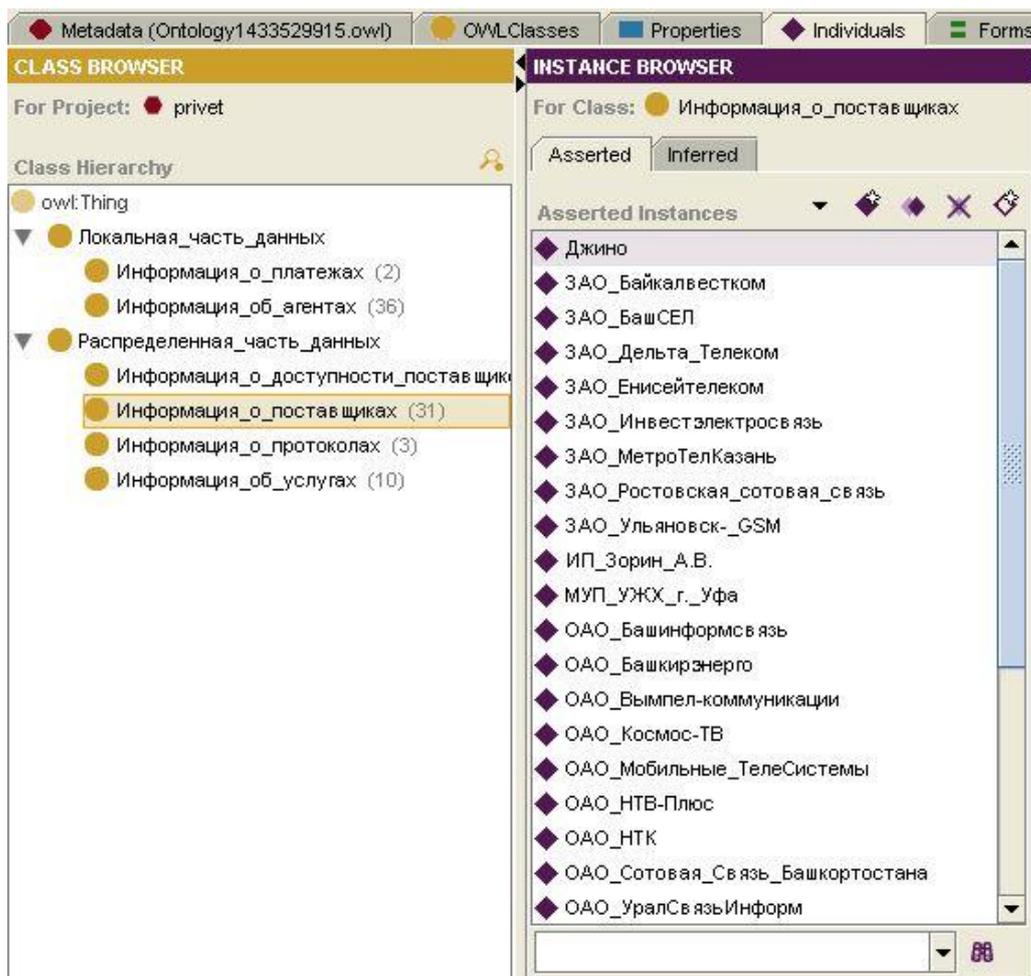


Рис. 6. Окно создания сущностей

Класс «Информация об агентах» имеет несколько свойств – доступность и имя поставщика. За каждое из этих свойств отвечает некоторый класс (рис. 7).

Для наглядной визуализации иерархии классов был использован модуль Jambalaya (рис. 8)

Таким образом, с помощью программного обеспечения Protégé, была создана онтология, для системы моментального платежа.

5. Анализ эффективности разработанной онтологии

Актуальная проблема современного онтологического инжиниринга - это оценки качества онтологий. Разными группами ученых разработано множество различных подходов в области оценки онтологий. На данный момент известно большое количество методов и способов оценки онтологии, и в месте с этим растет и сложность выбора подходящей методики.

