

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Кибернетики _____
Направление подготовки (специальность) 09.03.03. Прикладная информатика _____
Кафедра Программной инженерии _____

БАКАЛАВРСКАЯ РАБОТА

Тема работы	
Разработка модуля импорта метаданных в базу знаний научного эксперимента	
УДК 371.385.3:004.623-022.57	

Студент

Группа	ФИО	Подпись	Дата
8К31	Кайда Анастасия Юрьевна		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ведущий программист	Губин М.Ю.			

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. МЕН	Тухватулина Л.Р.	к.ф.н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Пустовойтова М.И.	к.х.н.		

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ПИ	Иванов М.А.	к.т.н		

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Кибернетики _____
Направление подготовки (специальность) 09.03.03. Прикладная информатика _____
Кафедра Программной инженерии _____

УТВЕРЖДАЮ:

Зав. кафедрой

(Подпись) (Дата) Иванов М.А.
(Ф.И.О.)

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

Студенту:

Группа	ФИО
8К31	Кайда Анастасии Юрьевне

Тема работы:

Разработка модуля импорта метаданных в базу знаний научного эксперимента

Утверждена приказом директора (дата, номер)

03.02.2017 №665/С

Срок сдачи студентом выполненной работы:

16 июня 2017 г.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	1. Файлы метаданных 2. Техническое задание
Перечень подлежащих исследованию, проектированию и разработке вопросов	1. Обзор предметной области 2. Формирование представления данных в синтаксисе Turtle 3. Обоснование выбора языка программирования 4. Реализация модуля 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение 6. Социальная ответственность

Перечень графического материала	1. Общая архитектура системы 2. Таблица используемых классов и свойств 3. Модель предметной области
Консультанты по разделам выпускной квалификационной работы (с указанием разделов)	
Раздел	Консультант
Раздел 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Тухватулина Лилия Равильевна
Раздел 6. Социальная ответственность	Пустовойтова Марина Игоревна

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	
---	--

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ведущий программист	Губин Максим Юрьевич			

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К31	Кайда Анастасия Юрьевна		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8K31	Кайда Анастасии Юрьевне

Институт	ИК	Кафедра	ПИ
Уровень образования	Бакалавриат	Направление/специальность	09.03.03 Прикладная информатика

Исходные данные к разделу «Социальная ответственность»:

1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения.	Разработка модуля для информационной системы на рабочем месте, при помощи персонального компьютера, монитора, клавиатуры и компьютерной мыши.
---	---

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Производственная безопасность: 1.1. Анализ вредных факторов проектируемой производственной среды в следующей последовательности: – физико-химическая природа вредности, её связь с разрабатываемой темой; – действие фактора на организм человека; – приведение допустимых норм с необходимой размерностью (со ссылкой на соответствующий нормативно-технический документ); – предлагаемые средства защиты – (сначала коллективной защиты, затем – индивидуальные защитные средства) 1.2. Анализ опасных факторов проектируемой произведённой среды в следующей последовательности: – термические опасности (источники, средства защиты); – электробезопасность (в т.ч. статическое электричество, молниезащита – источники, средства защиты); – пожаровзрывобезопасность (причины, профилактические мероприятия, первичные средства пожаротушения).	Анализ вредных факторов: – Параметры микроклимата по СанПиН 2.2.2/ 2.4.1340-03 – Освещенность рабочего места по СанПиН 2.2.1/2.1.1.1278-03 – Уровень шума по СанПиН 2.2.4.3359-16 – Умственное перенапряжение по ТОО Р-45-084-01 Анализ опасных факторов: – Опасность поражения электрическим током по ГОСТ Р 50571. 17-2000
2. Экологическая безопасность: – защита селитебной зоны; – анализ воздействия объекта на атмосферу (выбросы); – анализ воздействия объекта на гидросферу (сбросы);	– Утилизация компьютерного оборудования по ГОСТ Р 56397-2015. – Утилизация люминесцентных ламп по ГОСТ 12.3.031-83.

– анализ воздействия объекта на литосферу (отходы); – разработать решения по обеспечению экологической безопасности со ссылками на НТД по охране окружающей среды.	
3. Безопасность в чрезвычайных ситуациях: – перечень возможных ЧС при разработке и эксплуатации проектируемого решения; – выбор наиболее типичной ЧС; – разработка превентивных мер по предупреждению ЧС; – разработка действий в результате возникшей ЧС и мер по ликвидации её последствий.	Вероятным ЧС является пожар в результате возгорания электропроводки, перегрева рабочих частей ПК. Ознакомление с планом эвакуации и использованием огнетушителей ОУ-5.
4. Правовые и организационные вопросы обеспечения безопасности: – специальные (характерные для проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны	Организация рабочего места согласно ГОСТ 12.2.032-78 Анализ конструкции рабочей мебели для работ сидя, согласно ГОСТ 12.2.061-81
Перечень графического материала:	
1. Схема эвакуации	

Дата выдачи задания для раздела по линейному графику	
---	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Пустовойтова М.И.	к.х.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8К31	Кайда Анастасия Юрьевна		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8K31	Кайда Анастасии Юрьевне

Институт	ИК	Кафедра	ПИ
Уровень образования	Бакалавриат	Направление/специальность	09.03.03 Прикладная информатика

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Работа с информацией, предоставленной в российских и иностранных научных публикациях, аналитических материалах, статических бюллетенях и изданиях, нормативно-правовых документах.
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	SWOT-анализ, анализ конкурентных решений
2. Планирование и формирование бюджета научных исследований	Планирование этапов работ, определение трудоемкости и построение календарного графика, формирование бюджета
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Оценка сравнительной эффективности исследования

Перечень графического материала (с точным указанием обязательных чертежей):

1. Матрица SWOT 2. График Ганта

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент каф. МЕН	Тухватулина Л.Р.	к.ф.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8K31	Кайда Анастасия Юрьевна		

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт _____ Институт Кибернетики _____
Направление подготовки (специальность) _____ Прикладная информатика _____
Уровень образования _____ Бакалавр _____
Кафедра _____ Программной инженерии _____
Период выполнения _____ (осенний / весенний семестр 2016/2017 учебного года)

Форма представления работы:

бакалаврская работа

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

**КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы**

Срок сдачи студентом выполненной работы:	16 июня 2017 г.
--	-----------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
20.04.2017	<i>Раздел 1. Обзор предметной области</i>	
04.05.2017	<i>Раздел 2. Формирование представления данных в синтаксисе Turtle</i>	
25.05.2017	<i>Раздел 3. Обоснование выбора языка программирования</i>	
29.05.2017	<i>Раздел 4. Реализация модуля</i>	
23.05.2017	<i>Раздел 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</i>	
24.05.2017	<i>Раздел 6. Социальная ответственность</i>	

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ведущий программист	Губин М.Ю.			

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ПИ	Иванов М.А.	К.Т.Н.		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ООП

Код результата	Результат обучения
Профессиональные компетенции	
P1	Применять базовые и специальные естественно-научные и математические знания в области информатики, экономики, маркетинга и менеджмента, достаточные для комплексной инженерной деятельности.
P2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных и экономических задач.
P3	Ставить и решать задачи комплексного анализа, связанные с созданием новых информационных технологий и информационных систем в экономике, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
P4	Разрабатывать новые и модернизировать уже существующие информационные технологии и системы (в экономике) в соответствии с техническим заданием.
P5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области прикладной информатики. Проводить исследования, связанные с оценкой информационной безопасности проектов.
P6	Внедрять, эксплуатировать и обслуживать современные информационные технологии и системы, обеспечивать их

	высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
Универсальные компетенции	
P7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
P8	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
P9	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.
P10	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.
P11	Демонстрировать способность к самостоятельной к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии.

Реферат

Выпускная квалификационная работа содержит: 77 страниц, 13 рисунков, 24 таблицы, 16 источников, 3 приложения.

Ключевые слова: графовая модель данных, импорт метаданных, онтология, связанные данные, Python.

Объектом исследования являются метаданные научного эксперимента, извлекаемые из различных систем хранения.

Цель работы – обеспечить извлечение и конвертацию необходимых метаданных из файлов с определенной структурой.

В процессе исследования проводилось изучение механизмов работы базы знаний научного эксперимента, был рассмотрен синтаксис представления данных и структура онтологической модели данных.

В результате исследования был разработан прототип модуля импорта метаданных, позволяющий быстро извлекать данные из файлов JSON и TXT с последующей записью их в виде триплетов в синтаксисе Turtle в готовые для загрузки в систему файлы.

Степень внедрения: в данный момент сама база знаний, для которой был разработан данный модуль, находится в стадии прототипа. К концу 2017 г. планируется ее внедрение, а вместе с тем и внедрение разработки.

Область применения разработки: база знаний научного эксперимента, создаваемая для европейской организации по ядерным исследованиям (CERN).

Перечень условных обозначений, единиц и терминов

DKB – Data Knowledge Base, база знаний научного эксперимента.

HTTP – HyperText Transfer Protocol (протокол передачи гипертекста).
Протокол прикладного уровня передачи данных, в первую очередь, в виде текстовых сообщений.

JSON – JavaScript Object Notation, текстовый формат обмена данными.

N-Triples – текстовый формат, используемый для хранения и передачи графов.

OWL – Web Ontology Language. Язык описания онтологий.

Python – высокоуровневый язык программирования общего назначения.

RDF – Resource Description Framework, модель для представления данных, разработанная консорциумом Всемирной паутины.

RDFS – RDF Schema language, является языком с средствами выразительности достаточными для описания базовых понятий общности и изменчивости.

Semantic Web – семантическая паутина, общедоступная глобальная семантическая сеть, формируемая на базе Всемирной паутины путём стандартизации представления информации в виде, пригодном для машинной обработки.

SPARQL – язык запросов к данным, а также протокол для передачи этих запросов и ответов на них.

Turtle – Terse RDF Triple Language, специальный синтаксис для записи RDF-триплетов.

TXT – текстовый файл.

Unicode – стандарт кодирования символов, позволяющий представить знаки почти всех письменных языков.

URI – Uniform Resource Identifier, унифицированный (единообразный) идентификатор ресурса.

W3C – World Wide Web Consortium, консорциум Всемирной паутины, организация, разрабатывающая и внедряющая технологические стандарты для Всемирной паутины.

Онтология – модель данных, представляющая собой ориентированный граф. Состоит из экземпляров, понятий, атрибутов и отношений.

Триплет – тройка «субъект-предикат-объект».

Оглавление

Реферат	10
Перечень условных обозначений, единиц и терминов	11
Введение.....	15
1. Обзор предметной области	16
1.1. Описание предметной области.....	16
1.2. Аналоги разрабатываемого модуля	17
1.3. Преимущества используемых технологий.....	17
1.4. Функциональные требования к разработке	19
2. Формирование представления данных в синтаксисе Turtle	21
2.1. Анализ структуры модели данных	21
2.2. Анализ входных данных	23
2.3. Формирование триплетов	26
3. Обоснование выбора языка программирования	28
4. Реализация модуля.....	29
4.1. Реализация пользовательского интерфейса.....	29
4.2. Импорт метаданных из TXT-файлов.....	30
4.3. Импорт метаданных из JSON-файлов	32
4.3.1. Импорт данных из хорошо структурированных файлов	32
4.3.2. Импорт данных из плохо структурированных файлов	33
5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	36
5.1. Анализ конкурентных технических решений	36
5.2. SWOT-анализ	37
5.3. Определение возможных альтернатив проведения научных исследований	39
5.3.1. Структура работ в рамках научного исследования	39
5.3.2. Определение трудоемкости выполнения работ	40
5.3.3. Разработка графика проведения научного исследования	41
5.3.4. Бюджет научно-технического исследования (НТИ)	44
5.3.4.1. Расчет материальных затрат НТИ.....	44

5.3.4.2. Расчет затрат на специальное оборудование для научных (экспериментальных) работ	46
5.3.4.3. Основная заработная плата исполнителей темы	46
5.3.4.4. Дополнительная заработная плата исполнителей темы	48
5.3.4.5. Отчисления во внебюджетные фонды (страховые отчисления)	48
5.3.4.6. Расчет затрат на научные и производственные командировки	50
5.3.4.7. Контрагентные расходы	50
5.3.4.8. Накладные расходы	50
5.3.4.9. Формирование бюджета затрат научно-исследовательского проекта	51
5.3.5. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	51
6. Социальная ответственность	55
6.1. Производственная безопасность	55
6.1.1. Параметры микроклимата	56
6.1.2. Освещенность рабочего места	57
6.1.3. Уровень шума	59
6.1.4. Умственное перенапряжение	60
6.1.5. Опасность поражения электрическим током	61
6.2. Экологическая безопасность	63
6.3. Безопасность в чрезвычайных ситуациях	64
6.4. Правовые и организационные вопросы обеспечения безопасности...	65
Заключение	67
Список использованных источников	68
Приложение А	70
Приложение Б	73
Приложение В	75

Введение

В настоящее время научные эксперименты сопряжены с интенсивной обработкой данных, требующей высокой скорости не только записи первичных и преобразованных данных об эксперименте, но и доступности в последующем. Большое количество разрозненных источников требует объединения данных в единую систему, а также программных компонентов, реализующих преобразование и импорт данных, полученных из первоисточника, в унифицированный формат.

Проблемой является отсутствие на рынке программных продуктов, соответствующих требованиям создателей базы знаний научного эксперимента и способных решить поставленную задачу.

Объектом исследования являются метаданные научного эксперимента. Предметом исследования является способ их преобразования и представления в виде триплетов для онтологической модели.

Целью работы является разработка модуля с целью сделать его связующим звеном между поступающими на входе файлами, получаемыми из других репозиторий, и хранилищем данных базы знаний научного эксперимента.

С практической точки зрения, модифицированные версии модуля могут быть использованы для импорта метаданных из других аналогичных источников, подключаемых в будущем.

1. Обзор предметной области

1.1. Описание предметной области

В современных научных экспериментах активно используется разнообразное программное обеспечение, фиксирующее и хранящее результаты в разнообразном виде. При необходимости нужные данные можно извлечь из хранилища для последующей обработки. Однако одной из проблем современной науки в некоторых научных экспериментах является быстрый рост объема разнородных данных, достигающий петабайт информации.

Данная проблема коснулась и научных коллабораций, ведущих свои исследования в проектах CERN, таких как ATLAS, ALICE, LHC и т.д. Поиск в больших объемах разнородных данных становится затруднителен, требует большого количества времени и высоких вычислительных мощностей. В связи с этим было принято решение о создании базы знаний научного эксперимента (Data Knowledge Base). База знаний включает в себя набор классов, объекты которых описывают научный эксперимент, набор свойств этих объектов и свойств данных.

В 2006 году была опубликована статья Тима Бернерса-Ли [1], описывающая подход представления информации, со слов автора, «простой, но до сих пор в большей степени, нереализованный». В основе представления лежит концепция так называемых связанных данных, представленных в общем виде в RDF.

Одной из основных задач DKB является интеграция гетерогенной информации в RDF-хранилище на базе Virtuoso Universal Server.

