

УВАЖАЕМЫЙ ПОЛЬЗОВАТЕЛЬ!

Обращаем ваше внимание, что система «Антиплагиат» отвечает на вопрос, является ли тот или иной фрагмент текста заимствованным или нет. Ответ на вопрос, является ли заимствованный фрагмент именно плагиатом, а не законной цитатой, система оставляет на ваше усмотрение. Данный отчет не подлежит использованию в коммерческих целях.

Отчет о проверке на заимствования №1

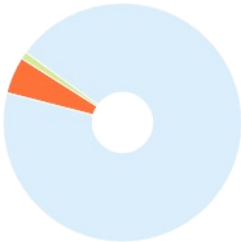
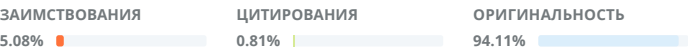
Дата загрузки: 09.06.2017 07:00:55
Автор: Каф Информационные системы ФИСТ infosys@ulstu.ru / ID: 158
Проверяющий: Каф Информационные системы ФИСТ (infosys@ulstu.ru / ID: 158)
Организация: Ульяновский государственный технический университет
Отчет предоставлен сервисом «Антиплагиат» - <http://ulstu.antiplagiat.ru>

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 13
Имя исходного файла: moiseev_20170609_zhest
Размер текста: 4824 кБ
Тип документа: Магистерская диссертация
Символов в тексте: 133312
Слов в тексте: 15902
Число предложений: 1094

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Отчет от 09.06.2017 07:01:00 - Последний готовый отчет (ред.)
Комментарии: не указано
Модули поиска:



№	Заимств.	Источник	Ссылка	Загрузка	Модуль поиска
[01]	1.23%	ГИС ЖКХ	http://ru.wikipedia.org	20 Ноя 2016	Модуль поиска Интернет
[02]	1.19%	Правовой календарь: изменения действующего законодательства, даты всту..	http://garant.ru	11 Ноя 2016	Модуль поиска Интернет
[03]	1.19%	Правовой календарь: изменения действующего законодательства, даты всту..	http://garant.ru	11 Ноя 2016	Модуль поиска Интернет
[04]	1.14%	Дипломный проект	http://studopedia.net	03 Мар 2016	Модуль поиска Интернет
[05]	1.14%	не указано	не указано	раньше 2011	Цитирование
[06]	1.13%	Чебан А.А. гр.123зс.doc	не указано	11 Мар 2016	Кольцо вузов
[07]	1.11%	69091	http://e.lanbook.com	раньше 2011	Модуль поиска ЭБС "Лань"
[08]	1.09%	О департаментеОсновными задачами Департамента яв.txt	не указано	17 Мая 2016	Кольцо вузов
[09]	1.09%	Эксплуатация жилищного фонда - процесс осуществлен.txt	не указано	22 Мая 2016	Кольцо вузов
[10]	0.79%	Приказ Министерства связи и массовых коммуникаций РФ и Министерства с..	http://garant.ru	23 Апр 2017	Модуль поиска Интернет
[11]	0.69%	КочубейСА 12-кб-пи1	не указано	09 Июн 2016	Кольцо вузов
[12]	0.69%	ДердянАА 13-см-2310	не указано	09 Июн 2016	Кольцо вузов
[13]	0.69%	СафаровРИ 12-кб-пи1	не указано	10 Июн 2016	Кольцо вузов
[14]	0.69%	БЕДРОСОВА_ОД 12-кб-пи1	не указано	14 Июн 2016	Кольцо вузов
[15]	0.63%	Сборник ПИС-2016.pdf	https://sibsutis.ru	16 Дек 2016	Модуль поиска Интернет
[16]	0.62%	БухминМЕ 13-см-2310	не указано	14 Июн 2016	Кольцо вузов
[17]	0.61%	Читать книгу Информационное право. Конспект лекций. Учебное пособие К..	http://iknigi.net	20 Фев 2017	Модуль поиска Интернет
[18]	0.58%	Веб-сервис для выявления заданной целевой аудитории на основе данных со..	не указано	17 Июн 2016	Кольцо вузов
[19]	0.55%	СИЗОВА ЖК Победа.doc	не указано	24 Июн 2015	Кольцо вузов
[20]	0.54%	ШереметЮО 13-см-2310	не указано	15 Июн 2016	Кольцо вузов
[21]	0.54%	СултановРА 12-кб-пр1	не указано	14 Июн 2016	Кольцо вузов
[22]	0.53%	Отделение прикладной математики и информатики/471ПМ Рогожин Игорь ...	не указано	05 Июн 2014	Кольцо вузов
[23]	0.52%	47415	http://e.lanbook.com	09 Мар 2016	Модуль поиска ЭБС "Лань"
[24]	0.5%	Разработка программного приложения «Расписание Транспорта» с использ...	не указано	17 Июн 2015	Кольцо вузов
[25]	0.49%	Сервис временных ссылок	не указано	28 Мая 2015	Кольцо вузов

[26]	0.44%	Факультет бизнес-информатики/273-М Жигалова Екатерина Сергеевна Автом.	не указано	13 Июн 2013	Кольцо вузов
[27]	0.43%	Технология проектирования автоматизированных систем обработки инфор..	http://ibooks.ru	09 Дек 2016	Модуль поиска ЭБС "Айбукс"
[28]	0.41%	МорозовДС 14-км-пр1	не указано	20 Июн 2016	Кольцо вузов
[29]	0.39%	Пояснительная записка к ВКР Подгорный ВА.docx	не указано	08 Дек 2016	Кольцо вузов
[30]	0.36%	DIPLOM.doc	не указано	16 Июл 2016	Кольцо вузов
[31]	0.36%	Программная реализация алгоритма "Гомоморфного шифрования" на прим.	не указано	02 Июн 2015	Кольцо вузов
[32]	0.31%	Прикладная информатика: технологии курсового и дипломного проектиров..	http://bibliorossica.com	26 Мая 2016	Модуль поиска ЭБС "БиблиоРоссика"
[33]	0.31%	49644	http://e.lanbook.com	09 Мар 2016	Модуль поиска ЭБС "Лань"
[34]	0.31%	Волокитина, Евгения Сергеевна диссертация ... кандидата технических наук : ...	http://dlib.rsl.ru	28 Фев 2014	Коллекция диссертаций РГБ
[35]	0.29%	Развитие индустрии по борьбе с вредоносным и мошенническим по б	http://davaiknam.ru	24 Мая 2016	Модуль поиска Интернет
[36]	0.29%	227997	http://biblioclub.ru	19 Апр 2016	Модуль поиска ЭБС "Университетская библиотека онлайн"
[37]	0.27%	С# 3.0. Справочник. 3 изд.	http://ibooks.ru	09 Дек 2016	Модуль поиска ЭБС "Айбукс"
[38]	0.25%	Вехов, Виталий Борисович диссертация ... доктора юридических наук : 12.00.0.	http://dlib.rsl.ru	30 Авг 2012	Коллекция диссертаций РГБ
[39]	0.23%	Бизнес в законе	http://ibooks.ru	09 Дек 2016	Модуль поиска ЭБС "Айбукс"
[40]	0.19%	Организация рекламной деятельности : учебное пособие	http://bibliorossica.com	27 Мая 2016	Модуль поиска ЭБС "БиблиоРоссика"
[41]	0.19%	259222	http://biblioclub.ru	19 Апр 2016	Модуль поиска ЭБС "Университетская библиотека онлайн"
[42]	0.19%	ЖИЛИЩНОЕ ПРАВО. Учебник и практикум для академического бакалавриат...	не указано	22 Фев 2017	Модуль поиска ЭБС "Юрайт"
[43]	0.18%	УПРАВЛЕНИЕ НЕДВИЖИМОСТЬЮ. Учебник и практикум для академического..	не указано	22 Фев 2017	Модуль поиска ЭБС "Юрайт"
[44]	0.17%	Социальная политика в сфере жилищно-коммунальных комплексов регион...	http://bibliorossica.com	26 Мая 2016	Модуль поиска ЭБС "БиблиоРоссика"
[45]	0.17%	Полный юридический справочник владельца квартиры, агента по недвижим..	http://ibooks.ru	09 Дек 2016	Модуль поиска ЭБС "Айбукс"
[46]	0.15%	65912	http://e.lanbook.com	09 Мар 2016	Модуль поиска ЭБС "Лань"
[47]	0.15%	61940	http://e.lanbook.com	09 Мар 2016	Модуль поиска ЭБС "Лань"
[48]	0.14%	Сборник (часть 2)	http://fa.ru	10 Ноя 2016	Модуль поиска Интернет
[49]	0.14%	61744	http://e.lanbook.com	09 Мар 2016	Модуль поиска ЭБС "Лань"
[50]	0.14%	Ильиных, Анастасия Леонидовна диссертация ... кандидата технических наук..	http://dlib.rsl.ru	30 Июл 2012	Коллекция диссертаций РГБ
[51]	0.14%	ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ. Учебник и практикум для..	не указано	22 Фев 2017	Модуль поиска ЭБС "Юрайт"
[52]	0.13%	Современные наукоемкие технологии. № 5, 2014. Часть 1	http://bibliorossica.com	26 Мая 2016	Модуль поиска ЭБС "БиблиоРоссика"
[53]	0.13%	Кондратьева, Мария Николаевна диссертация ... доктора экономических нау..	http://dlib.rsl.ru	26 Янв 2011	Коллекция диссертаций РГБ
[54]	0.13%	Организация и ведение научных исследований аспирантами	http://ibooks.ru	09 Дек 2016	Модуль поиска ЭБС "Айбукс"
[55]	0.12%	39164	http://e.lanbook.com	09 Мар 2016	Модуль поиска ЭБС "Лань"
[56]	0.11%	Основы электронной коммерции и бизнеса	http://bibliorossica.com	25 Мая 2016	Модуль поиска ЭБС "БиблиоРоссика"
[57]	0.11%	Защита компьютерной информации. Эффективные методы и средства	http://bibliorossica.com	27 Мая 2016	Модуль поиска ЭБС "БиблиоРоссика"
[58]	0.11%	Платонов, Юрий Георгиевич диссертация ... кандидата технических наук : 05...	http://dlib.rsl.ru	25 Дек 2015	Коллекция диссертаций РГБ
[59]	0.11%	Информационная безопасность: нормативно-правовые аспекты. Учебное п...	http://ibooks.ru	09 Дек 2016	Модуль поиска ЭБС "Айбукс"
[60]	0.11%	13783	http://e.lanbook.com	09 Мар 2016	Модуль поиска ЭБС "Лань"
[61]	0.11%	71733	http://e.lanbook.com	10 Мар 2016	Модуль поиска ЭБС "Лань"
[62]	0.11%	ЭЛЕКТРОННАЯ КОММЕРЦИЯ. Учебник и практикум для бакалавриата и маги..	не указано	22 Фев 2017	Модуль поиска ЭБС "Юрайт"
[63]	0.11%	49988	http://e.lanbook.com	09 Мар 2016	Модуль поиска ЭБС "Лань"
[64]	0.1%	Международный бухгалтерский учет. 2011. № 13/24	http://biblioclub.ru	20 Апр 2016	Модуль поиска ЭБС "Университетская библиотека онлайн"
[65]	0.1%	Защита компьютерной информации. Эффективные методы и средства	http://biblioclub.ru	21 Апр 2016	Модуль поиска ЭБС "Университетская библиотека онлайн"