Существующие методы оценки онтологий ставят перед собой одну из следующих целей:

- Полнота и точность словаря предметной области
- Адекватность структуры с точки зрения таксономии, отношений и т.п.
- Воспринимаемость (с когнитивной точки зрения).
- Производительность при использовании в приложениях

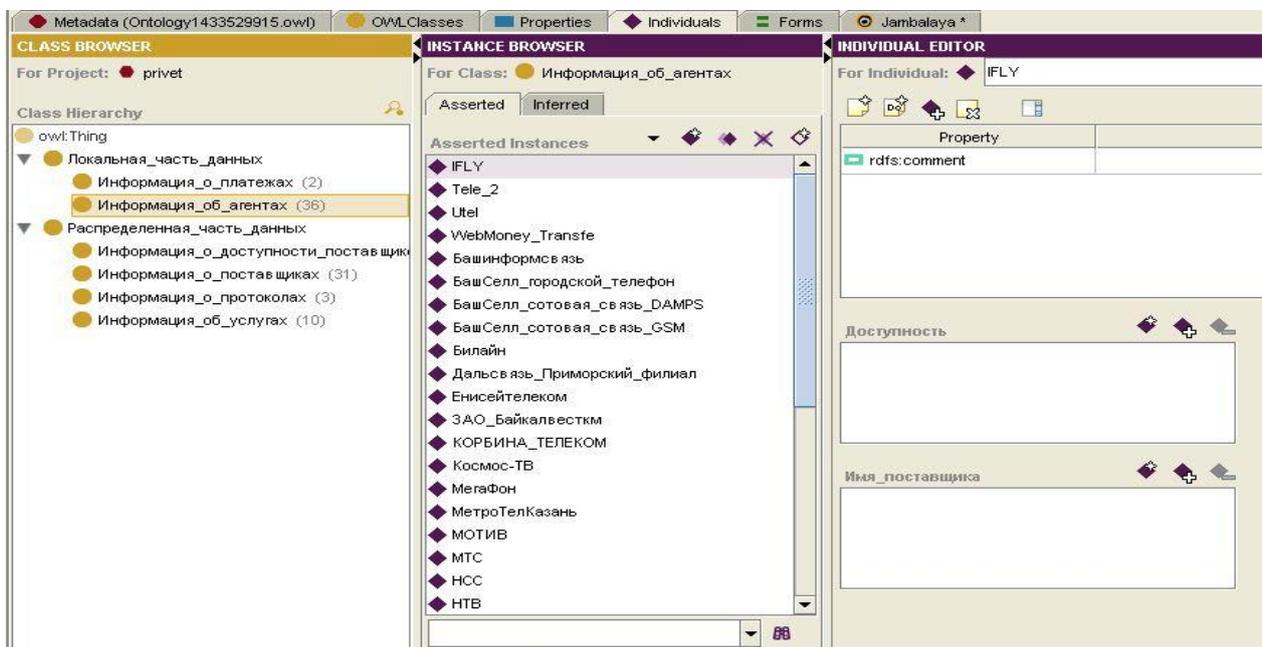


Рис. 7. Сущности класса "Информация об агентах"