Разрабатываемый модуль может решить задачу по импорту метаданных из файлов текстовых форматов, легко читаемых людьми, в частности TXT и JSON. Модуль реализует извлечение из входных файлов информации о проектах, кампаниях и периодах получения данных, которые используются в модели в качестве общих атрибутов, формирование полученных данных в RDF-триплеты и запись в готовые для импорта в Virtuoso файлы.

1.2. Аналоги разрабатываемого модуля

В силу своей специфики разрабатываемый модуль не имеет прямых аналогов, поскольку сам модуль предназначен для импорта в базу знаний, применяющую концептуально новый подход, в основе которого заложена идея создания RDF-хранилища, объединяющего данные из обширного перечня источников. Помимо этого, стоит отметить, что работу со столь большими объемами данных, соразмерными с объемами, получаемыми с детекторов в европейской организации ядерных исследований, ведут далеко не все организации, что делает разрабатываемый модуль менее востребованным для широкого круга потребителей.

Из схожих разработок известны RDF-Translator[2] и EasyRDF[3], представленные в виде web-сервисов и выполняющие схожие друг с другом функции. Стоит отметить, что ни тот, ни другой не осуществляют извлечение данных из файлов нужного типа, а также не предоставляют возможности встраивания в общую систему. Более того, RDF-Translator не работает с синтаксисом Turtle (представляет данные в схожем синтаксисе N-Triples), как таковым.

Основная проблема заключается в том, что нынешние разработки поддерживают только строго структурированные форматы, являющиеся стандартами консорциума W3C, в то время как источниками данных могут оказаться данные из источников с иными расширениями.

1.3. Преимущества используемых технологий

Основными принципами концепции связанных данных являются:

- Использование URI для объектов;
- Использование HTTP URI для простоты нахождения имен объектов;
- Представление информации с использованием принятых стандартов (RDF и SPARQL);

- Добавлять ссылки на другие URI, тем самым позволяя пользователям открывать и другие объекты.

При возникновении необходимости использовать данные из различных источников, потребуется выполнять соединение (слияние) содержания этих источников в единую модель. Одним из достоинств представления информации в виде триплетов является легкость, с которой такого рода слияние может быть выполнено. Так как информация описана в виде триплетов, то соединить информацию из двух графов так же просто, как и сформировать граф всех триплетов из каждого отдельного графа, объединив их.

Стоит отметить, что URI идентифицирует как абстрактный, так и физический ресурс, что позволяет ссылаться на объект любого типа. Документально нет ограничений на использование множества пространств имен в одном источнике данных, или даже в одном триplete. Выбор пространств имен полностью неограничен, того, что касается модели данных и стандартов. Однако достаточно распространенным приемом является ссылка на взаимосвязанные идентификаторы в одном пространстве имен.

Модель данных, основанная на RDF, является стандартом, позволяющим реализовать эту концепцию на практике, и представляет информацию как ориентированный граф (граф, имеющий определенное направление). Такая модель данных предназначена для представления интегрированной гетерогенной информации, взятой из множества разнородных источников. В RDF данные представляются в виде троек «субъект-предикат-объект» – триплетов.

Субъект представляет собой унифицированный идентификатор ресурса URI, ссылающийся на конкретный описываемый объект. Объект может быть представлен как в виде литерала, так и в виде ссылки URI. В свою очередь, предикат, соединяющий субъект и объект, является своеобразным индикатором, указывающим на то, как связаны объект и субъект в модели. Каждое свойство может обладать рядом характеристик, таких как

транзитивность, рефлексивность, иррефлексивность, симметричность, инверсия и т.д.

В контексте концепции связанных данных RDF-модель имеет следующий ряд преимуществ:

- Такая модель данных позволяет задать RDF-ссылки между объектами, ссылающимися на различные ресурсы;
- Пользователи могут искать любой доступный URI посредством сети Интернет для получения дополнительной информации;
- Информация, собранная с различных ресурсов, может быть скомбинирована путем объединения наборов триплетов в единый граф;
- RDF позволяет представить информацию, описанную с помощью различных схем, в едином виде внутри одного графа, при этом разные значения, обозначающие одно и то же, могут быть объединены в словаре.

1.4. Функциональные требования к разработке

Разработка предназначена для включения в общую архитектуру системы по интеграции данных в базу знаний и несет в себе цель автоматизации процесса импорта в режиме обмена потоками данных.

Модуль должен реализовывать:

- Определение формата входного файла;
- Извлечение данных из TXT и JSON;
- Построчное чтение текстовых файлов;
- Удаление лишних символов в литералах;
- Опцию изменения пути графа/онтологии;
- Формирование триплетов;
- Запись в файл.

Основная функция модуля – подготовка данных. Роль модуля в общей архитектуре системы согласно концепции связанных данных [4] представлена на рисунке 1.

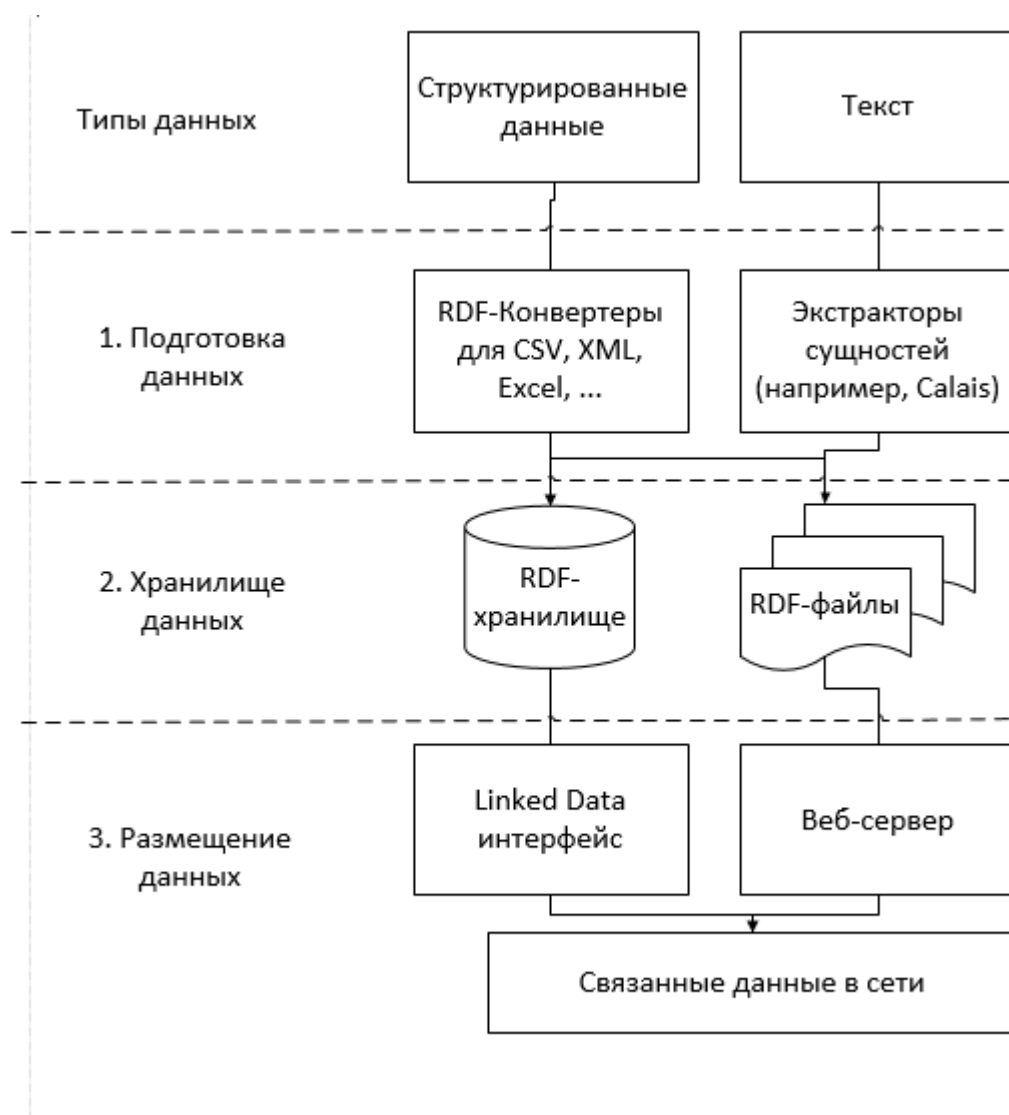


Рисунок 1. Общая архитектура системы согласно концепции связанных данных

Разрабатываемый модуль выполняет функцию подготовки данных наряду с RDF-конвертерами согласно представленной схеме. Единственное отличие в том, что утвержденным форматом для импорта является Turtle с упрощенным синтаксисом.

2. Формирование представления данных в синтаксисе Turtle

2.1. Анализ структуры модели данных

Исходная модель данных, на основе структуры которой должны быть сформированы шаблоны триплетов, представлена на языке OWL – среди языков моделирования Семантической паутины, таких как RDF и RDFS, язык OWL является самым выразительным. В частности, он позволяет разработчикам моделей подробно описать ограничения между классами, сущностями и свойствами.

Для формирования шаблонов триплетов была проанализирована модель на языке OWL. Используемые сущности и связи представлены в таблице 1.

Таблица 1. Используемые классы и свойства

Название	Принадлежность	Описание
Campaign	Classes	Класс, содержащий названия кампаний
DataSampleFormat	Classes	Класс, содержащий один из атрибутов полученных образцов данных
DataTakingPeriod	Classes	Класс, содержащий периоды снятия данных
Level	Classes	Класс, содержащий уровни
Period	Classes	Класс, содержащий периоды проведения эксперимента
ProductionStep	Classes	Класс, содержащий ступени
Project	Classes	Класс, содержащий названия проектов физического анализа
Year	Classes	Класс, содержащий годы проведения экспериментов

hasDescription	Data properties	Описание субъекта: субъект имеет описание (в виде литерала)
hasStatus	Data properties	Описание субъекта: субъект имеет статус (в виде литерала)
hasDataSampleFormat	Object properties	Описание субъекта: субъект имеет атрибут DataSampleFormat
hasDataTakingPeriod	Object properties	Описание субъекта: субъект имеет определенный период снятия данных
hasLevel	Object properties	Описание субъекта: субъект имеет уровень
hasProductionStep	Object properties	Описание субъекта: субъект имеет ступень
hasProject	Object properties	Описание субъекта: субъект включает в себя некоторый проект
hasYear	Object properties	Описание субъекта: субъект относится к определенному году проведения эксперимента

В качестве субъектов выступают только объекты классов (принадлежность – Classes). Объектами являются как объекты классов, так и литералы (Data properties). В качестве связей выступают свойства объектов (Object properties).

2.2. Анализ входных данных

Входные данные представлены в виде файлов с расширениями .txt или .json. В файлах с расширением .txt передаются данные по имеющимся проектам, в файлах JSON представлены в виде списка словарей или в виде списка вложенных словарей. Сложность второго варианта заключается в отсутствии в парах «ключ-значение» ключа, по которому можно было бы произвести поиск необходимого словаря и вложенных словарей, в связи с чем увеличивается алгоритмическая сложность.

Исходная модель данных является сложной многоуровневой структурой с большим количеством разнообразных связей. Классы сущностей, данные по которым будут извлекаться из файлов, представлены на рисунках 2 и 3.