[66]	0.1%	ИНФОРМАЦИОННОЕ ПРАВО. АКТУАЛЬНЫЕ ПРОБЛЕМЫ ТЕОРИИ И ПРАКТИКИ.	не указано	06 Мар 2017	Модуль поиска ЭБС "Юрайт"
[67]	0.1%	274647	http://biblioclub.ru	20 Апр 2016	Модуль поиска ЭБС "Университетская библиотека онлайн"
[68]	0.1%	13884	http://e.lanbook.com	09 Мар 2016	Модуль поиска ЭБС "Лань"
[69]	0.1%	Вычислительные системы, сети и телекоммуникации. 3-е изд.	http://ibooks.ru	09 Дек 2016	Модуль поиска ЭБС "Айбукс"
[70]	0.1%	Пробелы в российском законодательстве	http://ibooks.ru	09 Дек 2016	Модуль поиска ЭБС "Айбукс"
[71]	0.09%	Автоматизированное рабочее место врача-реаниматолога по направлению...	http://diplomba.ru	31 Дек 2015	Модуль поиска Интернет
[72]	0.08%	Информационные системы и базы данных: организация и проектирование	http://ibooks.ru	09 Дек 2016	Модуль поиска ЭБС "Айбукс"
[73]	0.08%	Диссертация	http://rsue.ru	13 Ноя 2016	Модуль поиска Интернет
[74]	0.08%	55342	http://e.lanbook.com	09 Мар 2016	Модуль поиска ЭБС "Лань"
[75]	0.08%	65020	http://e.lanbook.com	09 Мар 2016	Модуль поиска ЭБС "Лань"
[76]	0.08%	50569	http://e.lanbook.com	09 Мар 2016	Модуль поиска ЭБС "Лань"
[77]	0.07%	59434	http://e.lanbook.com	09 Мар 2016	Модуль поиска ЭБС "Лань"
[78]	0.07%	Критически важные объекты и кибертерроризм. Часть 2. Аспекты программ..	http://ibooks.ru	09 Дек 2016	Модуль поиска ЭБС "Айбукс"
[79]	0.07%	Пискунов, Владимир Владимирович диссертация ... кандидата экономически...	http://dlib.rsl.ru	раньше 2011	Коллекция диссертаций РГБ
[80]	0.07%	Информационные таможенные технологии: учебник: в 2 ч. Ч. 1	http://ibooks.ru	09 Дек 2016	Модуль поиска ЭБС "Айбукс"
[81]	0.06%	Информационные технологии в налогообложении (Налоговые информацио.	http://ibooks.ru	09 Дек 2016	Модуль поиска ЭБС "Айбукс"
[82]	0.06%	ЖИЛИЩНОЕ ПРАВО РОССИЙСКОЙ ФЕДЕРАЦИИ 6-е изд., пер. и доп. Учебник...	не указано	21 Фев 2017	Модуль поиска ЭБС "Юрайт"
[83]	0.06%	Административное право Российской Федерации	http://ibooks.ru	09 Дек 2016	Модуль поиска ЭБС "Айбукс"
[84]	0.05%	ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЮРИДИЧЕСКОЙ ДЕЯТЕЛЬНОСТИ 3-е и...	не указано	21 Фев 2017	Модуль поиска ЭБС "Юрайт"
[85]	0.05%	ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЮРИДИЧЕСКОЙ ДЕЯТЕЛЬНОСТИ 3-е и...	не указано	21 Фев 2017	Модуль поиска ЭБС "Юрайт"
[86]	0.05%	Вестник УлГТУ. - 2016. - ? 3 (75). - 80 с.	http://venec.ulstu.ru	11 Дек 2016	Модуль поиска Интернет
[87]	0.05%	Ребрий, Александр Валерьевич диссертация ... кандидата технических наук : ...	http://dlib.rsl.ru	раньше 2011	Коллекция диссертаций РГБ
[88]	0.05%	ko`chirish	http://library.tuit.uz	09 Ноя 2012	Модуль поиска Интернет
[89]	0.05%	63099	http://e.lanbook.com	09 Мар 2016	Модуль поиска ЭБС "Лань"

Текст отчета

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» 15

Факультет 4 ИСТ Кафедра Информационные системы

К ЗАЩИТЕ ДОПУСТИТЬ

Зам. зав. кафедрой

_____/ Т.В. Афанасьева / подпись инициалы, фамилия

« 15 » июня 2017 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема Организация межмодульного и межсистемного информационного обмена ИС ЖКХ .

Обучающийся _____ / В.В. Моисеев /

подпись инициалы, фамилия

Обозначение ВКР МД-УлГТУ-09.04.04-15/991-2017 Группа ПИМд-21

для технических направлений подготовки/специальностей

Направление подготовки (специальность) 09.04.04 « 4 Программная инженерия»

код, наименование

Руководитель ВКР _____ / В.В. Воронина /

подпись, дата инициалы, фамилия

Рецензент _____ / ____ В.М. Кандаулов ____/

подпись, дата инициалы, фамилия 4

Ульяновск

2017 4

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» 15

Факультет 4 ИСТ Кафедра Информационные системы

Направление подготовки (специальность) 09.04.04 «Программная инженерия»

УТВЕРЖДАЮ

Зам. зав. кафедрой

_____ / Т.В. Афанасьева / подпись инициалы, фамилия

« » 2017 г.

ЗАДАНИЕ

на магистерскую диссертацию

вид ВКР (дипломный проект (работа) / бакалаврская работа / магистерская диссертация)

обучающемуся Моисееву Владиславу Валерьевичу курса 2 группы ПИмд-21

фамилия, имя, отчество

Тема ВКР Организация межмодульного и межсистемного информационного обмена
ИС ЖКХ.

утверждена приказом по университету от «22» декабря 2016 г. No 2607

Срок сдачи обучающимся законченной ВКР: «15» 4 июня 2017 г.

Исходные данные к ВКР 4 Существующие модели данных информационных систем
жилищно-коммунального хозяйства, существующая организация обмена между
системами, требования к информационному обмену.

Содержание расчетно-пояснительной записки (перечень подлежащих разработке
вопросов) 4 формулировка проблемы информационного обмена; формальное
определение механизма межмодульного и межсистемного информационного
обмена; описание доработки программной системы для организации
информационного обмена; описание вычислительных экспериментов по теме
магистерской диссертации.