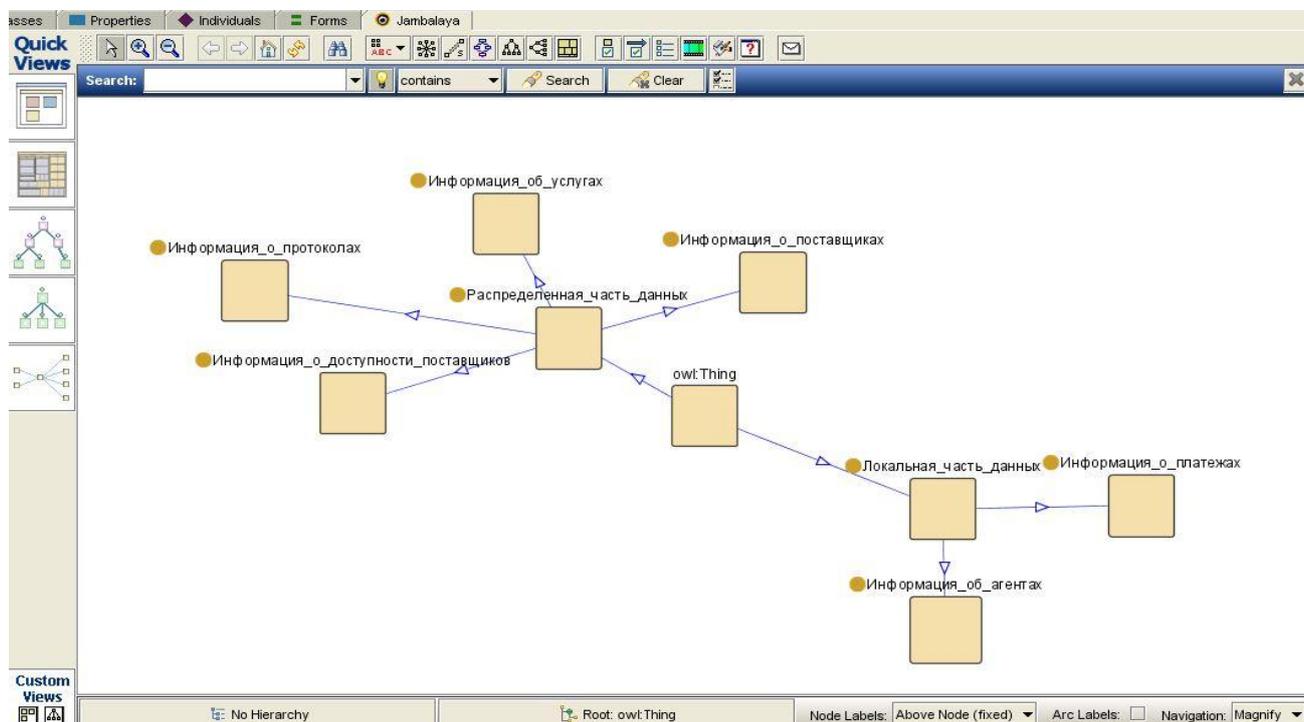


Рис. 8. Визуализация классов

Оценим разработанную онтологию с этих точек зрения. На рисунке 9 приведен граф онтологии. Ниже, в таблице 1, приводится достаточно много метрик, используемых для анализа качества

онтологии, часть из которых рассчитывается на основе топологии графа онтологии. Приведем здесь те из них, которые относятся к метрикам когнитивной эргономичности.

Создание распределенного репозитория системы моментальных платежей на основе онтологии

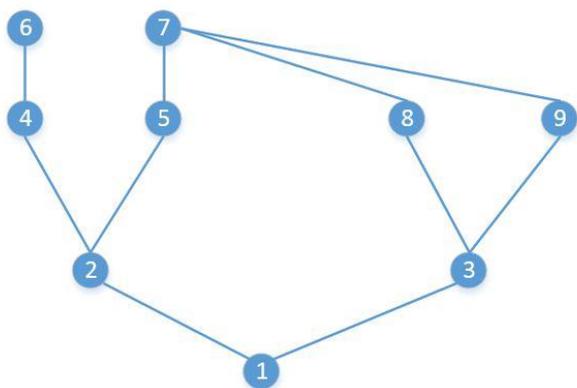


Рис. 9. Граф разработанной онтологии

Таблица 1

Название	Описание\Алгоритм вычисления	Комментарии и рекомендации	Расчетный показатель
Абсолютная глубина	Сумма длин всех путей графа	Чем больше глубина, тем труднее граф поддается восприятию.	$1+2+3+4+5+6+7+8+9 = 44$
Средняя глубина	Абсолютная глубина деленная на количество путей в графе		$\frac{44}{10} = 4,4$
Максимальная глубина	Максимальная длина пути		$1+2+5+7+9+3=27$
Абсолютная ширина	Сумма количества вершин для каждого уровня иерархии по всем уровням	Чем она меньше, тем лучше с точки зрения когнитивной эргономики.	9
Средняя ширина	Абсолютная ширина, деленная на количество уровней иерархии		2,25
Максимальная ширина	Равняется количеству вершин, на самом большом		4