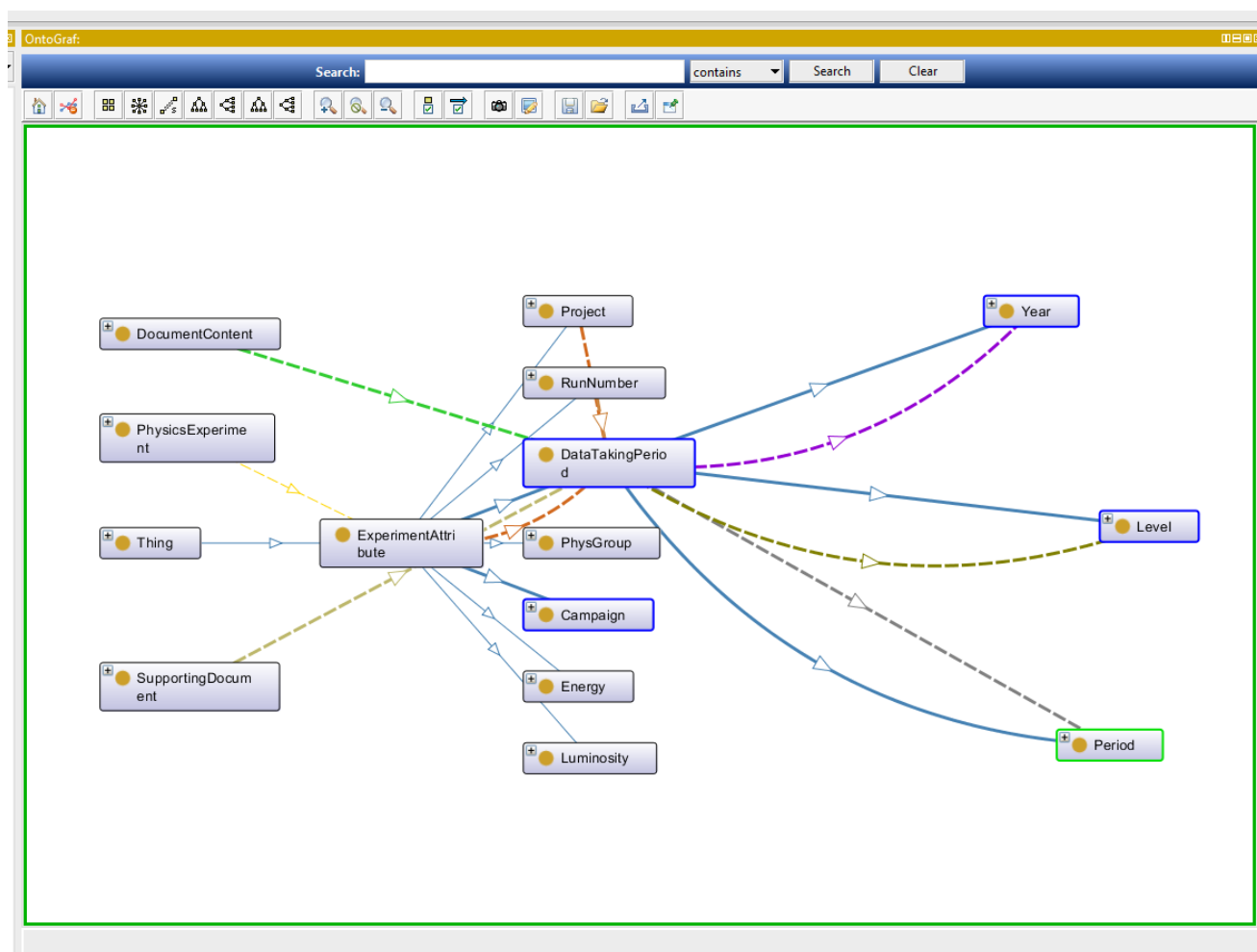


Рисунок 2. Часть онтологической модели данных

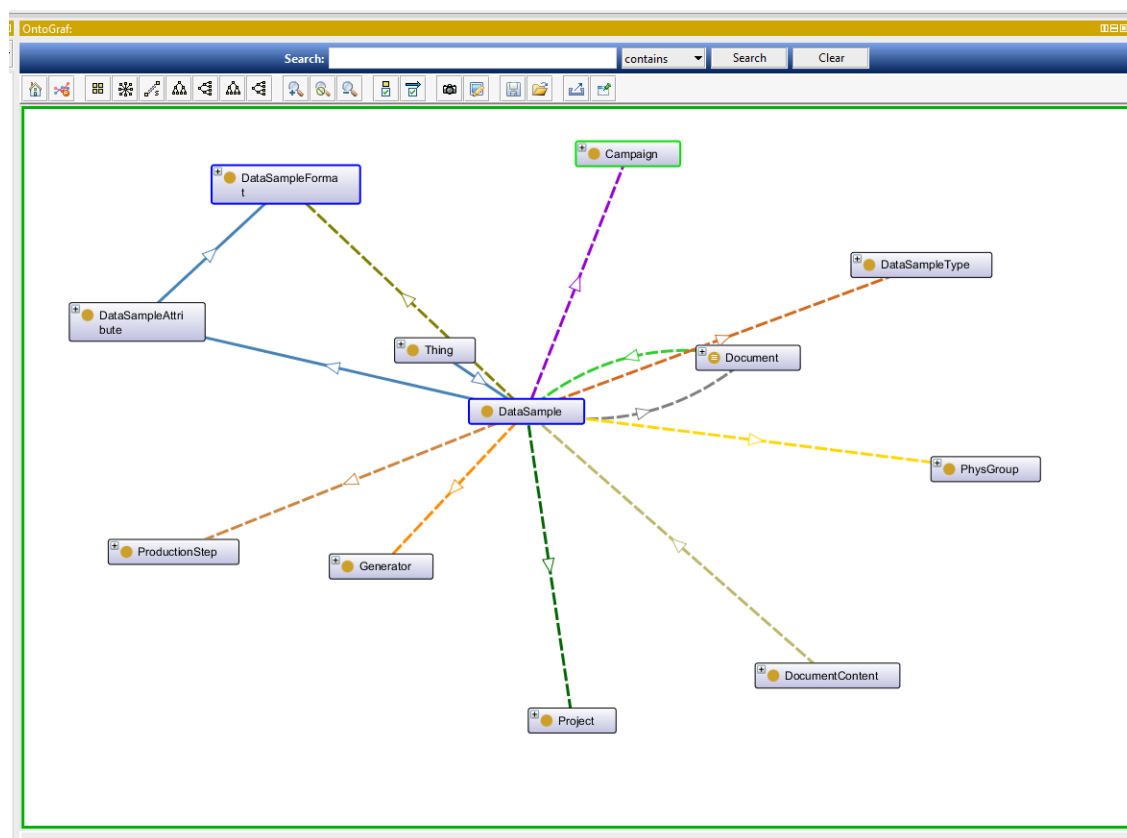


Рисунок 3. Часть онтологической модели данных

На данном этапе в онтологическую модель данных уже занесены некоторые экземпляры классов (Instances), что продемонстрировано на рисунках 4 и 5. Стоит отметить, что это помогает связать данные из файлов, где в словарях в паре «ключ-значение» вместо ключа указано одно из значений, с определенным классом. Зачастую невозможно определить без набора исходных данных, добавленных в модель, определились в файлах с такого рода структурой, какой субъект и предикат будет записан в триплет с объектом.

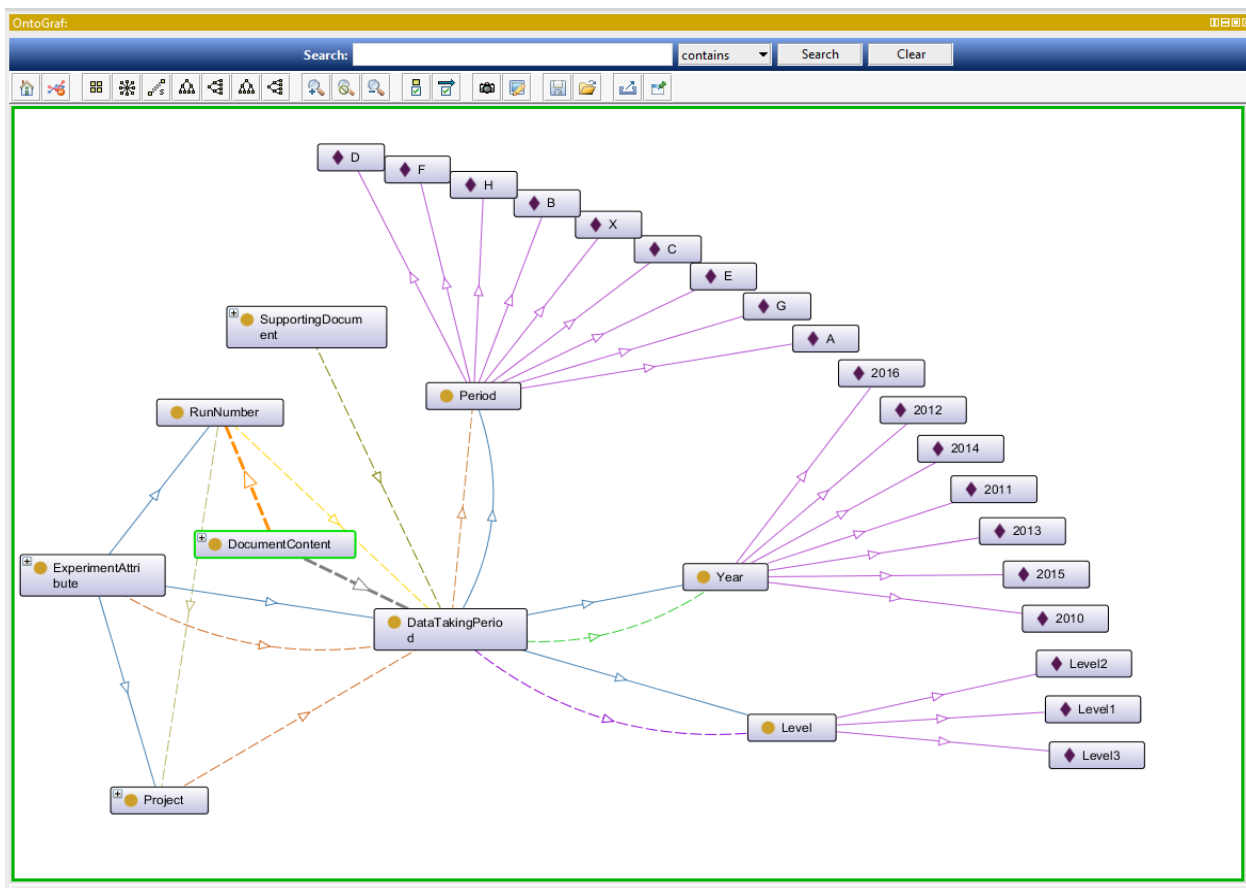


Рисунок 4. Экземпляры классов, относящиеся к DataTakingPeriod

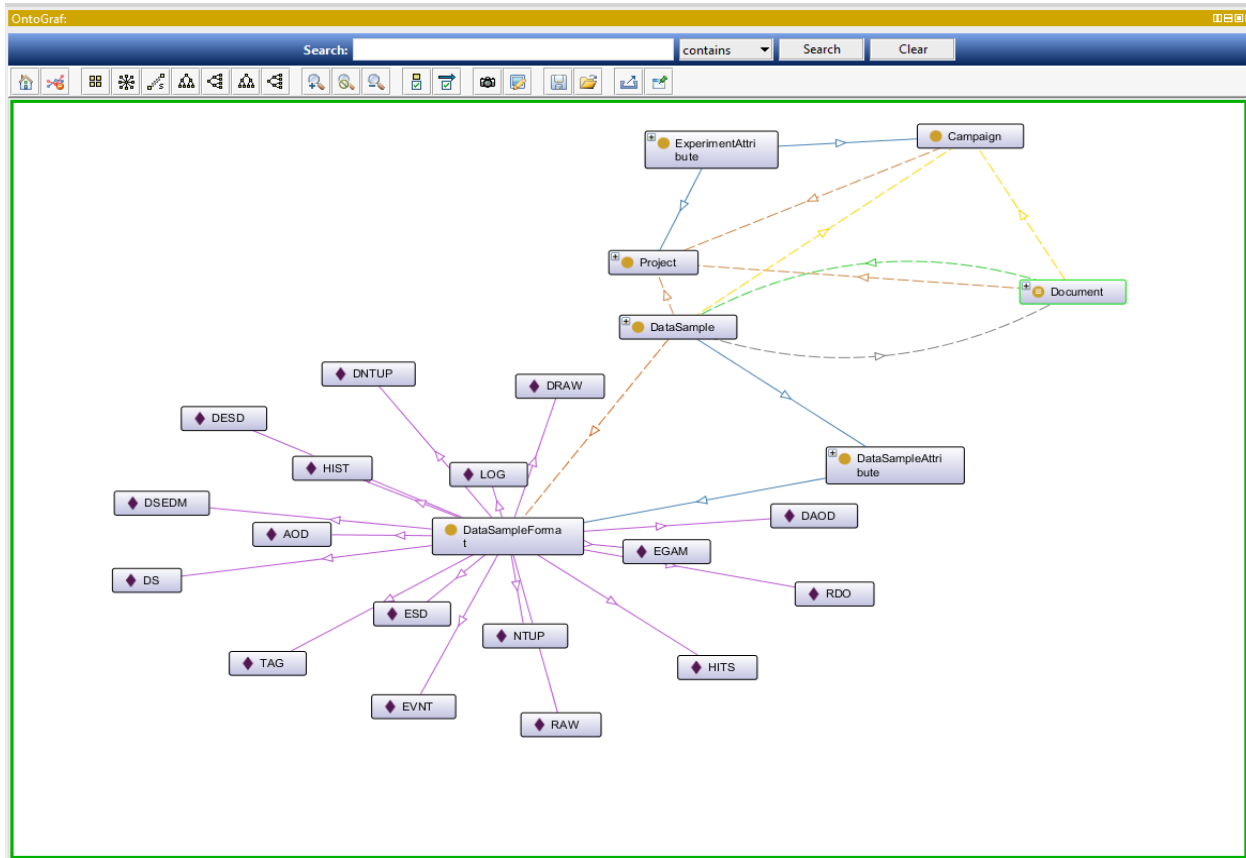


Рисунок 5. Экземпляры классов, относящиеся к Campaigns

2.3. Формирование триплетов

Как было описано ранее, выходные данные записываются с использованием синтаксиса Turtle. Turtle – конкретизированный синтаксис записи RDF-триплетов, расширяющий другой синтаксис N-Triples. Turtle позволяет представить триплет в компактной текстовой форме. В триплете три ресурса записываются в порядке субъект/предикат/объект, после которых ставится точка. Точка указывает на окончание триплета.

При определении субъекта, предиката и объекта стоит обратить внимание на некоторые аспекты:

- Возможные повторения;
- Необозначенная транзитивность;
- Необъявленный элемент класса в модели;
- Области определения и области значения свойств объектов онтологии;
- Единообразие структуры (разноименное обозначение одних и тех же связей между объектами онтологии может в дальнейшем привести к трудностям при обращении к данным);
- Пустые вершины;
- Исключения.

Наибольшую опасность представляют неправильно заданные связи между сущностями. Несмотря на то, что практика допускает использование словарей синонимов, в текущей модели данных они отсутствуют, и создание синонимов крайне нежелательно, поскольку впоследствии это может нарушить корректную работу с данными и усложнить поиск необходимых данных, что противоречит первоначальной идее создания базы знаний.

Выделенные триплеты представлены на рисунке 6.

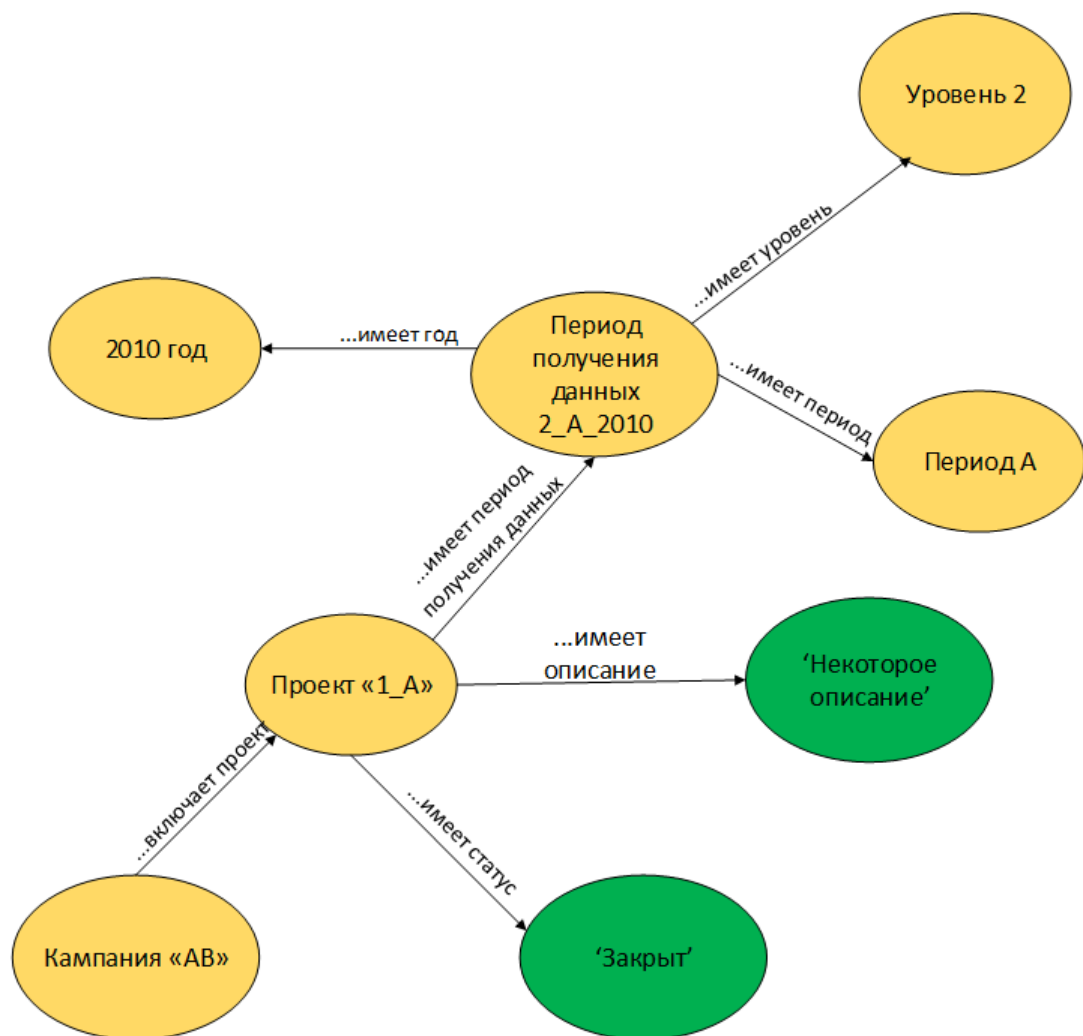


Рисунок 6. Интерпретация связей между сущностями

3. Обоснование выбора языка программирования

Для реализации модуля выбран язык программирования Python. Данный высокоуровневый язык программирования отлично подходит для написания скриптов. Следует отметить некоторые из его особенностей:

- Относительно высокая читаемость кода
- Поддержка нескольких парадигм программирования;
- Автоматическое управление памятью;
- Высокоуровневые структуры данных;
- Большое количество стандартных библиотек;
- Переносимость кода между платформами.