Перечень графического материала (с точным указанием
обязательных

чертежей) _____

Календарный график работы над ВКР на весь период (с указанием сроков выполнения
и содержания отдельных этапов) 4

No

этапа

Содержание этапа Срок выполнения

- 1 4 Определение проблемы информационного обмена Декабрь 2017 года
- 2 Формализация информационного обмена Январь 2017 года
- 3 Описание видов информационного обмена Февраль 2017 года
- 4 Доработка программной системы Март 2017 года
- 5 Проведение вычислительных экспериментов Апрель 2017 года
- 6 Написание текста диссертации Май 2017 года

Дата выдачи задания «__» _____ 20__ г.

Руководитель доцент кафедры ИС, к.т.н. _____ / В.В. Воронина /

подпись обучающегося

инициалы, фамилия

АННОТАЦИЯ

Выпускная квалификационная работа Моисеева Владислава
Валерьевича по теме «Организация межмодульного и межсистемного

информационного обмена ИС ЖКХ». Руководитель Воронина Валерия
Вадимовна. Защищена на кафедре «Информационные системы» УлГТУ в 2017
году.

Пояснительная записка: 106 с., 4 разд., 4 прил., 22 рис., 17 табл., 26 ист.

Ключевые слова: информационный обмен,

жилищно-коммунальное хозяйство, РИАС ЖКХ, ГИС ЖКХ, SOAP, REST, SQL.

В данной выпускной квалификационной работе рассматривается вопрос
организации информационного обмена между информационными системами и
их мо-

дулями, разработанными в сфере жилищно-коммунального хозяйства.

В первой главе приводится описание рассматриваемых систем жилищно-
коммунального хозяйства, определение и виды информационного обмена, а также
ука-

зываются правовые основы информационного обмена в России.

Во второй главе приводится формализация модели информационного
обмена, выделяются основные характеристики элементов модели, приводятся
основные комбинации характеристик, применимых в сфере ЖКХ, а также
рекомендации к их
применению.

В третьей главе приводится описание дорабатываемой программной системы,
где информационный процесс построен на основании выведенной модели.

В четвёртой главе описываются вычислительные эксперименты, которые
позволяют сделать выводы о корректности применения различных механизмов
информационного обмена к прикладным задачам.

2.2.1 Характеристики источника и приёмника данных 28

2.2.2 Характеристики канала связи 30

2.2.3 Характеристики протокола передачи данных 31

2.2.4 Характеристики данных (сообщений)

2.3 Основные алгоритмы построения информационного взаимо-
31

действия 32

2.3.1 Организация прямого обмена 32

2.3.2 Обмен реестрами 34

2.3.3 Построение веб-сервисов SOAP и REST API 37

3 Программное обеспечение модели 40

3.1 Модуль интеграции РИАС ЖКХ с ГИС ЖКХ 40

3.1.1 Общее описание 40

3.1.2 Механизм информационного обмена 42

3.1.3 Синхронизация данных 44

3.1.4 Пользовательский интерфейс 46

3.1.5 Проблемы информационного взаимодействия 47

3.1.6 Описание модели данных 51

3.1.7 Объём самостоятельной разработки 52

3.2 Платёжный шлюз РИАС ЖКХ 52

3.2.1 Описание модели данных 54

3.2.2 Объём самостоятельной разработки

3.3 Модуль интеграции РИАС ЖКХ с «АИС Город. Система на-
59

числений» 59

3.3.1 Объем самостоятельной разработки 63

3.4 Подсистема обработки реестров РИАС ЖКХ 63

3.4.1 Описание модели данных 64

3.4.2 Объем самостоятельной разработки 68

4 Описание экспериментов 70

4.1 Нагрузочное тестирование веб-сервиса платежей ГИС ЖКХ . . 70

4.2 Нагрузочное тестирование веб-сервиса платежей РИАС ЖКХ .

4.3 Сравнение скорости передачи данных различными методами

74

обмена 76

Заключение 83

Список использованных источников 84

Приложение А Текст программы

Приложение Б SQL-команды генерации данных для вычислитель-

88

ных экспериментов 98

Приложение В Апробация работы 103

Приложение Г Список публикаций по теме диссертации 106

СПИСОК ИСПОЛЬЗОВАННЫХ СОКРАЩЕНИЙ И

ОБОЗНАЧЕНИЙ

ГЖИ— государственная жилищная инспекция.

ГИС ЖКХ— государственная информационная система жилищно-ком-

мунального хозяйства.

ГУИД— глобальный уникальный идентификатор.

ЕГРЮЛ — единый государственный реестр юридических лиц.

ИС— информационная система.

ЖКХ—жилищно-коммунальное хозяйство.

НСИ— нормативно-справочная информация.

ОКЕИ— общероссийский классификатор единиц измерения.

ОКТМО— общероссийский классификатор территорий муниципальных об-

разований (ОК 033—2013).

РИАСЖКХ—региональнаяинформационно-аналитическаясистемажилищ-

но-коммунального хозяйства.

СУБД— система управления базами данных.

ФИАС— федеральная информационная адресная система.

REST—RepresentationalStateTransfer.Архитектурныйстильвзаимодействия

компонентов распределённого приложения в сети.

SOAP — Simple Object Access Protocol. Протокол обмена структурированны-

ми сообщениями в распределённой вычислительной с реде.

SQL — Structured Quer 25 y Language. Формальный непроцедурный язык

программирования, применяемый для создания, модификации и управления

данными

в произвольной реляционной базе данных, управляемой соответств 51 ующей СУБД.

XML — eXtensible Markup Language. Расширяемый язы 29 к разметки. 87

ВВЕДЕНИЕ

Краткое описание предметной области

Жилищно-коммунальное хозяйство Российской Федерации — крупнейшая

сфера экономики государства, которой присуща информационная

неоднородность. Множество компаний ведут свою деятельность в этой сфере,

существует огромное множество нормативно-правовых актов, регулирующих

деятельность таких орга-

низаций.

Повышение прозрачности в жилищно-коммунальном хозяйстве — одна из

важнейших целей государства. Она достигается различными способами. Одним

из таких является обязанность некоторых типов организаций сферы ЖКХ

раскрывать информацию о собственной деятельности в электронном виде.

Например, управляющие организации обязаны раскрывать информацию о финансово-хозяйственной деятельности, банки — о поступивших платежах за ЖКХ, ресурсоснабжающие организации — о договорах на поставку ресурсов.

Помимо поставщиков услуг и ресурсов, существуют организации и органы власти, регулирующие отношения и распределение финансовых потоков в сфере ЖКХ. К таким можно отнести государственные жилищные инспекции (главрегионнадзоры или госжилинспекции) и фонды капитального ремонта. Таким организациям требуется как изучать электронную отчётность поставщиков, так и самим отчитываться в электронной форме.

Для повышения прозрачности и раскрытия информации о деятельности организаций-участников ЖКХ было создано огромное множество информационных

систем. Такие информационные системы можно разделить на:

- государственные;
- региональные;
- муниципальные;
- собственные.

Так как с каждым годом растёт нагрузка на поставщиков информации по раскрытию информации, были придуманы и реализованы механизмы информационного взаимодействия между информационными системами. Они должны значительно упростить процесс раскрытия информации, так как до сих пор существует двойное (а иногда и тройное) дублирование информации в различные информационные системы.

Актуальность

Бурное развитие информационных технологий, несомненно, сказалось на увеличении комбинирования цифровых и материальных данных. Документооборот переходит в цифровую среду, как и подпись документов. Многие книги учёта переводятся в электронную форму, так как с цифровой информацией легче работать. Сфера жилищно-коммунального хозяйства не стала исключением. На территории Ульяновской области в первом десятилетии XXI века был разработан ряд информационных систем для систематизации и упрощения работы в сфере жилищно-коммунального хозяйства. В 2015-2016 годах эти информационные системы было решено объединить в РИАС ЖКХ субъекта Российской Федерации — программный комплекс, охватывающий всю сферу ЖКХ региона внедрения.

Параллельно с этим процессом государство решило разработать государственную информационную систему жилищно-коммунального хозяйства (ГИС ЖКХ), описанную в федеральном законе 209-ФЗ от 21 июля 2014 года. Данная информационная система предполагает хранение всей доступной информации о сфере ЖКХ [17]. Так как в РИАС ЖКХ вся необходимая информация уже есть, было решено настроить информационный обмен между системами.

Существуют также и другие информационные системы (государственные и муниципальные), с которыми необходимо настраивать взаимодействие в качестве региональной информационной системы жилищно-коммунального хозяйства. Однако существующая на данный момент несогласованность некоторых данных в РИАС ЖКХ (отдельные системы разрабатывались параллельно около десяти лет) не даёт настроить обмен с ГИС ЖКХ и другими внешними системами корректно.

Данное исследование необходимо для понимания всей картины организации информационного обмена как между модулями РИАС ЖКХ, так и с внешними системами, с последующим описанием и реализацией выбранных методик обмена.

Научная новизна

Научная новизна данной работы: формализация механизмов обмена, применимых в сфере ЖКХ, а также выявленные на основе этого формального описания характеристик и ограничений.