	уровне		
Минимальная ширина	Абсолютная ширина, деленная на количество уровней иерархии		1
Запутанность онтологии	Количество вершин графа онтологии, деленное на количество вершин, у которых есть суперкласс	Чем меньше итоговое значение, тем лучше онтология с точки зрения когнитивной эргономики.	0
Отношение количества классов к количеству свойств	Чем больше, тем легче воспринимать онтологию.		1
Вершины с несколькими родителями	Количество вершин, имеющих более одного родителя		$\frac{9}{84} = \frac{1}{9,3} = 0,1$
Среднее количество родителей вершин у вершины графа	$m = \frac{1}{n_g} \sum_v N_{S_{v \in G}}$ $N_{S_{v \in G}}$ - количество всех родителей у вершины v	Более предпочтительно небольшие значения этих метрик	$\frac{1}{9} \sum_v^G 2 = 0,2$

Так как оценка онтологии обычно проходит в сравнении результатов между альтернативными вариантами на ту же тему, нельзя однозначно сказать хорошие это результаты или нет. Но так как все результаты вычисления так или иначе подходят под рекомендованные, можно сделать вывод об эффективности онтологии, с точки зрения когнитивной эргономики. Методы оценки когнитивной эргономичности также можно применять для оценки онтологий одной и той же предметной области, сделанных разными людьми/командами. Посчитанные метрики помогут понять, какая из них лучше с точки зрения когнитивной эргономики и сделать выбор в пользу

одной из них в случае, если оценки других важных критериев отличаются принципиально.

6. Заключение

Построение децентрализованного репозитория, на основе онтологии, представляет большой интерес для существующих систем моментальных платежей, так как в нем хранятся все сущности и связи между внутренними компонентами системы, что снижает риск выхода системы из строя из-за утери данных. Разработанная онтология, в среде разработки Protege, включает в себя 2 класса, 6 подклассов и 83 атрибута. Проведена оценка эффективности онтологии с помощью метрик, используемых для анализа качества онтологии, которая показала, что созданная онтология является эффективной с точки зрения когнитивной эргономики.

Список используемых источников

1. Котельников В.А. Онтологическое описание сетевых сервисов для распределенной системы финансовых операций // Труды Казанской школы по компьютерной и когнитивной лингвистике TEL-2006. Казань: Отечество, 2007. с. 127-138
2. Котельников В.А. О технологиях создания сетевых сервисов для системы финансовых операций на основе онтологического подхода (статья на англ. языке) // Материалы круглого стола «Информационные технологии и математические методы исследования в экономике» Башкирско-Саксонского форума. – Уфа, Изд. УГАТУ, 2007.
3. Котельников В.А., Попов Д.В. Разработка сетевых сервисов на основе онтологий для Интернет-коммерции: принципы и технологии. // "Обозрение прикладной и промышленной математики", 2008, т.15, в.1. стр. 145-148
4. Вайнерман И. А. Факторы и проблемы предоставления и использования сетевых сервисов/И. А. Вайнерман // Вестник УГАТУ / Уфим. гос. авиац. техн. ун-т, 2006.-С.Т. 8, N 1. - С. 43 - 47.
5. Гаврилова Т. Об одном подходе к онтологическому инжинирингу // Ж. «Новости искусственного интеллекта», №3, 2005. – с. 25-31.
6. Субъективные метрики оценки онтологий. Гаврилова Т.А., Горовой В.А., Болотникова Е.С., Горелов, В.В. Знания-Онтологии-Теории (ЗОНТ-09), 2009
7. Гаврилова, Т.А. Оценка когнитивной эргономичности онтологии на основе анализа графа / Т.А. Гаврилова, В.А. Горовой, Е.С.Болотникова // Искусственный интеллект и принятие решений. — 2009. — по. 03.