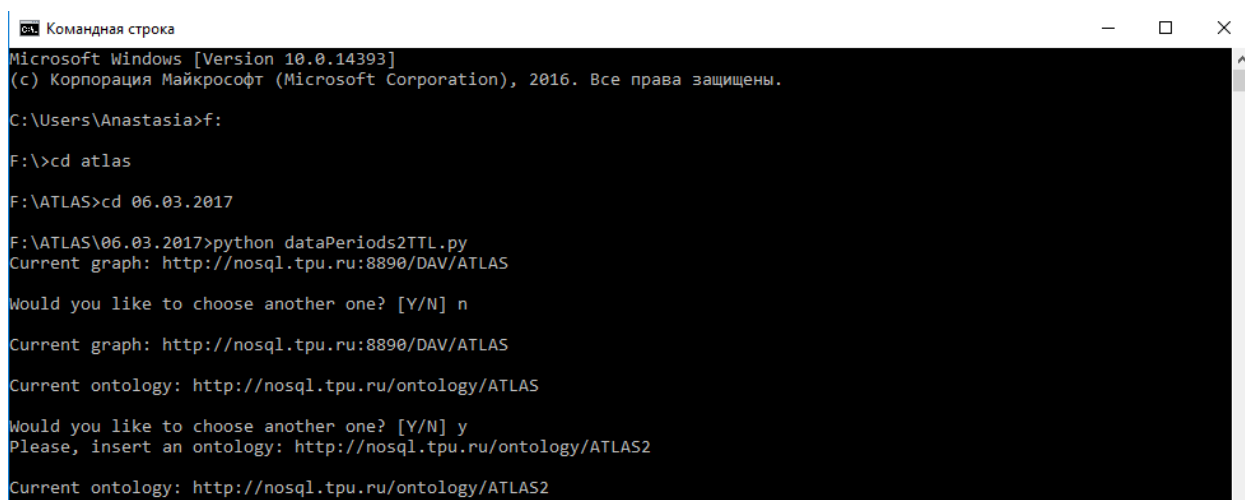
Python поддерживает динамическую типизацию, то есть тип переменной определяется только во время исполнения. В Python имеются встроенные типы: булевый, строка, Unicode-строка, целое число произвольной точности, число с плавающей запятой, комплексное число и некоторые другие. Из коллекций в Python встроены: список, кортеж (неизменяемый список), словарь, множество и другие. Все значения являются объектами, в том числе функции, методы, модули, классы [5]. В силу работы со структурированными файлами поддержка списков и словарей, а также библиотеки json значительно упрощает задачу по извлечению и представлению данных.

4. Реализация модуля

4.1. Реализация пользовательского интерфейса

На данном этапе разработка имеет пользовательский текстовый интерфейс для возможности изменять параметры пути и выбирать загружаемые файлы вручную для отслеживания показателей скорости выполнения и потребления памяти.

В элементы модуля встроены по умолчанию адреса графа и онтологии, которые отображаются при запуске скрипта. Пользователю предоставляется выбор, оставить ли адрес прежним или изменить его. Диалоговые элементы реализованы на английском языке. Пример представлен на рисунке 7.



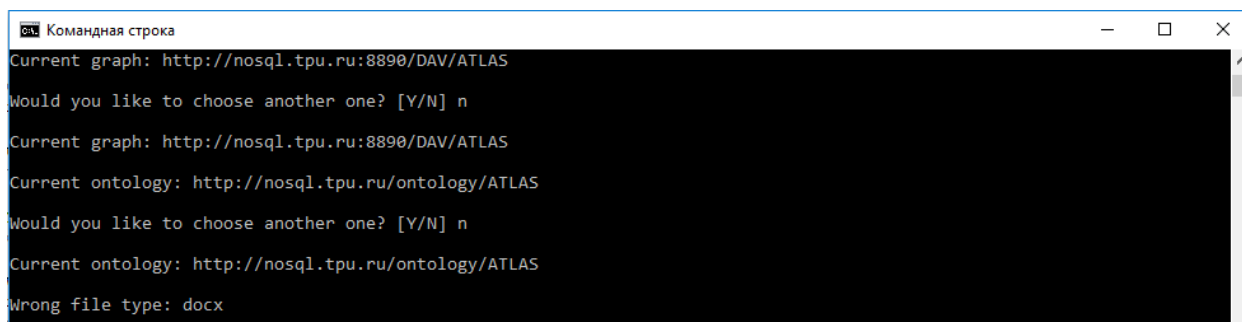
```
Командная строка
Microsoft Windows [Version 10.0.14393]
(c) Корпорация Майкрософт (Microsoft Corporation), 2016. Все права защищены.

C:\Users\Anastasia>f:
F:\>cd atlas
F:\ATLAS>cd 06.03.2017
F:\ATLAS\06.03.2017>python dataPeriods2TTL.py
Current graph: http://nosql.tpu.ru:8890/DAV/ATLAS
Would you like to choose another one? [Y/N] n
Current graph: http://nosql.tpu.ru:8890/DAV/ATLAS
Current ontology: http://nosql.tpu.ru/ontology/ATLAS
Would you like to choose another one? [Y/N] y
Please, insert an ontology: http://nosql.tpu.ru/ontology/ATLAS2
Current ontology: http://nosql.tpu.ru/ontology/ATLAS2
```

Рисунок 7. Опция выбора онтологии и графа

После утверждения пользователем графа и онтологии, которые будут указываться в TTL-файлах, как часть триплета, пользователю предоставляется возможность выбора загружаемого файла с помощью диалогового окна. Для реализации этого элемента подходят такие библиотеки, как PyQt5.QtWidgets или Tkinter. Вторая библиотека является кросс-платформенной, входит в стандартную библиотеку Python и не требует работы с дополнительными фреймворками, что хорошо подходит для решения поставленной задачи.

При неправильно выбранном файле выполнение скрипта будет преждевременно завершено, а неправильный тип файла выведен в консоль, как показано на рисунке 8.



```
Командная строка
Current graph: http://nosql.tpu.ru:8890/DAV/ATLAS
Would you like to choose another one? [Y/N] n
Current graph: http://nosql.tpu.ru:8890/DAV/ATLAS
Current ontology: http://nosql.tpu.ru/ontology/ATLAS
Would you like to choose another one? [Y/N] n
Current ontology: http://nosql.tpu.ru/ontology/ATLAS
Wrong file type: docx
```

Рисунок 8. Прекращение работы программы при неправильном типе файла

4.2. Импорт метаданных из TXT-файлов

На вход поступают текстовые файлы, содержащий некоторый список проектов, описываемых в базе знаний научного эксперимента. В начале такого вида файлов возможны комментарии. Остальные данные файла являются однородными. Реализация предполагает построчное чтение файла и, при нахождении строки, не являющейся комментарием, запись триплета, как показано в примере:

- `<http://nosql.tpu.ru:8890/DAV/ATLAS/project/cmcccond> a`
`<http://nosql.tpu.ru/ontology/ATLAS#Project> .`
- `<http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond09> a`
`<http://nosql.tpu.ru/ontology/ATLAS#Project> .`
- `<http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond09_data> a`
`<http://nosql.tpu.ru/ontology/ATLAS#Project> .`
- `<http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond09_mc> a`
`<http://nosql.tpu.ru/ontology/ATLAS#Project> .`
- `<http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond09_test> a`
`<http://nosql.tpu.ru/ontology/ATLAS#Project> .`
- `<http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond10> a`
`<http://nosql.tpu.ru/ontology/ATLAS#Project> .`

Пример представлен на рисунке 9.

```
Командная строка
Current ontology: http://nosql.tpu.ru/ontology/ATLAS
Would you like to choose another one? [Y/N] n
Current ontology: http://nosql.tpu.ru/ontology/ATLAS
Processing...

A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cmccond> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond09> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond09_data> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond09_mc> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond09_test> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond10> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond10_data> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond10_mc> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond11> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond11_data> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond11_mc> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
```

Рисунок 9. Выполнение скрипта

Выходные данные формируются триплетами и записываются в файл с расширением .ttl. Такие файлы готовы для импорта в RDF-хранилище. Пример содержимого файла представлен на рисунке 10.

```
projects.ttl
1 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cmccond> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
2 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond09> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
3 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond09_data> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
4 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond09_mc> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
5 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond09_test> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
6 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond10> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
7 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond10_data> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
8 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond10_mc> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
9 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond11> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
10 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond11_data> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
11 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond11_mc> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
12 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond11_test> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
13 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond12> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
14 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cond12_data> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
15 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/condR2> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
16 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/condR2_data> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
17 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/cvalid> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
18 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
19 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data08> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
20 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data08_1beam> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
21 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data08_1beammap> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
22 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data08_calib> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
23 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data08_calocomm> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
24 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data08_coll900> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
25 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data08_cos> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
26 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data08_cosm6> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
27 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data08_cosm7> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
28 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data08_cosmap> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
29 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data08_idcomm> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
30 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data08_larcomm> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
31 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data08_muoncomm> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
32 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data08_rof> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
33 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data08_tilecomm> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
34 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data09> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
35 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data09_1beam> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
36 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data09_2TeV> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
37 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data09_900GeV> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
38 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data09_calib> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
39 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data09_calocomm> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
40 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data09_calophvs> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
41 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data09_cos> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
42 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data09_cvalid> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
43 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data09_idcomm> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
44 <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data09_larcomm> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
```

Рисунок 10. Пример записанного файла с расширением .ttl

4.3. Импорт метаданных из JSON-файлов

4.3.1. Импорт данных из хорошо структурированных файлов

Одним из видов входных данных являются данные из JSON-файлов, интерпретируемые программой при считывании как список словарей с несколькими парами «ключ-значение». Такая структура удобна при оптимизации кода и дает возможность создать универсальный конвертер, позволяющий определять необходимую для выделения информацию «на лету» в дальнейшем при появлении новых подобных источников данных. Возможность идентифицировать, к какой сущности относятся ключи, не обращаясь к общей онтологии, упрощает алгоритмическую сложность от $O(N)$, если бы поиск проводился по списку, до $O(1)$ – в том случае если структура имеет один уровень.

Хорошо структурированными файлами являются файлы с данными по периодам получения данных в ходе научного эксперимента. Примеры формируемых триплетов:

- `<http://nosql.tpu.ru:8890/DAV/ATLAS/project/data10_7TeV>` a
`<http://nosql.tpu.ru/ontology/ATLAS#Project>` .
- `<http://nosql.tpu.ru:8890/DAV/ATLAS/project/data10_7TeV>`
`<http://nosql.tpu.ru/ontology/ATLAS#hasDescription>` 'unsqueezed stable beam data (beta*=10m): typical beam spot width in x and y is 50-60 microns.' .
- `<http://nosql.tpu.ru:8890/DAV/ATLAS/project/data10_7TeV>`
`<http://nosql.tpu.ru/ontology/ATLAS#hasStatus>` 'locked' .
- `<http://nosql.tpu.ru:8890/DAV/ATLAS/dataperiod/2010_A_2>` a
`<http://nosql.tpu.ru/ontology/ATLAS#DataTakingPeriod>` .
- `<http://nosql.tpu.ru:8890/DAV/ATLAS/dataperiod/2010_A_2>`
`<http://nosql.tpu.ru/ontology/ATLAS#hasYear>`
`<http://nosql.tpu.ru/ontology/ATLAS#2010>` .
- `<http://nosql.tpu.ru:8890/DAV/ATLAS/dataperiod/2010_A_2>`
`<http://nosql.tpu.ru/ontology/ATLAS#hasPeriod>`
`<http://nosql.tpu.ru/ontology/ATLAS#A>` .

- `<http://nosql.tpu.ru:8890/DAV/ATLAS/project/data10_7TeV>`
`<http://nosql.tpu.ru/ontology/ATLAS#hasLevel>`
`<http://nosql.tpu.ru/ontology/ATLAS#Level2>` .
- `<http://nosql.tpu.ru:8890/DAV/ATLAS/project/data10_7TeV>`
`<http://nosql.tpu.ru/ontology/ATLAS#hasDataTakingPeriod>`
`<http://nosql.tpu.ru:8890/DAV/ATLAS/dataperiod/2010_A_2>` .

Пример проиллюстрирован на рисунке 11.

```

Командная строка
Processing...
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data10_7TeV> a <http://nosql.tpu.ru/ontology/ATLAS#Project> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data10_7TeV> <http://nosql.tpu.ru/ontology/ATLAS#hasDescription> 'NULL' .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data10_7TeV> <http://nosql.tpu.ru/ontology/ATLAS#hasStatus> 'frozen' .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/dataperiod/2010_A_1> a <http://nosql.tpu.ru/ontology/ATLAS#DataTakingPeriod> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/dataperiod/2010_A_1> <http://nosql.tpu.ru/ontology/ATLAS#hasYear> <http://nosql.tpu.ru/ontology/ATLAS#2010> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/dataperiod/2010_A_1> <http://nosql.tpu.ru/ontology/ATLAS#hasPeriod> <http://nosql.tpu.ru/ontology/ATLAS#A1> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data10_7TeV> <http://nosql.tpu.ru/ontology/ATLAS#hasLevel> <http://nosql.tpu.ru/ontology/ATLAS#Level1> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data10_7TeV> <http://nosql.tpu.ru/ontology/ATLAS#hasDataTakingPeriod> <http://nosql.tpu.ru:8890/DAV/ATLAS/dataperiod/2010_A_1> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data10_7TeV> <http://nosql.tpu.ru/ontology/ATLAS#hasDescription> 'NULL' .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data10_7TeV> <http://nosql.tpu.ru/ontology/ATLAS#hasStatus> 'locked' .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/dataperiod/2010_B_1> a <http://nosql.tpu.ru/ontology/ATLAS#DataTakingPeriod> .

```

Рисунок 11. Выполнение скрипта

4.3.2. Импорт данных из плохо структурированных файлов

В процессе разработки возникла проблема реализации метода, описываемого в пункте 4.3.1. для файлов с данными о кампаниях. JSON-файлы представляют внутри список четырехуровневых словарей, в последнем уровне которого в паре «ключ-значение» в качестве ключа находится одна из сущностей `DataSampleFormat` вместо указания самого класса или синонимичного понятия, а в качестве значения представлен список элементов, не обозначенных внутри модели данных. На каждом уровне выше в качестве ключа выступает одна из сущностей определенного класса, а в качестве значения – вложенный словарь, что делает невозможным обращение к данным по ключу. Работа с такими данными происходит методом перебора всех элементов списка с последующим перебором всех вложений. На каждом уровне

формируется соответствующий триплет и позволяет работать с уже выделенными для записи графами. Так, в первом цикле формируются триплеты, объявляющие тот или иной граф с названием кампании сущностью класса Campaigns. Во вложенном цикле проверяются все вложенные названия проектов физического анализа и посредством предиката hasProject кампании присваивается тот или иной проект.