Сформулированные на основе выявленных характеристик и ограничений рекомендации по организации обмена могут быть применены для различных классов информационных систем, не обязательно связанных с жилищно-коммунальным хозяйством.

Положения, выносимые на защиту

- а) модель организации информационного обмена;
- б) программная система, реализующая описанную модель;
- в) результаты вычислительных экспериментов.

1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Описание ГИСЖКХ России

Государственная информационная система жилищно-коммунального хозяйства России — программный комплекс, реализуемый в рамках 209 ФЗ [17] и устанавливающий следующие цели:

- увеличение прозрачности сферы ЖКХ страны;
- унификация электронных видов информации, связанных с жилищно-коммунальным хозяйством;
- объединение разнородных федеральных информационных систем, связанных с ЖКХ;
- снижение расходов на информационное обеспечение участников сферы жилищно-коммунального хозяйства.

Для достижения поставленных целей перед ГИС ЖКХ поставлены следующие задачи:

- закрепление на федеральном уровне обязанности раскрывать информацию в ГИС ЖКХ всеми участниками сферы ЖКХ;
- закрепление на федеральном уровне перечня, способов и сроков раскрытия информации в ГИС ЖКХ;
- реализация информационного взаимодействия с федеральными системами и сервисами электронного правительства России;
- реализация информационного взаимодействия с региональными, муниципальными, коммерческими и собственными информационными системами;
- возможность формирования отчётов, в том числе для аналитических исследований.

Согласно 209 ФЗ ГИС ЖКХ должна учитывать следующие принципы: [17]

- открытость, прозрачность и общедоступность информации;
- однократность размещения в системе информации;
- многократность использования информации, размещённой в системе;
- непрерывность и бесперебойность функционирования системы;
- полнота, достоверность, актуальность информации и своевременность её размещения в системе;
- бесплатность размещения в системе информации;
- использование единых форматов для информационного взаимодействия информационных систем с системой.

1.1.1 Устройство ГИСЖКХ

ГИС ЖКХ разделяется на открытую и закрытую часть. Снимок экрана открытой части портала ГИС ЖКХ представлен на рисунке 1.

В закрытой части ГИС ЖКХ интерфейс зависит от функций организации в ГИС ЖКХ, а также от роли пользователя в организации.

В ГИС ЖКХ существует две роли: администратор организации и

уполномоченный специалист. Уполномоченный специалист может выполнять

действия по раскрытию информации от имени организации согласно 209-ФЗ.

Администратор организации может устанавливать ограничения в правах

уполномоченным специалистам, совершать в системе сделки от имени

организации, передавать полномочия на раскрытие данных от имени организации

другим компаниями или информаци-

онным системами.

Функций у одной организации может быть несколько. Перечень основных

функций организации, возможных в ГИС ЖКХ:

– управляющая организация;

– ресурсоснабжающая организация;

– оператор ГИС ЖКХ;

– орган исполнительной власти субъекта РФ, уполномоченный на

осуществление государственного жилищного надзора;

– 7 орган местного самоуправления, осуществляющий муниципальный жилищный

контроль;

– федеральный орган исполнительной власти в области государс 6 твенного

регулирования тарифов;

– орган государственной власти субъекта 6 РФ;

– орган местного самоуправления;

– региональный оператор по обращению с твердыми коммунальными отхо-

дами;

– товарищество собственников жилья;

– 1 жилищный кооператив;

– жилищно-строительный кооператив;

– 1 оператор информационной системы;

– оператор по приему платежей;

– платежный субагент;

– расчетный центр.

Снимок экрана закрытой части портала ГИС ЖКХ для уполномоченного спе-

циалиста управляющей организации представлен на рисунке 2.

С технической точки зрения ГИС ЖКХ можно разделить на веб-портал,

REST API для портала, набор открытых данных и информационное

взаимодействие с

внешними системами по протоколам SOAP.

Обмен данными может быть построен в ручном или автоматическом

режиме. Для ручного режима предусмотрен ввод данных через веб-портал ГИС

ЖКХ, а также при помощи Excel-шаблонов. Для автоматического обмена

существует ин-

формационное взаимодействие с внешними системами и набор открытых данных.

Необходимые данные для автоматического обмена находятся в разделе регла-

ментов и инструкций.

Для проведения испытаний обмена с ГИС ЖКХ существуют стенды информацион-

ного тестирования СИТ-01 и СИТ-02. Сам портал ГИС ЖКХ называется про-

Рисунок 1 – Снимок экрана открытой части портала ГИС ЖКХ

мышленным программного-аппаратным комплексом (ППАК).

Форматы информационного взаимодействия разделяются на текущие и

перспективные. Текущие форматы актуальны для ППАК ГИС ЖКХ и стенда СИТ-

01.

Соответственно, перспективные форматы актуальны для стенда СИТ-02.

1.2 Описание РИАС ЖКХ субъекта России

Региональная аналитическая система жилищно-коммунального хозяйства субъек-

екта Российской Федерации (РИАС ЖКХ) — программный комплекс решений по

автоматизации сферы ЖКХ. РИАС ЖКХ разрабатывается с 2010 года компанией

«АИС Город» (Ульяновск).

Отличительные особенности РИАС ЖКХ по сравнению с аналогичными раз-

работками:

– система может быть внедрена как в рамках субъекта России, так и не опираясь на конкретную территорию;

Лист

15

Рисунок 2 – Снимок экрана закрытой части портала ГИС ЖКХ для уполномоченного специалиста управляющей организации

– тесная интеграция с федеральными и коммерческими информационными системами;

– наличие справочников ФИАС, адресного плана ГИС ЖКХ, ОКТМО, ОКЕИ, НСИ ГИС ЖКХ и иных отраслевых справочников;

– модульность системы, благодаря чему достигается её низкая стоимость;

– реализация постановлений Правительства России No 731, No 1468 в сфере жилищно-коммунального хозяйства;

– гибкая настройка развёртки системы под нужды заказчика;

– быстрая реакция технической поддержки.

Снимок экрана главной страницы РИАС ЖКХ для администратора системы представлен на рисунке 3.

Лист

16

Рисунок 3 – Снимок экрана главной страницы РИАС ЖКХ для администратора системы

1.3 Проблема информационного взаимодействия

В современном IT-мире существует множество информационных систем и технологий, позволяющих хранить и обрабатывать данные. Зачастую эти данные дублируются, а если данные занесены пользователем, данные могут дублироваться частично. Соответственно, возникает множество конфликтных ситуаций. Например, какую информационную систему в данном случае считать более эталонной или

как избавиться от дублирования информации в различных системах.

Для разрешения последнего вопроса были разработаны методы взаимодействия между информационными системами, которые позволяют обмениваться информацией между системами в автоматическом или автоматизированном режимах.

Как и в любой сфере деятельности, в информационном взаимодействии инженеры постоянно стремились (и стремятся) всё формализовать и стандартизировать. Появляются стандарты и протоколы, описывающие формат сообщений, требования к каналам передачи данных, защите информации. Также формализуются шаблоны проектирования информационных систем для наименее затратной организации информационного обмена с иными системами.

Одни методы делают упор на скорость передачи данных, другие – на контроль целостности и защищённость, третьи – на лёгкость реализации механизмов обмена. Всё это требуется учитывать при реализации информационного взаимодействия.

С точки зрения разработки региональной системы жилищно-коммунального хозяйства проблема информационного взаимодействия состоит в выборе методов обмена с муниципальными и собственными информационными системами и в реализации клиентов информационного обмена с федеральными информационными системами.

1.4 Обоснование необходимости решения проблемы

Грамотная организация информационного обмена крупной информационной системы, состоящей из множества модулей, позволит:

– уменьшить количество вводимой пользователями информации;

– увеличить полезность системы на рынке ЖКХ;

– уменьшить несогласованность данных в различных информационных си-

стемах;

- добавить дополнительные точки роста системы;

- расширить отчётные данные (например, об использовании системы различными организациями).

1.5 Определение информационного взаимодействия

Информационное взаимодействие — процесс обмена информацией

между источником и приёмником по каналам связи. В более узком смысле

информационное взаимодействие двух информационных систем (межсистемный

информационный обмен) можно трактовать как процесс передачи информации

между информационными системами при помощи телекоммуникационной сети

Интернет. В свою очередь межмодульный информационный обмен — подвид

информацион-

ного взаимодействия, производимый внутри информационной системы.

Несмотря на схожесть определений межсистемного и межмодульного

обмена,

они решают разные цели и характеризуются следующими отличиями:

- межсистемный обмен надёжнее защищён нежели межмодульный;

- межсистемный обмен лучше документирован;

- межмодульный обмен намного быстрее межсистемного;

- межмодульный обмен может чаще обновляться, так как источник и

приёмник информации – сама информационная система;

- межмодульный обмен может использовать более узкий набор технологий.

1.6 Виды современного информационного взаимодействия 1.6.1

Прямой обмен между БД

Данный метод обычно используется при межмодульном взаимодействии.