Триплет будет иметь формат:

```
{GRAPH}/campaign/{CAMPAIGN} {ONTOLOGY}#hasProject  
{GRAPH}/project/{PROJECT}.
```

На следующем уровне, если таковой имеется, кампании присваиваются указанные в записях ступени (ProductionStep). На последнем уровне присваивается DataSampleFormat. Вложенный в качестве значения список остается без изменения, для формирования триплета используется только название ключа.

Примеры триплетов:

- <http://nosql.tpu.ru:8890/DAV/ATLAS/campaign/t0pro20> a
<http://nosql.tpu.ru/ontology/ATLAS#Campaign> .
- <http://nosql.tpu.ru:8890/DAV/ATLAS/campaign/t0pro20>
<http://nosql.tpu.ru/ontology/ATLAS#hasProject>
<http://nosql.tpu.ru:8890/DAV/ATLAS/project/data16_13TeV> .
- <http://nosql.tpu.ru:8890/DAV/ATLAS/campaign/t0pro20>
<http://nosql.tpu.ru/ontology/ATLAS#hasProductionStep>
<http://nosql.tpu.ru/ontology/ATLAS#recon> .
- <http://nosql.tpu.ru:8890/DAV/ATLAS/campaign/t0pro20>
<http://nosql.tpu.ru/ontology/ATLAS#hasDataSampleFormat>
<http://nosql.tpu.ru/ontology/ATLAS#AOD> .
- <http://nosql.tpu.ru:8890/DAV/ATLAS/campaign/repro20> a
<http://nosql.tpu.ru/ontology/ATLAS#Campaign> .

- `<http://nosql.tpu.ru:8890/DAV/ATLAS/campaign/repro20>`
`<http://nosql.tpu.ru/ontology/ATLAS#hasProject>`
`<http://nosql.tpu.ru:8890/DAV/ATLAS/project/data15_13TeV> .`
- `<http://nosql.tpu.ru:8890/DAV/ATLAS/campaign/repro20>`
`<http://nosql.tpu.ru/ontology/ATLAS#hasProductionStep>`
`<http://nosql.tpu.ru/ontology/ATLAS#recon> .`
- `<http://nosql.tpu.ru:8890/DAV/ATLAS/campaign/repro20>`
`<http://nosql.tpu.ru/ontology/ATLAS#hasDataSampleFormat>`
`<http://nosql.tpu.ru/ontology/ATLAS#AOD> .`

Пример проиллюстрирован на рисунке 12.

```

Командная строка
Would you like to choose another one? [Y/N] n
Current graph: http://nosql.tpu.ru:8890/DAV/ATLAS
Current ontology: http://nosql.tpu.ru/ontology/ATLAS
Would you like to choose another one? [Y/N] n
Current ontology: http://nosql.tpu.ru/ontology/ATLAS

Processing...
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/campaign/t0pro20> a <http://nosql.tpu.ru/ontology/ATLAS#Campaign> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/campaign/t0pro20> <http://nosql.tpu.ru/ontology/ATLAS#hasProject> <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data16_13TeV> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/campaign/t0pro20> <http://nosql.tpu.ru/ontology/ATLAS#hasProductionStep> <http://nosql.tpu.ru/ontology/ATLAS#recon> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/campaign/t0pro20> <http://nosql.tpu.ru/ontology/ATLAS#hasDataSampleFormat> <http://nosql.tpu.ru/ontology/ATLAS#AOD> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/campaign/repro20> a <http://nosql.tpu.ru/ontology/ATLAS#Campaign> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/campaign/repro20> <http://nosql.tpu.ru/ontology/ATLAS#hasProject> <http://nosql.tpu.ru:8890/DAV/ATLAS/project/data15_13TeV> .
A triple was formed: <http://nosql.tpu.ru:8890/DAV/ATLAS/campaign/repro20> <http://nosql.tpu.ru/ontology/ATLAS#hasProductionStep> <http://nosql.tpu.ru/ontology/ATLAS#recon> .

```

Рисунок 12. Выполнение скрипта

5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

5.1. Анализ конкурентных технических решений

Прямых конкурентов разработка в силу своей специфики не имеет, однако наиболее близкими по характеру и области применения являются NeOn и RDF-Translator (первый и второй конкурент соответственно). Результаты анализа представлены в виде оценочной карты в таблице 2.

Таблица 2. Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Вес критерия			Конкурентоспособность		
		Б _ф	Б _{к1}	Б _{к2}	К _ф	К _{к1}	К _{к2}
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Надежность	0,1	3	4	1	0,3	0,4	0,1
2. Соответствие требованиям потребителей	0,25	5	1	2	1,25	0,25	0,5
3. Потребность в ресурсах памяти	0,15	5	3	4	0,75	0,45	0,6
4. Дополнительные утилиты	0,1	1	4	2	0,1	0,4	0,2
Экономические критерии оценки эффективности							
1. Конкурентоспособность продукта	0,15	3	2	2	0,45	0,3	0,3
2. Цена	0,05	2	4	4	0,1	0,2	0,2
3. Предполагаемый срок эксплуатации	0,15	4	2	1	0,6	0,3	0,15
4. Уровень проникновения на рынок	0,05	1	4	2	0,05	0,2	0,1
Итого	1	24	24	18	3,6	2,5	2,15

Согласно результатам, разрабатываемый модуль в наибольшей степени и по техническим, и по экономическим критериям соответствует требованиям разработчиков базы знаний.

5.2. SWOT-анализ

SWOT-анализ представляет собой комплексный анализ научно-исследовательского проекта. Он проводится в два этапа. По результатам проведения анализа представлена матрица в таблице 3.

Таблица 3. Матрица SWOT

	Сильные стороны научно-исследовательского проекта: С1. Ускорение процесса обработки метаданных С2. Отсутствие прямых аналогов скриптов-конвертеров из-за редкости использования требуемого стандарта С3. Востребованность С4. Объединение знаний в области онтологий и программирования на высокоуровневом языке С5. Разработка на основе реальных данных	Слабые стороны научно-исследовательского проекта: Сл1. Сложность обработки входных данных при изменении их структуры Сл2. Синтаксическая ригидность модуля Сл3. Специфика метаданных научного эксперимента Сл4. Сложность встраивания в общую систему Сл5. Отсутствие контекстных пояснений к некоторым исходным данным
Возможности: В1. Потокная автоматизация процесса импорта метаданных В2. Использование для всех научных экспериментов сообщества В3. Подключение новых источников метаданных В4. Возможность перевода данных в RDF-триплеты В5. Появление дополнительного спроса на продукты с используемым	За счет своей уникальности и специфики области применения, аналоги данного модуля, подстроенные под другую узкоспециализированную область знаний, могут найти применение и в других проектах.	Сложность базы знаний научного эксперимента и ее структуры требует дополнительных знаний в области онтологий для возможности подключения к базе новых источников данных.
Угрозы: У1. Неизвестны будущие источники метаданных У2. Изменение стандартов языка, на котором написаны скрипты У3. Длительность и сложность современных исследований У4. Рост требований к вычислительным	Для дальнейшей перспективы работы с другими текстовыми источниками данных требуется унификация написанных элементов модуля.	Сложность поставленной задачи и работа с разнородными источниками входных данных ставит под угрозу возможность использования разработанного модуля с новыми источниками

мощностям оборудования У5. Многоуровневое согласование на всех этапах работы		данных.
--	--	---------

Результаты, представленные в матрице, получены в ходе составления интерактивных матриц проекта, представленных в таблицах 4-7.

Таблица 4. Интерактивная матрица проекта

Сильные стороны проекта						
Возможности проекта		C1	C2	C3	C4	C5
	B1	+	0	+	-	0
	B2	+	+	+	-	0
	B3	+	0	+	+	+
	B4	0	+	+	+	+
	B5	-	+	+	+	-

Таблица 5. Интерактивная матрица проекта

Слабые стороны проекта						
Возможности проекта		Сл1	Сл2	Сл3	Сл4	Сл5
	B1	+	+	-	+	-
	B2	+	+	+	+	-
	B3	0	+	+	+	+
	B4	-	+	+	-	+
	B5	-	-	+	-	0

Таблица 6. Интерактивная матрица проекта

Сильные стороны проекта						
Угрозы проекта		C1	C2	C3	C4	C5
	У1	+	-	0	-	+
	У2	-	+	0	+	0
	У3	0	-	+	-	+
	У4	+	0	+	-	+
	У5	-	0	+	-	+

Таблица 7. Интерактивная матрица проекта

Слабые стороны проекта						
Угрозы проекта		Сл1	Сл2	Сл3	Сл4	Сл5
	У1	+	+	+	+	+
	У2	+	-	-	+	-
	У3	+	-	+	+	-
	У4	+	+	+	+	+
	У5	-	+	0	+	+

5.3. Определение возможных альтернатив проведения научных исследований

В рамках поставленной задачи возможных вариантов реализации было два: написание скриптов на Python или Java-приложения. Результаты морфологического анализа представлены в таблице 8.

Таблица 8. Морфологическая матрица для модуля импорта метаданных

	1	2
А. Среда разработки	IDLE (Python 3.6 64-bit)	Eclipse
Б. Редактор онтологий	Protégé	
В. Язык программирования	Python	Java
Г. Реализация	Скрипт	Консольное приложение

Как видно из выше приведенной таблицы, сформированы варианты: A1B1B1Г1, A2B1B2Г2

5.3.1. Структура работ в рамках научного исследования

Перечень этапов выполнения проекта указаны в таблице 9.

Таблица 9. Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Разработка технического задания	1	Составление и утверждение технического задания	Руководитель
Выбор направления исследований	2	Подбор материалов по теме	Руководитель
	3	Изучение материалов по теме	Бакалавр
	4	Выбор направления	Руководитель, бакалавр
	5	Календарное планирование работ по теме	Руководитель, бакалавр
Проектирование структуры и разработка ИС	6	Проектирование структуры модуля	Бакалавр
	7	Разработка модуля	Бакалавр
	8	Тестирование модуля	Бакалавр
Обобщение и оценка результатов	9	Оценка эффективности полученных результатов	Руководитель, бакалавр
Оформление отчета по НИР (комплекта документации по	10	Составление пояснительной записки	Бакалавр

ОКР)		(эксплуатационно-технической документации)	
------	--	--	--

5.3.2. Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаев образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования.

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, т.к. зависит от множества трудно учитываемых факторов. Для определения ожидаемого (среднего) значения трудоемкости $t_{ожі}$ используется следующая формула:

$$t_{ожі} = \frac{3t_{\min i} + 2t_{\max i}}{5}, \quad (1)$$

где $t_{ожі}$ – ожидаемая трудоемкость выполнения i -ой работы чел.-дн.;

$t_{\min i}$ – минимально возможная трудоемкость выполнения заданной i -ой работы (оптимистическая оценка: в предположении наиболее благоприятного стечения обстоятельств), чел.-дн.;

$t_{\max i}$ – максимально возможная трудоемкость выполнения заданной i -ой работы (пессимистическая оценка: в предположении наиболее неблагоприятного стечения обстоятельств), чел.-дн.

Для выполнения перечисленных в таблице 9, указанной выше, работ требуются специалисты:

- бакалавр (Б);
- научный руководитель (Р).

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях T_p , учитывающая параллельность выполнения работ несколькими исполнителями. Такое вычисление необходимо для обоснованного расчета заработной платы, так как

удельный вес зарплаты в общей сметной стоимости научных исследований составляет около 65 %.

$$T_{pi} = \frac{t_{ожі}}{Ч_i}, \quad (2)$$

где T_{pi} – продолжительность одной работы, раб. дн.;

$t_{ожі}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн.

$Ч_i$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

5.3.3. Разработка графика проведения научного исследования

Для удобства построения графика, длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Для этого необходимо воспользоваться следующей формулой:

$$T_{ki} = T_{pi} \cdot k_{кал}, \quad (3)$$

где T_{ki} – продолжительность выполнения i -й работы в календарных днях;

T_{pi} – продолжительность выполнения i -й работы в рабочих днях;

$k_{кал}$ – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{кал} = \frac{T_{кал}}{T_{кал} - T_{вых} - T_{пр}}, \quad (4)$$

где $T_{кал}$ – календарные дни ($T_{кал} = 366$);

$T_{вых}$ – выходные дни ($T_{вых} = 92$);

$T_{пр}$ – праздничные дни ($T_{пр} = 26$).

$$T_K = \frac{365}{365 - 92 - 26} = 1,478$$

Временные показатели проведения научного исследования представлены в таблице 10.

Таблица 10. Временные показатели проведения научного исследования

№ работ	Трудоёмкость работ						Исполните ли		Длительность работ в рабочих днях T_{pi}		Длительность работ в календарных днях T_{ki}	
	t_{min} , чел-дни		t_{max} , чел-дни		$t_{ож\bar{c}i}$, чел-дни							
	Испол.1	Испол.2	Испол.1	Испол.2	Испол.1	Испол.2	Испол.1	Испол.2	Испол.1	Испол.2	Испол.1	Испол.2
1	10	10	15	15	12	12	Р	Р	12	12	14	14
2	8	8	12	12	9,6	9,6	Р	Р	9,6	9,6	12	12
3	10	10	20	20	14	14	Б	Б	14	14	20	20
4	1	1	2	2	1,4	1,4	Р,Б	Р,Б	0,7	0,7	3	3
5	1	1	2	2	1,4	1,4	Р,Б	Р,Б	0,7	0,7	1	1
6	5	5	8	8	6,2	6,2	Б	Б	6,2	6,2	9	9
7	15	20	20	30	17	24	Б	Б	17	24	23	32
8	1	1	2	2	1,4	1,4	Б	Б	1,4	1,4	1	1
9	3	3	5	5	3,8	3,8	Р,Б	Р,Б	1,9	1,9	2	2
10	20	20	23	23	21,2	21,2	Б	Б	21,2	21,2	12	12
Ит ого	Всего								84,7	91,7	97	106
	Руководитель								24,9	24,9	32	32
	Бакалавр								63,1	70,1	71	80

На основании таблицы 10 строится календарный план-график. График строится для максимального по длительности исполнения работ в рамках научно-исследовательского. План-график приведен в таблице 11.

Таблица 11. Календарный план-график

№ работ	Вид работ	Исполнители	T_{ki} , кал. дн.	Продолжительность выполнения работ											
				февраль			март			апрель			май		
				1	2	3	1	2	3	1	2	3	1	2	3
1	Составление и утверждение технического задания.	Руководитель	14	■	■										
2	Подбор материалов по теме	Руководитель	12		■	■									
3	Изучение материалов по теме	Бакалавр	20				▨	▨	▨						
4	Выбор направления	Руководитель, бакалавр	3					▨							
5	Календарное планирование работ по теме	Руководитель, бакалавр	1					▨							
6	Проектирование структуры ИС	Бакалавр	9						▨						
7	Разработка ИС	Бакалавр	23							▨	▨	▨			
8	Тестирование ИС	Бакалавр	1									▨			
9	Оценка эффективности и полученных результатов	Руководитель, бакалавр	2									▨			
10	Составление пояснительной записки	Бакалавр	12									▨	▨	▨	



- руководитель



- бакалавр

5.3.4. Бюджет научно-технического исследования (НТИ)

При планировании бюджета НТИ должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением. В процессе формирования бюджета НТИ используется следующая группировка затрат по статьям:

- материальные затраты НТИ;
- затраты на специальное оборудование для научных (экспериментальных) работ;
- основная заработная плата исполнителей темы;
- дополнительная заработная плата исполнителей темы;
- отчисления во внебюджетные фонды (страховые отчисления);
- затраты научные и производственные командировки;
- контрагентные расходы;
- накладные расходы.