Он заключается в том, что источник и приёмник используют одну и ту же базу

данных либо связанные базы данных. Например, СУБД Microsoft SQL Server

позволяет

связать несколько СУБД для доступа к базам данных не только той СУБД, где находится

исходная база данных.

Преимуществами прямого доступа к базе данных является:

- высокая скорость работы;

- отсутствие лишних издержек для построения взаимодействия.

Несмотря на преимущества, у данного метода есть и очевидные недостатки:

- отсутствие какой-либо защиты данных от несанкционированного доступа к

ним;

- «привязка» к определённой СУБД или технологии; – сложность изменения

форматов взаимодействия.

Прямой обмен между базами данных можно разделить на активный и

пассивный. При активном обмене источник данных сам передаёт данные в базу

данных получателя. При пассивном обмене происходит обратный процесс:

получатель

самостоятельно забирает данные из базы данных источника.

Также такой вид информационного взаимодействия можно разделить на

автоматизированный и автоматический. При автоматизированном обмене

персонал должен инициировать обмен. При автоматическом обмене участие

персонала не

требуется.

1.6.2 Обмен реестрами

Обмен структурированными файлами (реестрами) берёт своё начало очень

давно: ещё со времени, когда многие информационные системы не были

соединены между собой прямыми каналами связи (как сейчас при помощи сети

Интернет). Однако, несмотря на тотальное объединение информационных систем

в одну сеть,

данный метод информационного обмена является одним из самых популярных.

Преимущества обмена реестрами:

- простота реализации;
- информационные системы, разработанные давно, как правило, поддерживают обмен реестрами;
- организация защиты информации в данном методе обмена – не обязанность этого метода;

Недостатки обмена реестрами:

- принципиальная сложность передачи бинарных файлов;
- огромное количество кодировок могут исказить информацию;
- могут использоваться устаревшие технологии (например, dBase III).

В текущий момент набирает популярность аналог данного метода

информационного обмена: предоставление шаблонизированных книг Microsoft

Excel. С одной стороны, с ними гораздо удобнее работать, нежели с файлами CSV

или таблицами dBase, однако для этого требуется платное программное

обеспечение.

1.6.3 SOAP

SOAP представляет собой легковесный протокол, предназначенный для

обмена структурированными данными в децентрализованной информационной

среде [25]. Последняя актуальная версия протокола — 1.2. Для определения

форматов

сообщения протокол использует XML.

SOAP не предъявляет чётких требований к протоколу передачи данных, безопасности и гарантии доставки сообщений, однако позволяет использовать для этих

целей расширения [25]. Обычно для протокола передачи данных используется HTTPS

или HTTP, но возможны и более экзотические варианты, например, SMTP.

Для описания схем информационного обмена используется XSD. Для опре-

деления веб-служб используется язык WSDL.

Преимущества SOAP:

- формат сообщений стандартизирован;
- может быть использован любой протокол прикладного уровня;
- легко обеспечить защиту сообщений при помощи подписи данных;
- существует множество реализаций серверов и клиентов для современных языков программирования;
- самодocumentируемость протокола позволяет сократить объём артефактов, необходимых для организации информационного обмена.

Недостатки SOAP:

- используется избыточный формат сообщений —XML;
- необходима высокая квалификация аналитиков и разработчиков веб-сервиса, чтобы им было удобно пользоваться;
- протокол сложен в реализации на мобильных устройствах и некоторых настольных системах.

Протокол SOAP используется в основном для организации информационного

взаимодействия между внешними системами, слабо связанными друг с

другом. Например, такой вид обмена может быть настроен между

информационной системой государственного или регионального уровня с

более мелкими ИС. Для построения микросервисной архитектуры внутри

программного комплекса или реализации обмена с мобильными

платформами и фронтами веб-приложений рекомендуется ис-

пользовать подход REST.

1.6.4 REST

Существует альтернативный протоколу SOAP механизм информационного

обмена в WWW, распространённый в текущее время.

REST определяет набор архитектурных принципов, придерживаясь которых

можно создавать веб-сервисы, основываясь лишь на том, как информация будет

передаваться по протоколам HTTP (HTTPS) к клиентам, написанным на разных

язы-

ках программирования [24].

Основные принципы архитектурного стиля REST к построению информационного обмена следующие:

- явное использование методов HTTP (GET, POST, PUT, DELETE);
- запросы к API не зависят друг от друга (отсутствует понятие состояния или сессии на стороне сервера);
- URL-адреса доступа к API структурированы;
- для описания данных используется XML, JSON (рекомендуется) или обе технологии вместе [24].

Преимущества подхода REST к построению информационного взаимодействия:

- высокая полезная нагрузка сообщений из-за использования JSON;
- более лёгкая обработка данных при помощи современных технологий;
- из-за отсутствия на сервере сессий можно легко строить отказоустойчивые кластеры.

Недостатки REST:

- нет единого принятого стандарта описания форматов сообщений;
- отсутствует единое описание защиты передаваемых данных;
- если веб-сервис большой, или если веб-сервисов много, легко можно запутаться в URL-адресах конечных точек.

Данный вид взаимодействия хорошо себя показал при обмене данными внутри веб-приложений между клиентским и серверным кодом. Подход REST рекомендуется использовать при общении мобильных приложениями. Так же такой принцип можно использовать при организации синхронизации между модулями программного комплекса.

1.7 Правовые основы информационного обмена в ЖКХ

Ниже перечислены основные законы и подзаконные акты Российской Федерации, согласно которым любые организации, относящиеся к сфере жилищно-коммунального

хозяйства, должны вести отчётность в электронной форме:

а) Федеральный закон от 21 июля 2014 года N 209-ФЗ «О государственной информационной системе жилищно-коммунального хозяйства»

б) Приказ от 29.02.2016 года No 74/114/ пр «Об утверждении состава, сроков и периодичности размещения информации поставщиками информации в государственной информационной системе жилищно-коммунального хозяйства»;

в) Приказ от 02.03.2016 года No 77/120/ пр « Об утверждении состава, порядка, сроков и периодичности размещения в государственной информационной системе жилищно-коммунального хозяйства информации о предоставлении субъектам Российской Федерации и муниципальным образованиям финансовой поддержки на проведение капитального ремонта многоквартирных домов, переселение граждан из аварийного жилищного фон-

да, модернизацию систем коммунальной инфраструктуры, а также о выполнении условий предоставления такой финансово»;

г) Приказ от 28.01.2016 года No 18/34/ пр «Об утверждении состава, порядка, способов, сроков и периодичности размещения в государственной информационной системе жилищно-коммунального хозяйства информации о количестве зарегистрированных в жилых помещениях по месту пребывания и по месту жительства граждан»;

д) Приказ от 28.12.2015 года No 589/944/ пр « Об утверждении Порядка и способов размещения информации, ведения реестров в государственной информационной системе жилищно-коммунального хозяйства, доступа к си-

стеме и к информации, размещённой в ней»

е) Постановление Правительства РФ от 23.09.2010 года No 731 «**19** Об

утверждении стандарта раскрытия информации организациями,

осуществляю-

щими деятельность в сфере управления многоквартирными домами»;

ж **15**) Приказ от 1 декабря 2016 года No 871/пр « Об утверждении форм мони-

торинга и отчётности реализации субъектами Российской Федерации

региональных программ капитального ремонта общего имущества в **5**

многоквартирных домах и признании утратившими силу отдельных

Приказов Минстроя России». **5**

Нормативно-правовые акты как помогают организовать автоматизацию ЖКХ и перевод этой сферы в электронную форму, так и мешают выполнять некоторые действия. Например, в приказе No 74/114/пр описываются основные виды информации, необходимые к предоставлению в электронном виде, а также состав этих видов информации. Более подробно это можно изучить в [12]. С одной стороны, требование этого нормативного акта в техническом задании упрощает целый ряд действий по формализации модели автоматизации. С другой же стороны, этот документ накладывает огромные ограничения в удобстве работы системы, ведь многие виды информации в нём являются излишне формализованными и связанными.

Как пример, при ведении протоколов собраний собственников требуется заносить информацию о каждом вопросе с обязательным указанием того, сколько проголосовало за, против или воздержалось. Несомненно, это может помочь при анализе данных, однако, скорее всего, достаточным требованием было бы указать только результат собрания, а его детали можно узнать и из электронного образа документа (скан-копии протокола собрания), который также является обязательным.

2 МОДЕЛИ, МЕТОДЫ И АЛГОРИТМЫ
ИССЛЕДОВАНИЯ

2.1 Модель информационного взаимодействия

Для формального определения информационного взаимодействия необходи-
мо определить его элементы.