5.3.4.1. Расчет материальных затрат НТИ

Произведем расчет всех материалов, используемых при разработке проекта:

- приобретаемые со стороны сырье и материалы, необходимые для создания научно-технической продукции;
- покупные материалы, используемые в процессе создания научно-технической продукции для обеспечения нормального технологического процесса и для упаковки продукции или расходуемых на другие производственные и хозяйственные нужды (проведение испытаний, контроль, содержание, ремонт и эксплуатация оборудования, зданий, сооружений, других основных средств и прочее), а также запасные части для ремонта оборудования, износа инструментов, приспособлений, инвентаря, приборов, лабораторного оборудования и других средств труда, не относимых к основным средствам, износ спецодежды и других малоценных и быстроизнашивающихся предметов;

- покупные комплектующие изделия и полуфабрикаты, подвергающиеся в дальнейшем монтажу или дополнительной обработке;

- сырье и материалы, покупные комплектующие изделия и полуфабрикаты, используемые в качестве объектов исследований (испытаний) и для эксплуатации, технического обслуживания и ремонта изделий – объектов испытаний (исследований);

В материальные затраты, помимо вышеуказанных, включаются дополнительно затраты на канцелярские принадлежности, диски, картриджи и т.п. Однако их учет ведется в данной статье только в том случае, если в научной организации их не включают в расходы на использование оборудования или накладные расходы. В первом случае на них определяются соответствующие нормы расхода от установленной базы. Во втором случае их величина учитывается как некая доля в коэффициенте накладных расходов.

Расчет материальных затрат осуществляется по следующей формуле:

$$З_m = (1 + k_T) \cdot \sum_{i=1}^m Ц_i \cdot N_{расxi} , \quad (5)$$

где m – количество видов материальных ресурсов, потребляемых при выполнении научного исследования;

$N_{расxi}$ – количество материальных ресурсов i -го вида, планируемых к использованию при выполнении научного исследования (шт., кг, м, м² и т.д.);

$Ц_i$ – цена приобретения единицы i -го вида потребляемых материальных ресурсов (руб./шт., руб./кг, руб./м, руб./м² и т.д.);

k_T – коэффициент, учитывающий транспортно-заготовительные расходы.

Материальные затраты представлены в таблице 12.

Таблица 12 Материальные затраты

Наименование	Единица измерения	Количество		Цена за ед., руб.		Затраты на материалы, (З _м), руб.	
		Исп.1	Исп.2	Исп.1	Исп.2	Исп.1	Исп.2
Офисные принадлежности							
Бумага для принтера А4	уп	1	1	200	200	200	200
Картридж для принтера	шт	1	1	600	600	600	600
Электроэнергия							
Электроэнергия	кВт	370	384	4	4	1480	1536
Итого						2280	2336

5.3.4.2. Расчет затрат на специальное оборудование для научных (экспериментальных) работ

Дополнительное оборудование для научных (экспериментальных) работ не приобреталось.

5.3.4.3. Основная заработная плата исполнителей темы

Рассчитаем основную заработную плату работников, непосредственно занятых выполнением НИИ, (включая премии, доплаты) и дополнительную заработную плату:

$$З_{зп} = З_{осн} + З_{доп}, \quad (6)$$

где $З_{осн}$ – основная заработная плата;

$З_{доп}$ – дополнительная заработная плата (12-20 % от $З_{осн}$).

Основная заработная плата ($З_{осн}$) руководителя (лаборанта, инженера) от предприятия (при наличии руководителя от предприятия) рассчитывается по следующей формуле:

$$З_{осн} = З_{дн} \cdot T_p, \quad (7)$$

где $З_{осн}$ – основная заработная плата одного работника;

T_p – продолжительность работ, выполняемых научно-техническим работником, раб. дн. (табл. 7);

$З_{дн}$ – среднедневная заработная плата работника, руб.

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{\text{дн}} = \frac{Z_{\text{м}} \cdot M}{F_{\text{д}}}, \quad (8)$$

где $Z_{\text{м}}$ – месячный должностной оклад работника, руб.;

M – количество месяцев работы без отпуска в течение года:

при отпуске в 24 раб. дня $M = 11,2$ месяца, 5-дневная неделя;

$F_{\text{д}}$ – действительный годовой фонд рабочего времени научно-технического персонала, показанный в таблице 13.

Таблица 13. Баланс рабочего времени

Показатели рабочего времени	Руководитель	Студент
Календарное число дней	365	365
Количество нерабочих дней	107	107
- выходные дни		
- праздничные дни		
Потери рабочего времени	24	24
- отпуск		
- невыходы по болезни		
Действительный годовой фонд рабочего времени	234	234

Месячный должностной оклад работника:

$$Z_{\text{м}} = Z_{\text{тс}} \cdot (1 + k_{\text{пр}} + k_{\text{д}}) \cdot k_{\text{р}}, \quad (9)$$

где $Z_{\text{тс}}$ – заработная плата по тарифной ставке, руб.;

$k_{\text{р}}$ – районный коэффициент, равный 1,3 (для Томска).

Расчёт основной заработной платы приведён в таблице 14.

Таблица 14. Расчёт основной заработной платы

Исп.	Исполнители	Разряд	З _{тс} , руб.	k _{пр}	k _д	k _р	З _м , руб.	З _{дн} , руб.	T _р , раб. дн.	З _{осн} , руб.
Исполнение 1	Руководитель	Ведущий программист	20800	-	-	1,3	27040	1294,2	24,9	32225,58
	Бакалавр	-	2206,75	-	-	-	2206,75	106,1	63,1	6694,91
	Итого									38920,49
Исполнение 2	Руководитель	Ведущий программист	20800	-	-	1,3	27040	1294,2	24,9	32225,58
	Бакалавр	-	2206,75	-	-	-	2206,75	106,1	70,1	7437,61
	Итого									39663,19

5.3.4.4. Дополнительная заработная плата исполнителей темы

Расчет дополнительной заработной платы ведется по следующей формуле:

$$Z_{\text{доп}} = k_{\text{доп}} \cdot Z_{\text{осн}}, \quad (10)$$

где $k_{\text{доп}}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12 – 0,15). Примем коэффициент равный 0, 12. Расчет дополнительной заработной платы указан в таблице 15.

Таблица 15. Расчёт дополнительной заработной платы

Исполнитель	Основная заработная плата, руб.		k _{доп}	Дополнительная заработная плата, руб.	
	Испол.1	Испол.2		Испол.1	Испол.2
Руководитель	32225,58	32225,58	0,12	3867,06	3867,06
Итого				3867,06	3867,06

5.3.4.5. Отчисления во внебюджетные фонды (страховые отчисления)

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$З_{внеб} = k_{внеб} \cdot (З_{осн} + З_{доп}), \quad (11)$$

где $k_{внеб}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

На 2017 г. В соответствии с Федеральным законом от 24.07.2009 №212-ФЗ установлен размер страховых взносов равный 30%. На основании пункта 1 ст.58 закона №212-ФЗ для учреждений осуществляющих образовательную и научную деятельность в 2014 году водится пониженная ставка – 27,1%.

Отчисления во внебюджетные фонды представлены в таблице 16.

Таблица 16. Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.		Полная заработная плата, руб.	
	Испол.1	Испол.2	Испол.1	Испол.2
Руководитель	32225,58	32225,58	36092,64	36092,64
Коэффициент отчислений во внебюджетные фонды	0,271			
Итого				
Исполнение 1	9781,1			
Исполнение 2	9781,1			

5.3.4.6. Расчет затрат на научные и производственные командировки

На данном этапе в научных и производственных командировках нет необходимости.

5.3.4.7. Контрагентные расходы

В качестве контрагентных расходов учитывается услуга по предоставлению исходных данных – сторонними участниками коллаборации, не входящими в кадровый состав ТПУ. Приблизительная оценка – 5000 рублей.

5.3.4.8. Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы, размножение материалов и т.д. Их величина определяется по следующей формуле:

$$З_{\text{накл}} = (\text{сумма статей } 1 \div 7) \cdot k_{\text{нр}}, \quad (12)$$

где $k_{\text{нр}}$ – коэффициент, учитывающий накладные расходы.

Величину коэффициента накладных расходов можно взять в размере 16%.

5.3.4.9. Формирование бюджета затрат научно-исследовательского проекта

Определение бюджета затрат на научно-исследовательский проект по каждому варианту исполнения приведен в таблице 17.

Таблица 17. Расчет бюджета затрат НТИ

Наименование статьи	Сумма, руб.		Примечание
	Испол.1	Испол.2	
1. Материальные затраты НТИ	2280	2336	Пункт 5.4.3.1.
2. Затраты на специальное оборудование для научных (экспериментальных) работ	-	-	Пункт 5.4.3.2.
3. Затраты по основной заработной плате исполнителей темы	38920,49	39663,19	Пункт 5.4.3.3.
4. Затраты по дополнительной заработной плате исполнителей темы	3867,06	3867,06	Пункт 5.4.3.4.
5. Отчисления во внебюджетные фонды	9781,1	9781,1	Пункт 5.4.3.5.
6. Затраты на научные и производственные командировки	0	0	Пункт 5.4.3.6.
7. Контрагентские расходы	5000	5000	Пункт 5.4.3.7.
8. Накладные расходы	9575,78	9703,58	16 % от суммы ст. 1-7
9. Бюджет затрат НТИ	69424,43	70350,93	Сумма ст. 1- 8

5.3.5. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Интегральный показатель финансовой эффективности научного исследования получают в ходе оценки бюджета затрат трех (или более) вариантов исполнения научного исследования. Для этого наибольший интегральный показатель реализации технической задачи принимается за базу расчета (как знаменатель), с которым соотносятся финансовые значения по всем вариантам исполнения.

Интегральный финансовый показатель разработки определяется как:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}}, \quad (13)$$

где $I_{\text{финр}}^{\text{исп.}i}$ – интегральный финансовый показатель разработки;

Φ_{pi} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта (в т.ч. аналоги).

Полученная величина интегрального финансового показателя разработки отражает соответствующее численное увеличение бюджета затрат разработки в размах (значение больше единицы), либо соответствующее численное удешевление стоимости разработки в размах (значение меньше единицы, но больше нуля).

Максимальная стоимость составляет 70350,93рублей, следовательно:

$$I_{\text{финр}}^{\text{исп.}1} = \frac{\Phi_{p1}}{\Phi_{\text{max}}} = 0,987$$

$$I_{\text{финр}}^{\text{исп.}2} = \frac{\Phi_{p2}}{\Phi_{\text{max}}} = 1$$

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum a_i \cdot b_i, \quad (14)$$

где I_{pi} – интегральный показатель ресурсоэффективности для i -го варианта исполнения разработки;

a_i – весовой коэффициент i -го варианта исполнения разработки;

b_i^a, b_i^p – бальная оценка i -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания;

n – число параметров сравнения.

Таблица 18. Сравнительная оценка характеристик вариантов исполнения проекта

Критерии \ Объект исследования	Весовой коэффициент параметра	Испол.1	Испол.2
1. Удобство внедрения	0,25	5	4
2. Функциональная мощность	0,20	4	3
3. Удобство в эксплуатации	0,25	5	4
4. Потребность в ресурсах памяти	0,15	5	3
5. Надежность	0,15	4	4
ИТОГО	1		

$$I_{p-исп1} = 5*0,25+4*0,20+5*0,25+5*0,15+4*0,15 = 4,65;$$

$$I_{p-исп2} = 4*0,25+3*0,20+4*0,25+3*0,15+4*0,15 = 3,65;$$

Интегральный показатель эффективности вариантов исполнения разработки ($I_{исп.}$) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{исп.1} = \frac{I_{p-исп1}}{I_{финр.1}}, \quad I_{исп.2} = \frac{I_{p-исп2}}{I_{финр.2}} \text{ и т.д.} \quad (15)$$

$$I_{исп.1} = \frac{4,65}{0,987} = 4,71 ; \quad I_{исп.2} = \frac{3,65}{1} = 3,65 ;$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных. Сравнительная эффективность проекта ($\mathcal{E}_{cp}$):

$$\mathcal{E}_{cp} = \frac{I_{исп.1}}{I_{исп.2}} \quad (16)$$

$$\mathcal{E}_{cp1} = \frac{4,71}{3,65} = 1,28 ; \quad \mathcal{E}_{cp2} = \frac{3,65}{4,71} = 0,77 ;$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта. Сравнительная эффективность разработки представлена в таблице 19.

Таблица 19. Сравнительная эффективность разработки

№ п/п	Показатели	Испол.1	Испол.2
1	Интегральный финансовый показатель исполнения	0,987	1
2	Интегральный показатель ресурсоэффективности исполнения	4,65	3,65
3	Интегральный показатель эффективности вариантов исполнения	4,71	3,65
4	Сравнительная эффективность вариантов исполнения	1,28	0,77

Общий вывод по разделу:

В результате работы по разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» выявили и сравнили два варианта исполнения научно-исследовательской работы. Бюджет затрат первого варианта исполнения равен 69424,43рублей, второго 70350,93. Произвели сравнительную оценку эффективности разработки и, исходя из полученных результатов, можно сделать вывод, что наиболее эффективным вариантом решения поставленной в бакалаврской работе технической задачи с позиции финансовой и ресурсной эффективности является 1 вариант исполнения – написание скриптов на языке Python.

6. Социальная ответственность

Объектом исследования выступает рабочее место программиста, разрабатывающего модуль импорта метаданных, пользователи которого могут использовать его с помощью своих персональных компьютеров или других устройств, на которых обеспечен доступ к общей информационной системе, для которой разрабатывается модуль. Модуль выполняет функцию импорта метаданных о научных экспериментах с последующим приведением данных в вид, пригодный для загрузки в общую информационную модель, для последующего решения поставленных пользователями информационной системы задач.

6.1. Производственная безопасность

Опасные и вредные факторы при разработке и эксплуатации проектируемого решения. Факторы по ГОСТ 12.0.003-74 «Опасные и вредные производственные факторы. Классификация» [6] представлена в таблице 20.

Таблица 20. Основные элементы производственного процесса, формирующие опасные и вредные факторы

Наименование видов работ и параметров производственного процесса	Факторы (ГОСТ 12.0.003-74 ССБТ) [1]		Нормативные документы
	Вредные	Опасные	
Работа с ПК, устройствами ввода и вывода информации	Несоответствие параметрам микроклимата	Опасность поражения электрическим током	ГОСТ 12.1.005-88 [7], СанПиН 2.2.2/ 2.4.1340-03 [8], ГОСТ Р 50571. 17-2000 [9]
	Недостаточная освещенность рабочего места		СанПиН 2.2.1/2.1.1.1278-03 [10]
	Повышенный уровень шума		СанПиН 2.2.4.3359-16 [11]
	Умственное перенапряжение		ТОИ Р-45-084-01 [12]

6.1.1. Параметры микроклимата

Оптимальное состояние воздушной среды должно обеспечивать ощущение теплового комфорта в течение 8-часового рабочего дня, не вызывать отклонений в состоянии здоровья. Энергетические затраты организма измеряются в ккал/ч (Вт) и по затраченной энергии работы разделяются на категории. Так работа программиста относится к категории Ia – интенсивность энергозатрат до 120 ккал/ч (до 139 Вт) [8].

Работы производятся в основном сидя и сопровождаются незначительным физическим напряжением. Допустимые параметры микроклимата на рабочем месте для категории Ia приведены в таблице 21.

Таблица 21. Допустимые величины показателей воздушной среды на рабочих местах производственных помещений по СанПиН 2.2.2/2.4.1340-03

Сезон года	Категория тяжести выполняемых работ	Температура, °С		Относительная влажность, %		Скорость движения воздуха, м/сек	
		Факт. значение	Допустм. значение	Факт. значение	Допуст. значение	Фактич. значение	Допуст. значение
1	2	3	4	5	6	7	8
Холодный	Ia	22-24	20-25	60	15-75	0,1	0,1
Теплый	Ia	23-25	21-28	60	15-75	0,1	0,2

Температура воздуха в рабочем помещении в холодное время года поддерживается в диапазоне (согласно измерениям термометром) от 21 до 23°C, в теплое – от 23 до 25°C. Влажность в соответствии с нормами (согласно измерениям гигрометром) колеблется около 60%. Для поддержания соответствующих микроклиматических параметров используются системы отопления и вентиляции, а также проводится кондиционирование воздуха в помещении.

6.1.2. Освещенность рабочего места

Для обеспечения нормативных условий работы необходимо провести оценку освещенности рабочей зоны в соответствии с СанПиН 2.2.1/2.1.1.1278-03.

Правильное освещение рабочих мест и помещений является важным условием для создания безопасных и благоприятных условий труда.

Все поле зрения должно быть освещено равномерно – это является основным гигиеническим требованием. Другими словами, уровень естественного освещения рабочего места и яркость дисплея компьютера должны быть приблизительно одинаковыми, т.к. яркий свет в зоне периферийного зрения заметно увеличивает глазное напряжение, что приводит к их быстрой утомляемости.

Для снижения отраженной блескости наряду с перечисленными выше рекомендуются следующие мероприятия:

Для внутренней отделки интерьера помещений с компьютерами должны использоваться диффузно отражающие материалы с коэффициентом отражения для потолка 0,7 – 0,8, для стен 0,5 – 0,6, для пола – 0,3 - 0,5.

Дизайн ПЭВМ должен предусматривать окраску корпуса в спокойные мягкие тона с диффузным рассеиванием света. Корпус ПЭВМ, клавиатура и другие блоки и устройства ПЭВМ должны иметь матовую поверхность с коэффициентом отражения 0,4 – 0,6 и не иметь блестящих деталей, способных создавать блики.

Для освещения помещений с ПЭВМ рекомендуется применять светильники с зеркальными параболическими решетками. Применение светильников без рассеивателей или экранирующих решеток нежелательно.

Персоналу, эксплуатирующему компьютеры с черными экранами, не рекомендуется использование светлой или блестящей одежды.

Фактические и требуемые параметры систем естественного и искусственного освещения показаны в таблице 22.

Таблица 22. Параметры систем естественного и искусственного освещения на рабочих местах по СанПиН

2.2.1/2.1.1.1278-03

Наименование рабочего места	Рабочая поверхность, м	Коэффициент естественной освещенности КЕО, %		Коэффициент совмещенного освещения, КЕО, %		Искусственное освещение				
		при верхнем или комбинированном освещении	при боковом освещении	при верхнем или комбинированном освещении	при боковом освещении	Освещенность, лк		Показатель диска форта, М, не более	Коэффициент пульсации освещенности, %, не более	
						при комбинированном освещении	при общем освещении			
						Всего	от общего			
Кабинет информатики	Г – 0,8; Экран дисплея: В – 1	3,5 -	1,2 -	2,1 -	0,7 -	500 -	300 -	300 200	15 -	10 -
Аудитории, учебные кабинеты	Г – 0,8;	3,5	1,2	2,1	0,7	-	-	300	40	10

Для кабинета информатики и учебных кабинетов норматив по СанПиН 2.2.1/2.1.1.1278-03 (искусственное общее освещение) минимальные значения освещённости - 300 люкс.

Равномерное освещение горизонтальной рабочей поверхности достигается при определенных отношениях расстояния между центрами светильников L (м) ($L = 1,75 \cdot H$), к высоте их подвеса над рабочей поверхностью H (м). В помещении для разработки исследования высота составляет 2,5.

Количество люминесцентных ламп ЛБ 40 в количестве 24 шт. Согласно ГОСТ 6825-91. "Лампы люминесцентные трубчатые для общего освещения" номинальный световой поток люминесцентные лампы ЛБ 40 составляет 3000лк.

$$L = 1.75 \cdot 2.5 = 4.375(\text{м})$$

$$\Phi = \text{НСП} \cdot N = 3000 \cdot 24 = 72000 \text{ (люмен)}$$

НСП - номинальный световой поток лампы, N - количество ламп

$$I = \Phi / 2\pi = 72000 / 6.28 = 11465 \text{ (кандел)}$$

$$E = I / L \cdot L = 11465 / 4.375 \cdot 4.375 = 599 \text{ (люкс)} \text{ (без учёта потерь в плафоне светильника)}$$

6.1.3. Уровень шума

Шум ухудшает условия труда, оказывают вредное воздействие на организм человека. Действие шума различно: он затрудняет разборчивость речи, вызывает снижение работоспособности, повышает утомляемость, вызывает необратимые изменения в органах слуха человека, ослабляет внимание, ухудшает память, снижает реакцию, увеличивает число ошибок при работе.

Источником шума в учебных помещениях, эксплуатирующих вычислительную технику, являются сами вычислительные машины (встроенные в стойки ЭВМ вентиляторы, принтеры и т.д.), центральная система вентиляции и кондиционирования воздуха и другое оборудование.

На рабочем месте программиста исследователя предельно допустимый уровень звука приведен таблице 23.

Таблица 23. Предельно допустимый уровень звука по СанПиН 2.2.4.3359-16

Рабочие места	Уровень звука, дБА
Учебные кабинеты, аудитории	80
Лаборатории при учебных кабинетах	80

При выполнении основной работы уровень звука не должен превышать 80 дБА. По субъективным ощущениям шумовая обстановка на рабочем месте исследователя соответствует норме.

Снизить уровень шума в помещениях можно использованием звукопоглощающих материалов с максимальными коэффициентами звукопоглощения в области частот 63-8000 Гц для отделки стен и потолка помещений. Дополнительный звукопоглощающий эффект создают однотонные занавески из плотной ткани, повешенные в складку на расстоянии 15-20 см от ограждения. Ширина занавески должна быть в 2 раза больше ширины окна.

6.1.4. Умственное перенапряжение

Для видов трудовой деятельности устанавливается 3 категории тяжести и напряженности работы с компьютером, которые определяются: для группы А - по суммарному числу считываемых знаков за рабочую смену (не более 60000 знаков за смену); для группы Б - по суммарному числу считываемых или вводимых знаков за рабочую смену (не более 40000 знаков за смену); для группы В - по суммарному времени непосредственной работы с компьютером за рабочую смену (не более 6 часов за смену).

Таблица 24. Категория работы по тяжести и напряженности по ТОО Р 45-084-01

Категория работы по тяжести и напряженности	Уровень нагрузки за рабочую смену при видах работы на ПК		
	Группа А Количество знаков	Группа Б Количество знаков	Группа В Время работы, ч
III	До 60000	До 40000	До 6,0

При 8-часовой рабочей смене и работе на ПК регламентированные перерывы следует устанавливать:

Для третьей категории работ — через 1,5- 2,0 часа от начала рабочей смены и через 1,5-2,0 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

6.1.5. Опасность поражения электрическим током

Согласно СанПиН 2.2.2/2.4.1340-03. «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы», рабочее место должно находиться в безопасной зоне, которое не характеризуется наличием таких условий, как повышенная влажность (относительная влажность воздуха длительно превышает 75%), высокая температура (более 35°C), токопроводящая пыль, токопроводящие полы, возможность одновременного соприкосновения к имеющим соединения с землей металлическим элементам и металлическим корпусам электрооборудования.

Электрические установки, к которым относится ПК, представляют для человека большую потенциальную опасность, так как в процессе эксплуатации или проведения профилактических работ человек может коснуться комплектующих компьютера, находящихся под напряжением.

Специфическая опасность — корпуса ПК и прочего оборудования, оказавшегося под напряжением в результате повреждения или пробоя изоляции, не подают каких-либо сигналов, которые предупреждают человека

об опасности. Причинами электропоражений являются: провода с поврежденной изоляцией, розетки сети без предохранительных кожухов.

Для защиты от поражения электрическим током все токоведущие части должны быть защищены от случайных прикосновений кожухами, корпус устройства должен быть заземлен. Заземление выполняется изолированным медным проводом сечением 1.5 мм, который присоединяется к общей шине заземления с общим сечением 48 мм при помощи сварки. Общая шина присоединяется к заземлению, сопротивление которого не должно превышать 4 Ом.

Согласно ГОСТ Р 50571.17-2000. «Требования по обеспечению безопасности электроустановок в зданиях. Выбор мер защиты в зависимости от внешних условий. Защита от пожара», питание устройства в помещении, в котором выполнялась работа, осуществляется от силового щита через автоматический предохранитель, который срабатывает при коротком замыкании нагрузки. Для снижения величин возникающих разрядов применяются покрытия из антистатического материала.

Программист исследователь работает с электроприборами: компьютером (монитор, системный блок, компьютерная мышь и клавиатура).

В данном случае существует опасность электропоражения:

- при непосредственном прикосновении к токоведущим частям во время ремонта ПК;
- при прикосновении к нетоковедущим частям, оказавшимся под напряжением (в случае нарушения изоляции токоведущих частей ПК);
- при соприкосновении с полом, стенами, оказавшимися под напряжением;
- имеется опасность короткого замыкания в высоковольтных блоках: блоке питания и блоке дисплейной развёртки.

Рабочие место программиста исследователя оборудовано таким образом, чтобы исключить взаимное соприкосновение кабелей и шнуров питания соседних компьютеров.

К организационно-техническим мероприятиям относится первичный инструктаж по технике безопасности. Первичный инструктаж по технике безопасности является обязательным условием для допуска к работе в данном помещении.

6.2. Экологическая безопасность

Разрабатываемый проект не имеет влияния на окружающую среду, так как само решение разрабатывается и используется внутри персональных компьютеров, которые могут стать источниками различных загрязнений.

Защита литосферы. Согласно ГОСТ Р 56397-2015 «Техническая экспертиза работоспособности радиоэлектронной аппаратуры, оборудования информационных технологий, электрических машин и приборов. Общие требования» пункт 5.8.1, после проведения технической экспертизы если оборудование не ремонтпригодно, то оно признается неработоспособным и рекомендуется к списанию (замене); в случае деградационного отказа оборудования и нецелесообразности его ремонта и модернизации даются рекомендации о необходимости его списания и утилизации. [12]

Согласно «Методики проведения работ по комплексной утилизации вторичных драгоценных металлов из отработанных средств вычислительной техники», утвержденной Государственным Комитетом РФ по телекоммуникациям от 19 октября 1999г. В п.3.1.3. «Технология разборки универсальных ЭВМ» расписаны 4 этапа разборки и подготовки к утилизации внутренних частей ПК.

В результате выполнения этапов формируется партия сырья, включающая сортировку электронного лома по типу, проведение расчета количества ячеек, соединителей, серебросодержащих кабельных изделий, ячеек и типовых элементов замены, содержащие драгоценные металлы, а также партии черных и цветных металлов и сплавов (медь, сталь, никель, латунь, бронза, алюминий, дюралюминий, свинцово - оловянные припои) направляются на переработку на заводы ВДМ, полупроводниковые приборы

(диоды, транзисторы), микросхемы в металлических и металлокерамических корпусах, а также конденсаторы в металлических корпусах демонтируются с плат и сортируются по типу, интегральные микросхемы в пластмассовых корпусах (серии 155, 551 и пр.) демонтируются и собираются отдельно, керамические конденсаторы типа КМ и резисторы после демонтажа также собираются отдельно.

На рабочем месте программиста используются 16 люминесцентных ламп ЛБ40, Согласно ГОСТ 12.3.031-83 «Работы со ртутью. Требования безопасности» п.2.1. все ртутьсодержащие отходы и вышедшие из строя приборы, содержащих ртуть, подлежат сбору и возврату для последующей регенерации ртути в специализированных организациях. В п.2.2. К работе по замене и сбору отработанных ртутьсодержащих ламп допускаются только электромонтеры. Главным условием при замене и сборе отработанных ртутьсодержащих ламп является сохранение герметичности. В п.2.13. Факт сдачи ртутьсодержащих отходов подтверждается возвращением паспорта на вывоз отходов с отметкой о приеме представителя специализированного предприятия [13].

6.3. Безопасность в чрезвычайных ситуациях

К наиболее вероятным ЧС можно отнести следующие: пожар (взрыв) в здании, авария на коммунальных системах жизнеобеспечения, землетрясение. Источниками возгорания может стать электропроводка, внутренние работающие устройства ПК, взрывоопасные предметы в помещении исследователя согласно ГОСТ 12.1.044-89 «Система стандартов безопасности труда. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения» [14].

Превентивными мерами по предупреждению ЧС могут служить системы звукового и визуального оповещения персонала лаборатории и кабинетов об опасности, обучение персонала, методам работы с компьютером, наличие средств пожаротушения и информационных досок с планами эвакуации.

В случае угрозы возникновения ЧС необходимо отключить электропитание, вызвать по телефону пожарную команду, эвакуировать людей из помещения согласно плану эвакуации. При наличии небольшого очага пламени можно воспользоваться подручными средствами с целью прекращения доступа воздуха к объекту возгорания. В качестве подручных средств можно использовать углекислотные огнетушители ОУ-5 высокого давления с зарядом жидкой двуокиси углерода (по ГОСТ 8050-85), расположение которых можно найти на плане эвакуации людей при пожаре и других ЧС из помещений Института кибернетики ТПУ показанном на рисунке 13.