Прежде всего, в обмене информацией участвует источник данных S и приёмник D. В реальной жизни ими являются программного-аппаратные комплексы информационных систем, однако в случае моделирования и источник, и приёмник можно описать конечным набором некоторых характеристик. Представим AS как множество характеристик источника и AD — множество характеристик приёмника. Тогда $AS \cap AD = ASD$ — множество совпадающих характеристик и источника, и приёмника данных. Представим источник S и приёмник данных D через столбцы их характеристик:

$$\begin{matrix} s_1 & d_1 \\ \dots & \dots \\ S = s_n, D = d_n, \dots \dots SD \mid s \mid k \mid SD \mid d \mid m \mid S \mid D \end{matrix}$$

где n – мощность множества $A (n = A)$, $k = A$, $m = A$, $s \in A$, $d \in A$.
Примеры характеристик источника и приёмника данных описаны в п. 2.2.1.
Данные передаются по каналу связи C. Им может являться телекоммуникационная сеть Интернет, локальная вычислительная сети предприятия и даже ручная передача данных от источника к приёмнику. Канал связи также обладает некоторым набором характеристик AC, через которые можно его описать. Таким образом,

$$\begin{matrix} c_1 \\ \dots \\ C \\ c_m \end{matrix}$$

где $m = |AC|$, $c \in AC$. Примеры характеристик канала связи описаны в п. 2.2.2.
Данные от источника к приёмнику передаются по определённым правилам,

закреплённым в протоколе информационного обмена Р. С точки зрения моделирования процесса информационного взаимодействия нас не интересуют отдельные правила передачи данных, закреплённые в протоколе. Таким образом, протокол Р

также можно описать при помощи столбца характеристик AP:

$$\begin{matrix} p_1 \\ \vdots \\ p_j \end{matrix}$$

где $j = |AP|$, $p \in AP$. Примеры характеристик протокола описаны в п. 2.2.3.

Информационный обмен невозможен без описания самих данных, передаваемых в ходе взаимодействия. Для упрощения назовём такие данные сообщениями. Пусть множество AM – множество характеристик передаваемых сообщений. Тогда сообщения M можно описать через набор их характеристик:

$$\begin{matrix} m_1 \\ \vdots \\ m_q \end{matrix}$$

где $q = |AM|$, $m \in AM$. Примеры характеристик данных (сообщений) описаны в п. 2.2.4.

Если представить IS как множество данных у источника, необходимых к передаче, а ID как множество записей у приёмника данных, полученных в ходе информационного обмена, то процесс информационного обмена можно описать как поиск оператора F, позволяющего инъективно отобразить IS на ID. Таким образом, $F : IS \rightarrow ID, \forall is \in IS \exists id \in ID : id = F(is)$

Таким образом, информационное взаимодействие E можно определить следующим образом:

$$E = \langle S, D, C, P, M, IS, ID, F \rangle,$$

где E – информационное взаимодействие, S – источник данных, D – приёмник данных, C – канал связи, P – протокол передачи данных, M – передаваемое сообщение, IS – множество записей источника, ID – множество данных приёмника, F – оператор отображения. Визуально описанные выше компоненты можно представить в виде рисунка 4.

2.2 Характеристики информационного взаимодействия

2.2.1 Характеристики источника и приёмника данных

Общими характеристиками источника и приёмника данных могут являться:

- скорость обработки данных;
- максимальное число потоков обработки данных;
- вид и величина деградации скорости обработки данных при увеличении объёма;
- вид и величина деградации скорости обработки данных при увеличении потоков.

Основной характеристикой является скорость обработки данных. Её можно измерять как в байтах в секунду, так и в абстрактных единицах передачи информации (аналог бод) в единицу времени.

Рисунок 4 – Упрощённая схема информационного взаимодействия (аналог бод) в единицу времени. Прежде всего, скорость обработки зависит от вычислительной мощности аппаратного комплекса и от оптимизации программного комплекса источника или приёмника данных. В информационном обмене нет смысла выделять величину оперативной памяти, накладные расходы на исполнение команд или тактовую частоту процессора: все эти величины объединяются в скорость обработки.

Также, если источников или приёмников данных несколько, необходимо учитывать в модели информационного обмена и максимальное число потоков обработки. Эта характеристика напрямую зависит от конфигурации программной платформ-

мы, так как аппаратно многопоточность работает даже на одноядерных системах.

Дополнительно в характеристиках источника и приёмника данных можно выделить характеристики высших порядков. Примерами их могут служить вид и величина деградации скорость обработки данных при увеличении объёма или потоков. Эти характеристики удобно использовать при анализе отказоустойчивости системы. Аналог указанных характеристик в программной инженерии – оценка сложности алгоритма (по действиям и по памяти).

2.2.2 Характеристики канала связи

Характеристиками канала связи могут являться:

- максимальная пропускная способность;
- процент ошибок при передаче данных;
- возможность определения и/или корректировки ошибок;
- предельная дальность передачи информации;
- временная задержка при передаче.

Основной величиной описания модели канала связи является максимальная пропускная способность, так как именно эта характеристика наиболее явно влияет на скорость обмена. Чем больше пропускная способность канала связи, тем больше информации можно передать за единицу времени. Однако, в случае, если пропускная способность намного выше скорость обработки данных одним из участников информационного обмена (источником или приёмником данных), то дальнейшее увеличение указанной характеристики бессмысленно.

Любой канал связи может искажать информацию. Зачастую это случается из-за неоднородности среды, где этот канал прокладывается. Например, в случае с электричеством это могут быть различные наводки (особенно сильно они влияют

в многожильных каналах связи). Обычно протоколы передачи информации позволяют определять и корректировать ошибки, и с точки зрения информационного обмена между ИС никаких ошибок не возникает. Но не стоит упускать из внимания данную

характеристику, так как она напрямую влияет на скорость передачи данных.

В некоторых моделях информационного обмена играет роль предельная дальность передачи информации. Она обычно измеряется в метрах или километрах. Если источник и приёмник данных находятся дальше указанного расстояния, обмен происходить не будет.

Также иногда играет роль задержка при передаче информации от источника к приёмнику. Это может быть важно при проектировании быстрых сервисов или сервисов реального времени. В ЖКХ нет таких чётких ограничений по задержке информации как, например, в оборонной сфере, поэтому данная характеристика приводится для общего ознакомления.

2.2.3 Характеристики протокола передачи данных

Характеристиками протокола передачи данных могут являться:

- коэффициент избыточности сериализованных данных;
- сложность сериализации и десериализации;
- возможность шифрования;
- возможность валидации.

При описывании протокола передачи данных важно учитывать особенности для конкретных целей моделирования.

Например, при оценке нагрузки на вычислительную сеть будет полезно ввести характеристику избыточности сериализованных данных. Как известно, данные могут передаваться в текстовом и бинарном виде. В каждом из этих

представлений

могут быть дополнительные особенности и усложнения (как в XML).

Другой стороной описания протокола передачи данных могут служить характеристики, описывающие возможности этого протокола. Например, возможность шифрования или валидации данных. Данные характеристики полезны при сравне-

нии нескольких протоколов данных для выбора оптимального по возможностям.

2.2.4 Характеристики данных (сообщений)

Характеристиками сообщений могут являться:

- разнородность;
- структурированность;
- связность;
- средний объём сообщения;
- средний объём единицы информации в сообщении.

2.3 Основные алгоритмы построения информационного взаимодействия

Для описания основных особенностей различных механизмов обмена выделим характеристики, по которым и будет проводиться идентификация видов инфор-

мационного обмена.

Для источника и приёмника данных:

- sd1 – вид источника/приёмника;
- sd2 – возможность работать в несколько потоков.

Для протокола:

- p1 – возможность шифрования;
- p2 – возможность подписи; – p3 – режим передачи сообщений.

Для сообщений:

- m1 – сложность сообщения;
- m2 – формат сериализации и десериализации; – m3 – кодировка.

2.3.1 Организация прямого обмена

Нет Без вложенности

База данных Б инарный В зависимости от СУБД

S, D =, P = Нет, M = В зависимости от СУБД

Да

Перед реализацией прямого обмена необходимо решить, какие СУБД будут

Лист

32

обмениваться информацией, так как не все системы могут взаимодействовать друг с другом. Например, СУБД Oracle и SQL Server могут выполнять запросы друг к другу, а вот SQL Server и MySQL – нет.

Затем необходимо выбрать вариант обмена: активный или пассивный и автоматический или автоматизированный. Об этом более подробно описано в п. 1.6.1.

Затем необходимо источнику и приёмнику данных договориться о форматах информационного взаимодействия. Это может быть описание физической модели данных, набора хранимых процедур (с описанием логики их работы но без конкретной релизации) или логической модели данных, если последнее допустимо и необходимо. В случае реляционных СУБД обычно создаётся отдельный набор таблиц для прямого обмена как у источника, так у приёмника.

Также допускается создание

представлений для обмена.

Дополнительно на этапе проектирования необходимо согласовать правила занесения данных в физическую модель данных прямого обмена. Этот шаг позволяет значительно сократить время на интеграционные испытания и решение конфликтных ситуаций, которые часто возникают при информационном обмене, спроекти-
рованном без участия всех заинтересованных сторон.

После согласования разработчики информационных систем приступают к реализации моделей данных и механизмов их заполнения, а также обмена данными между моделями. На стороне источника данных обычно создаются:

- физическая модель данных;
- механизм заполнения и актуализации модели прямого обмена;
- механизм проверки целостности данных в модели прямого обмена.

На стороне приёмника данных обычно создаются те же элементы совместно с:

- механизм контроля связи до источника данных;
- механизм обработки данных из модели прямого обмена;
- механизм проверки корректности данных, занесённый в систему из модели прямого обмена.

Дополнительно рекомендуется реализовать механизм кросс-контроля, когда источник данных может проверить собственные данные в системе-приёмнике, однако для этого необходимо либо сильно усложнять прямой обмен, либо использовать другие методы прямого обмена.

После реализации происходит процесс интеграционного тестирования, когда разработчики источника и приёмника совместными усилиями проверяют обмен и все связанные с ним механизмы в своих системах. Особое внимание на данном этапе стоит уделять корректности данных в модели прямого обмена между источником и приёмником. К примеру, необходимо удостовериться, что все поля из физической модели источника корректно разложились по соответствующим полям физической модели прямого обмена приёмника данных. Любые несоответствия должны быть разрешены. Например, одна система отсутствием данных считает пустую строку, а другая – специальное значение NULL. Такой случай необходимо согласовать отчётливо и задокументировать.

После проведения интеграционных испытаний и необходимых доработок источника и приёмника данных механизм переходит в промышленную эксплуатацию.

2.3.2 Обмен реестрами

Нет Без вложенности

ИС

S, D =, P = Нет, M = CSV,DBF

Нет Т екстовый или бинарный К ОИ-8,CP-1251,UTF-8

Алгоритм обмена реестрами похож на прямой обмен, однако в данном случае нет необходимости заботиться о совместимости хранилищ данных.

Однако, данный механизм содержит ряд особенностей, описанных ниже.

Лист

34

При согласовании форматов информационного обмена как правило последнее слово лежит за более эталонной системой вне зависимости от того, приёмником данных она является или же их источником. Например, если региональной ИС ЖКХ необходим адресный справочник адресов ФИАС, то формат взаимодействия задаётся источником данных (федеральной адресной системой), а приёмник данных – региональная система ЖКХ – должна строить приём данных на основании этих форматов. И обратный случай: региональная система как приёмник данных определяет форматы реестров для собственных и коммерческих систем, которые хотят размещать свои данные в региональной системе.

Описанный принцип монополии в определении форматов взаимодействия распространяется и на другие механизмы обмена. Исключением может быть разве что процесс прямого обмена между БД, так как такой метод обычно не используется

при построении обмена между системами, состоящими в эталонной иерархии на разных ступенях. Также не следует считать, что «подчинённая» информационная система не может как-то влиять на процесс и форматы информационного обмена. Это более чем реально при адекватности запросов, аргументированности просьб и соблюдении профессиональной этики.

Передпроектированием реализации обмена реестрами необходимо решить:

необходимо разрабатывать клиента для раскрытия или получения информации, или

же необходимо разработать сервер для получения данных.

При проектировании и разработке клиента необходимо грамотно прочитать документацию, в которой должен быть описан режим доступа к сервису обмена, режим обмена с ним и форматы информационного взаимодействия. Если клиент разрабатывается для раскрытия информации при помощи реестров, необходимо проверять данные на тестовых стендах сервера. Не допускается раскрывать не проверенную информацию на промышленных стендах, так как её корректировка и/или удаление может быть трудоёмким, особенно при обмене реестрами. Вне зависимости от направления потока информации, в конце проведения испытаний требуется процедура проверки выгруженных/полученных данных. По сравнению, например, с обменом по REST или SOAP, сделать это при обмене реестрами проблематично. Как правило, это выполняется в ручном режиме на этапе интеграционного тестирования.

Так как реестры обычно содержат большой массив информации, можно делать проверку получения/выгрузки данных контрольными значениями.

Например, точно известно, что в ФИАС имеется шестьдесят тысяч адресных объектов по

Ульяновской области. Соответственно, после загрузки этих объектов в региональную

ИС ЖКХ должно быть ровно шестьдесят тысяч адресных объектов по

Ульяновской области, у которых установлена связь с ФИАС. Если контрольное значение меньше, то какие-то данные были ошибочно отфильтрованы. Если контрольное значение больше, то какие-то данные были либо не удалены, либо задублированы в региональной ИС ЖКХ.

При проектировании и разработке сервера, принимающего реестры с данными от других информационных систем, необходимо в первую очередь задокументировать режим доступа, форматы взаимодействия, правила обработки записей и

механизмы проверки загружаемых данных.

В режиме доступа должно быть указано, как можно получить возможность загружать реестры в информационную систему. Это может быть либо письменное заявление, либо это право есть у пользователей ИС автоматически, либо любым иным способом. Идентификация, откуда приходят данные, очень важна для разрешения конфликтов по причине несогласованности данных, которые в любом случае могут возникнуть. К примеру, управляющая организация в реестре сведений о домах указала, что в доме 9 этажей, а ресурсоснабжающая организация указала для этого же дома 10 этажей. Обе организации имеют право размещать такие данные, так как у обеих есть договоры управления или поставки ресурсов на конкретный дом. В этом случае можно применить принцип эталонности данных. Он заключается в том, что если управляющая организация внесла данные об этажности, то ресурсоснабжающая

Лист
36
щая организация уже не в праве сменить такие данные, однако обратный процесс возможен. Конечно же, такие правила должны быть закреплены в документации. В форматах взаимодействия должно быть чётко описан формат данных (DBF, CSV CP1251, CSV UTF8 и т.д.), набор полей с указанием типа данных, описания, флага или условия обязательности, набора возможных значений (для

справочных полей).

Правила обработки записей должны включать исчерпывающие описания алгоритмов обработки строк реестров в зависимости от данных в этих строках. Эти сложности могут быть не совсем очевидны, если реестр загружает простой набор данных без выполнения каких-либо бизнес-процессов. Однако, если обработчик реестра должен после занесения данных в хранилище данных ещё выполнить набор бизнес-процессов (провести операцию оплаты, обновить процент заполнения паспорта ОЖФ, аннулировать договор управления), то эти

правила необходимы. При любом обмене данными следует учитывать, что выполнение неочевидных действий должно быть задокументировано.

Также при проектировании и разработке сервера обработки реестров необходимо предусмотреть механизм проверки загружаемых данных. Это может быть реализовано при помощи возвращения клиенту обработанного файла с реестром с дополненными системными столбцами (например, столбцы «Статус обработки записи», «Идентификатор записи», «Сообщение об ошибке обработки») или же возможностью просмотра ошибок обмена в интерфейсе ИС.

В любом случае, необходимо предусмотреть обратную связь сервера с клиентами.

2.3.3 Построение веб-сервисов SOAP и REST API

Для обмена при помощи протокола SOAP:

Да С вложенностью ИС
S, D =, P = Да, M = XML Да Т текстовый
UTF-8

Для архитектурного стиля REST: Нет С
вложенностью
ИС

S, D =, P = Да, M = JSON,XML
Да Т текстовый UTF-8

При построении веб-сервисов (не важно, SOAP это или REST API) необходимо-

мо в первую очередь определить следующее:

- технология обращения к веб-сервису;
- технология и правила авторизации и аутентификации;
- механизм поддержания версииности веб-сервиса;
- набор операций (методов) веб-сервиса;

– технология и процесс обмена бинарными файлами.

Технология обращения к веб-сервису определяется на основании целей и задач разработки веб-сервиса, а также разнородности информационных систем, кото-

рые будут с этим веб-сервисом общаться.

Подтверждение пользователя (или информационной системы) является важной частью любого механизма обмена. Поэтому необходимо изначально определиться с набором алгоритмов прохождения пользователем веб-сервиса авторизации и аутентификации. Стоит также отметить, что для разных технологий веб-сервисов обычно используются разные механизмы подтверждения клиентов. К примеру, в SOAP обычно используется подпись канала связи и/или бизнес-данных, а при построении REST API используется OAuth 2, JWT или OpenConnect.

Поддержание версииности веб-сервиса – основная проблема поддержки веб-сервисов. Версионность в данном случае понимается как возможность веб-сервиса

кущих. Выделяются следующие механизмы поддержания версииности:

- жёсткий. Клиенты со старыми форматами взаимодействия не допускаются к обмену информацией;
 - скрытый. Клиент не указывает или указывает один раз при авторизации версию своих форматов данных. Веб-сервис самостоятельно принимает решение об обработке таких данных.
 - ручной. Клиент в каждом запросе указывает версию форматов данных.
- Также адрес веб-сервиса может отличаться для каждой версии форматов данных.

Веб-сервис должен включать на каждый вид информации как минимум две операции: получение и размещение. Для REST API таких операций быть четыре: размещение разделяется на создание и изменение; добавляется удаление. Если какой-то тип операции отсутствует, это можно привести к неполноценному обмену: потеряется возможность или контролировать выгрузку данных к такому веб-сервису, или заносить данные в систему при помощи API. Поэтому на этапе проектирования веб-сервиса необходимо грамотно продумать все методы (операции) информационного обмена.