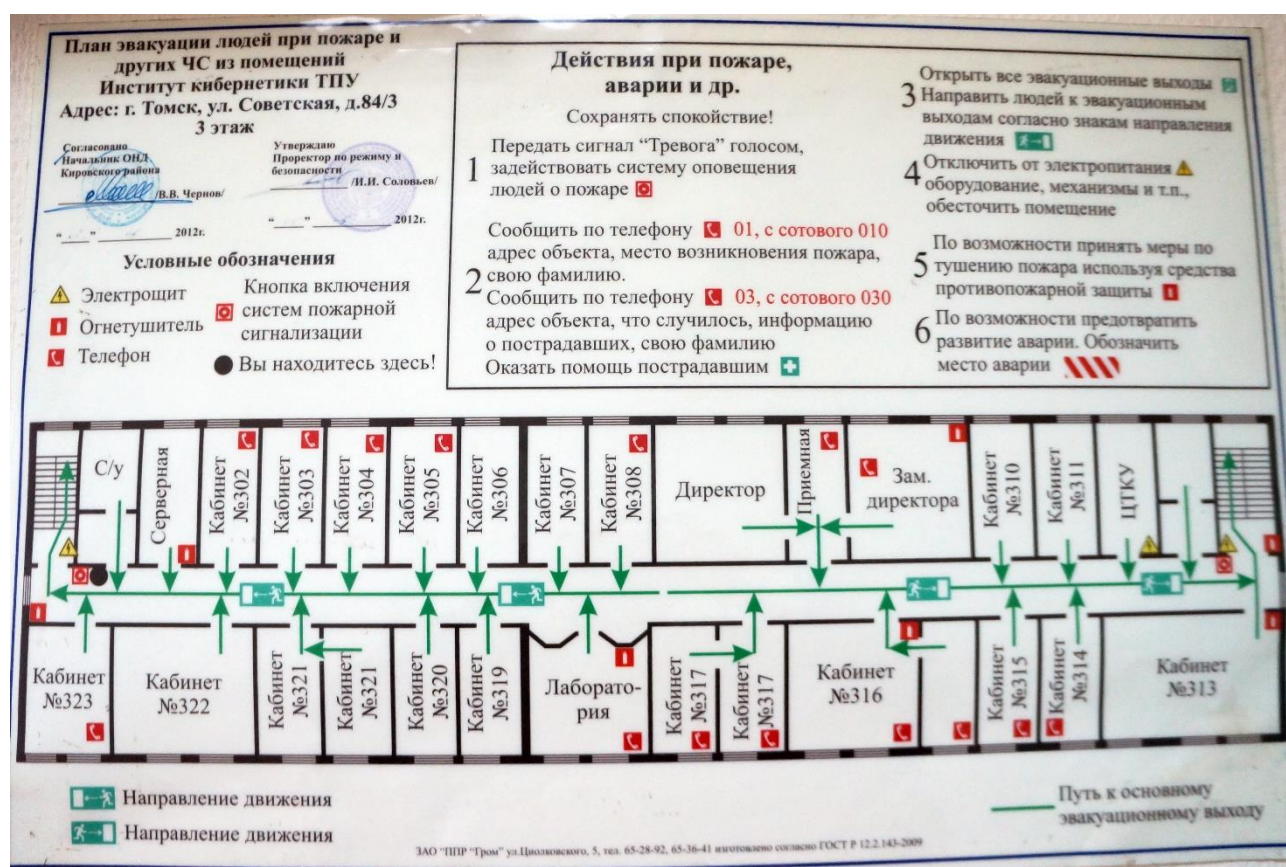


Рисунок 13. План эвакуации при пожаре и других ЧС

6.4. Правовые и организационные вопросы обеспечения безопасности

Рабочие места должны соответствовать требованиям ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические

требования» [15] и ГОСТ 12.2.061-81 «ССБТ. Оборудование производственное. Общие требования безопасности к рабочим местам» [16].

Работа с применением персональных компьютеров сопряжена со значительными зрительными и нервно-психологическими нагрузками, что повышает требования к организации труда пользователей ПК. Конструкция рабочей мебели должна обеспечивать возможность индивидуальной регулировки, соответственно росту работающего, и создавать удобную позу. Часто используемые предметы труда и органы управления должны находиться в оптимальной рабочей зоне. Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количественных и конструктивных особенностей, а также характера выполняемой работы. Высота рабочей поверхности стола должна регулироваться в пределах 680-800 мм, при отсутствии такой возможности его высота должна быть не менее 725 мм.

На поверхности рабочего стола для документов необходимо предусматривать размещение специальной подставки, расстояние которой от глаз должно быть аналогичным расстоянию от глаз до клавиатуры. Модульными размерами рабочей поверхности стола, на основании которых должны рассчитываться конструктивные размеры, следует считать: ширину 800, 1000, 1200 и 1400 мм, глубину 800 и 1000 мм при нерегулируемой его высоте, равной 725 мм. Под столешницей рабочего стола должно быть свободное пространство для ног с размерами по высоте не менее 600 мм, по ширине 500 мм, по глубине 650 мм.

Конструкция рабочего стула должна обеспечивать поддержание рациональной рабочей позы при работе, что позволит изменять позу для снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления.

Заключение

В результате выполнения работы был разработан прототип модуля импорта метаданных в базу знаний научного эксперимента.

Стандарты W3C, относящиеся к Semantic Web, на протяжении лет не находили отклика у широкой аудитории, однако являются эффективным набором инструментов в решении задач по созданию баз знаний и агрегированию гетерогенных данных в единую систему, в связи с чем и возникла необходимость создания модуля импорта, чья логика работы строго зависит от установленной структуры входных данных и онтологической модели данных.

На данный момент сама база знаний также находится на стадии прототипа, что предполагает изменения, как в самой структуре модели данных, так и в структуре файлов, пока не будет выбран оптимальный вариант. Следовательно, предполагается доработка модуля в соответствии с внесенными изменениями в структуру самой системы.

Несмотря на наличие пользовательского текстового интерфейса на данном этапе, в дальнейшем планируется его включение в общую систему с возможностью запуска из Apache Kafka, платформой, работающей с потоками данных.

Ввод в эксплуатацию планируется в конце 2017 года.

Список использованных источников

1. Shadbolt N., Hall W., Berners-Lee T., "The Semantic Web Revisited", IEEE Intelligent Systems Journal, May/June 2006, 96-101 pp.
2. RDF Translator [Электронный ресурс] (дата обращения: 20.05.2017)
URL: <http://rdf-translator.appspot.com/>
3. Easy RDF – RDF Library for PHP [Электронный ресурс] (дата обращения 20.05.2017) URL: <http://www.easyrdf.org/>
4. Heath T., Bizer Ch. – Linked Data. Evolving the Web into a Global Data Space. 2011
5. Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с
6. ГОСТ 12.1.005-88 Система стандартов безопасности труда (ССБТ). Общие санитарно-гигиенические требования к воздуху рабочей зоны (с Изменением N 1).
7. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.
8. ГОСТ Р 50571.17-2000. Электроустановки зданий. Часть 4. Требования по обеспечению безопасности. Глава 48. Выбор мер защиты в зависимости от внешних условий. Раздел 482. Защита от пожара.
9. СанПиН 2.2.1/2.1.1.1278-03. Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий.
10. СанПиН 2.2.4.3359-16 . Санитарно-эпидемиологические требования к физическим факторам на рабочих местах.
11. ТОИ Р-45-084-01. Типовая инструкция по охране труда при работе на персональном компьютере.
12. ГОСТ Р 56397-2015. Техническая экспертиза работоспособности радиоэлектронной аппаратуры, оборудования информационных технологий, электрических машин и приборов. Общие требования.

13. ГОСТ 12.3.031-83. Система стандартов безопасности труда. Работы со ртутью. Требования безопасности.
14. ГОСТ 12.1.044-89. Система стандартов безопасности труда. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методы их определения.
15. ГОСТ 12.2.032-78. Система стандартов безопасности труда. Рабочее место при выполнении работ сидя. Общие эргономические требования.
16. ГОСТ 12.2.061-81. Система стандартов безопасности труда. Оборудование производственное. Общие требования безопасности к рабочим местам.

Приложение А (Листинг программы dataPeriods2TTL.py)

```
#choose graph
print("Current graph: " + graph + "\n")
graph_answer = input("Would you like to choose another one? [Y/N] ")
if graph_answer.lower() in ['y', 'yes']:
    graph = input("Please, insert a graph: ")
print("\nCurrent graph: " + graph + "\n")

#choose ontology
print("Current ontology: " + ontology + "\n")
ontology_answer = input("Would you like to choose another one? [Y/N] ")
if ontology_answer.lower() in ['y', 'yes']:
    ontology = input("Please, insert an ontology: ")
print("\nCurrent ontology: " + ontology + "\n")

#path
chosen_path = os.path.normpath(askopenfilename())

#year
#year = '20'+ chosen_path.rstrip('.json')[len(chosen_path.rstrip('.json'))-
2:len(chosen_path.rstrip('.json'))]

#a ttl document with default name
output_data = open("data_periods.ttl", 'w')

#input
with open(chosen_path) as data_file:
    input_data = json.load(data_file)
```

```

listProj = []
for i, item in enumerate(input_data):
    year = "20" + str(re.findall(r'\d+', input_data[i]['projectName'])[0])
    dataPeriod = "<%s/dataperiod/%s_%s_%s>" %
(graph,year,input_data[i]['period'],input_data[i]['periodLevel'])
    #project name
    projectName = input_data[i]['projectName']
    #a subject in ontology
    project = "<%s/project/%s>" % (graph,input_data[i]['projectName'])
    #deleting of newline symbol from description
    description = input_data[i]['description'].replace("\n"," ")
    DATAPERIODS = {
        'graph': graph,
        'ontology': ontology,
        'dataPeriod' : dataPeriod,
        'project_name': project,
        'period' : input_data[i]['period'],
        'periodLevel' : input_data[i]['periodLevel'],
        'year' : year,
        'description' : description,
        'status' : input_data[i]['status']
    }

    if not input_data[i]['projectName'] in listProj:
        tripleProject = "{project_name} a <{ontology}#Project>
.\n".format(**DATAPERIODS)
        listProj.append(input_data[i]['projectName'])
        output_data.write(tripleProject)

```

```

        tripleDescription = "{project_name} <{ontology}#hasDescription>
'{description}' .\n".format(**DATAPERIODS)
        output_data.write(tripleDescription)
        tripleStatus = "{project_name} <{ontology}#hasStatus> '{status}'
.\n".format(**DATAPERIODS)
        output_data.write(tripleStatus)
        tripleData = "{dataPeriod} a <{ontology}#DataTakingPeriod>
.\n".format(**DATAPERIODS)
        output_data.write(tripleData)
        tripleAttrYear = "{dataPeriod} <{ontology}#hasYear>
<{ontology}#{year}> .\n".format(**DATAPERIODS)
        output_data.write(tripleAttrYear)
        tripleAttrPeriod = "{dataPeriod} <{ontology}#hasPeriod>
<{ontology}#{period}> .\n".format(**DATAPERIODS)
        output_data.write(tripleAttrPeriod)
        tripleAttrLevel = "{project_name} <{ontology}#hasLevel>
<{ontology}#Level{periodLevel}> .\n".format(**DATAPERIODS)
        output_data.write(tripleAttrLevel)
        mapping = "{project_name} <{ontology}#hasDataTakingPeriod>
{dataPeriod} .\n".format(**DATAPERIODS)
        output_data.write(mapping)

output_data.close()

```


Приложение Б (Листинг программы projects2TTL.py)

```
#choose graph
print("Current graph: " + graph + "\n")
graph_answer = input("Would you like to choose another one? [Y/N] ")
if graph_answer.lower() in ['y', 'yes']:
    graph = input("Please, insert a graph: ")
print("\nCurrent graph: " + graph + "\n")

#choose ontology
print("Current ontology: " + ontology + "\n")
ontology_answer = input("Would you like to choose another one? [Y/N] ")
if ontology_answer.lower() in ['y', 'yes']:
    ontology = input("Please, insert an ontology: ")
print("\nCurrent ontology: " + ontology + "\n")

#a ttl document with default name
output_object = open("projects.ttl", 'w')

#path to txt Projects
chosen_path = os.path.normpath(askopenfilename())

#getting extension
output_type = os.path.splitext(chosen_path)[1][1:].strip().lower()
if not ((output_type == 'txt') or (output_type == '')):
    print ('Wrong file type: ' + output_type)
    exit()

file_object = open(chosen_path, "r")
lines = file_object.readlines()
```

```

for i in lines:
    if not i.startswith("#"):
        project = "<%s/project/%s>" % (graph,i.rstrip('\n'))
        PROJECTS = {
            'graph': graph,
            'ontology': ontology,
            'project_name': project
        }
        triple = "{project_name} a <{ontology}#Project>
\n".format(**PROJECTS)
        output_object.write(triple)

output_object.close()
file_object.close()

```

Приложение В (Листинг программы campaigns2TTL.py)

```
#choose graph
print("Current graph: " + graph + "\n")
graph_answer = input("Would you like to choose another one? [Y/N] ")
if graph_answer.lower() in ['y', 'yes']:
    graph = input("Please, insert a graph: ")
print("\nCurrent graph: " + graph + "\n")

#choose ontology
print("Current ontology: " + ontology + "\n")
ontology_answer = input("Would you like to choose another one? [Y/N] ")
if ontology_answer.lower() in ['y', 'yes']:
    ontology = input("Please, insert an ontology: ")
print("\nCurrent ontology: " + ontology + "\n")

#path
chosen_path = os.path.normpath(askopenfilename())

#getting extension
output_type = os.path.splitext(chosen_path)[1][1:].strip().lower()
if output_type != 'json':
    print ('Wrong file type: ' + output_type)
    exit()

#a ttl document with default name
output_data = open("campaigns.ttl", 'w')

#input
with open(chosen_path) as data_file:
    input_data = json.load(data_file)
```

```

print("\nProcessing...\n")

for i, entry in enumerate(input_data):
    for j, campaign in enumerate(entry):
        #triple where campaign_name means campaign
        campaign_name = "<%s/campaign/%s>" % (graph,campaign)
        triple_campaign = campaign_name + " a <" + ontology +
"#Campaign> .\n"
        print("A triple was formed: " + triple_campaign)
        output_data.write(triple_campaign)
        for k, project in enumerate(entry[campaign]):
            #triple where campaign has project
            project_name = "<%s/project/%s>" % (graph,project)
            hasProject = "<%s#hasProject>" % (ontology)

            triple_hasProj = campaign_name + " " + hasProject + " " +
project_name + " .\n"
            print("A triple was formed: " + triple_hasProj)
            #writing a triple
            output_data.write(triple_hasProj)

            #the rest of the loop might be deleted if not needed
            for l, stepGroup in enumerate(entry[campaign][project]):
                hasProductionStep = "<%s#hasProductionStep>" %
(ontology)

                production_step = "<%s#%s>" % (ontology,stepGroup)
                triple_step = campaign_name + " " + hasProductionStep
+ " " + production_step + " .\n"
                print("A triple was formed: " + triple_step)

```

```

        output_data.write(triple_step)
    for m, dataFormat in
enumerate(entry[campaign][project][stepGroup]):
        hasDataSampleFormat =
"<%s#hasDataSampleFormat>" % (ontology)
        dataSampleFormat = "<%s#%s>" %
(ontology,dataFormat)
        triple_dataSampleFormat = campaign_name + " " +
hasDataSampleFormat + " " + dataSampleFormat + ".\n"
        print("A triple was formed: " +
triple_dataSampleFormat)
        output_data.write(triple_dataSampleFormat)
output_data.close()

```