Если информационное взаимодействие предполагает обмен бинарными файлами (что, например, практически невозможно при прямом обмене или обмене реестрами), то необходимо на самом раннем этапе определиться с технологией передачи такого вида информации. Например, в SOAP передавать бинарные файлы слишком накладно (шифр base64 увеличивает объём передаваемой информации в 1,5 раза), и поэтому в некоторых случаях поднимается REST-сервис для работы с файловым хранилищем, а в SOAP используются только идентификаторы загруженных ранее файлов.

3 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МОДЕЛИ

3.1 Модуль интеграции РИАСЖКХ с ГИСЖКХ

3.1.1 Общее описание

В рамках работы был разработан модуль интеграции «РИАС ЖКХ. Модуль интеграции с ГИС ЖКХ».

Модуль позволяет проводить двусторонний обмен с государственной информационной системой ЖКХ следующими видами информации:

- договоры управления и уставы;
- договоры ресурсоснабжения;
- сведения о домах и помещениях;
- лицевые счета;
- приборы учёта и их показания;
- платёжные документы;
- факты оплат и отзыв платежей;
- перечни работ управляющих организаций;
- проверки ГЖИ и планы проверок.

Дополнительно в модуле интеграции с государственной системой ЖКХ реализована следующая функциональность:

- получение из ГИС ЖКХ реестра организаций согласно ЕГРЮЛ/ЕГРИП;
- получение из ГИС ЖКХ реестра временных адресов, отсутствующих в федеральной информационной адресной системе (ФИАС);
- получение из ГИС ЖКХ нормативно-справочной информации (НСИ);
- двусторонний обмен файлами.

Программное обеспечение использует следующие технологии информационного взаимодействия:

- SOAP по зашифрованному каналу связи с подписью бизнес-данных по протоколу xades-bes согласно ГОСТ 34.10-2012, описанному в [5];
- разбор xml-ответов от веб-сервиса с открытыми данными;
- обмен файлами по зашифрованному каналу связи по принципу REST API;
- парсинг csv-реестров, упакованных в архивах.

С программной точки зрения модуль интеграции РИАС ЖКХ с ГИС ЖКХ представляет службу ОС Windows. Общее количество значимых строк кода превышает 10 тысяч. При разработки модуля интеграции использовались следующие технологии и библиотеки:

- .NET Framework 4.5 ;
- язык программирования Visual C# 6 ;
- КriptoПРО .NET;
- Json.NET;
- NLog.

Модуль разделён на несколько составных частей:

- ядро обмена (AIS.HM.Integration.GIS.Core);
- служба windows (AIS.HM.Integration.GIS.Production);
- тестовый клиент интеграции (AIS.HM.Integration.GIS.Test);
- графический интерфейс для администратора системы и поставщиков информации (AIS.HM.UI.GIS).

Каждый компонент программного обеспечения использует модель данных РИАС ЖКХ. Здесь можно выделить основные сущности РИАС ЖКХ (Организация, Дом, Помещение) и специфичные для модуля (ГИС Запрос, ГИС Операция, ГИС ЛогПлатёжногоШлюза). Полное описание служебных сущностей логической (концептуальной) моде ли данных представлено в п. 3.1.6 в таблице 1.

Общая схема взаимодействия компонентов РАИС ЖКХ для организации обмена с ГИС ЖКХ представлена на рисунке 5.

Рисунок 5 – Общая схема взаимодействия компонентов РАИС ЖКХ для организации обмена с ГИС ЖКХ

3.1.2 Механизм информационного обмена

Основными принципами информационного обмена с ГИС ЖКХ являются:

- поддержание целостности данных, размещённых в РАИС ЖКХ и ГИС ЖКХ;
- максимально быстрая автоматическая выгрузка данных в ГИС ЖКХ;
- интеграция данных из смежных систем, развёрнутых совместно с РАИС ЖКХ, и дальнейшая передача их в ГИС ЖКХ;
- гарантия раскрытия информации в ГИС ЖКХ.

Согласно реализации информационного взаимодействия с ГИС ЖКХ,

описанному в п. 1.1, было принято весь процесс отправки сообщений в федеральную

систему разделить на следующие этапы:

а) для операции обмена (например, выгрузка договоров или приём домов)

определяется набор поставщиков информации, которые имеют право

фор-

мировать такие запросы;

б) для каждого поставщика информации формируется запросы:

определяется набор данных, подлежащих обмену, формируются

первичные XML-

представления запросов (untrustedxmlrequest), формируются пакеты в слу-

чае пакетной отправки данных;

в) в отдельном потоке для одиночных запросов или в текущем потоке для

пакетных запросов XML-представления подписываются по ГОСТ

34.102012 [5] и отправляются в веб-сервисы ГИС ЖКХ по

зашифрованному

каналу связи. GUID ответа (AckResult.MessageGUID) записывается в хра-

нилище данных РИАС ЖКХ;

г) в отдельном потоке идёт опрос результатов обработки запроса. Если

запрос был обработан, запускается необходимый обработчик, который

за-

висит от операции обмена и веб-сервиса.

Диаграмма последовательностей обработки запроса к ГИС ЖКХ представле-

на на рисунке 6.

Также была реализована поэтапная синхронизация поставщиков информации.

Это означает, что если поставщик информации передал в ГИС ЖКХ необходимые полномочия на раскрытие данных региональной информационной системе и разрешил в РИАС ЖКХ обмен, то в следующий сеанс обмена ничего выгружаться из РИАС ЖКХ в ГИС ЖКХ не будет. Сначала будут из ГИС ЖКХ приниматься дан-
ные по договорам (правоустанавливающие документы на раскрытие информации в разрезе домов), затем в РИАС ЖКХ будут импортированы данные о домах (операция hcs-house-management.ExportHouseData), затем о данных внутри домов (помещения, лицевые счета, приборы учёта и проч.). Если ошибок при приёме данных не возникло, статус интеграции поставщика информации сменится на «Синхронизация с ГИС ЖКХ завершена» и включатся операции выгрузки данных из РИАС ЖКХ в федеральную информационную систему.

Для уменьшения нагрузки на сервера формирование и отправка запросов происходит раз в сутки и запускается в 3 часа ночи по времени сервера. Опрос статуса ответа в ГИС ЖКХ проверяется раз в 5 минут. Обмен информацией о платежах их при наличии происходит раз в 5 минут.

Рисунок 6 – Диаграмма последовательностей обработки запроса к ГИС ЖКХ

3.1.3 Синхронизация данных

При получении данных из федеральной системы их необходимо корректно сопоставить с данными, которые уже имеются в РИАС ЖКХ. Например, при получении данных о доме, его подъездах и помещениях, существующие помещения по бизнес-ключу (в данном случае это флаг «Является жилым» и номер помещения) необходимо обновлять, новые создавать. Затем необходимо создать связь за-
писи в РИАС ЖКХ с аналогичной записи в ГИС ЖКХ. Обычно это делается че-
рез специальную таблицу fgis.GISEntityCompare, в которой каждой необходимой строке в хранилище данных РИАС ЖКХ присваивается ГУИД аналогичной записи в ГИС ЖКХ. Если в ГИС ЖКХ запись является версионной (а согласно целевой схеме ГИС ЖКХ практически все сущности должны быть версионны), то в fgis.GISEntityCompare записывается корневой ГУИД записи, а ГУИД версии с её
номером (при наличии) записывается в таблицу fgis.GISEntityVersion. Иногда ГУ-

Лист

44

ИД записи в ГИС ЖКХ сохраняется напрямую в сущности РИАС ЖКХ без
исполь-
зования fgis.GISEntityCompare. Например, так сделано в информации о платежах (или фактах оплат) жителями за ЖКХ. Это позволяет ускорить выборку данных и гарантировать целостность связи, но не позволяет учитывать удалённые данные из хранилища РИАС ЖКХ. Таким образом, в любой момент времени можно получить информацию о том, сколько данных каких типов в РИАС ЖКХ
синхронизировано
с государственной информационной системой ЖКХ.

Отдельной трудностью является синхронизация файлов с ГИС ЖКХ. Во-первых, файловый сервис ГИС ЖКХ или канал связи до него работает медленно как на скачивание, так и на загрузку файлов (обычно файловые сервисы работают медленно только на загрузку файлов). Во-вторых, технология передачи файлов отличается от передачи сообщений: вместо SOAP используется REST API. Изначально планировалось, что файлы будут передаваться прямо в SOAP-запросах [1]. Однако, данный подход является удобным только при выгрузке малого объёма файлов. В ГИС ЖКХ был реализован абсолютно иной подход, который технически является более совершенным, но накладывает некоторые сложности при его использовании. Согласно этому подходу файлами необходимо обмениваться через отдельный файловый сервис, построенный по принципам

REST API. Файл можно скачать и загрузить целиком, если его объём не превышает 5 Мб. В противном случае файл следует разбивать на части.

Однако, сложность заключается не в самом механизме обмена файлами, а в механизмах сопоставления файлов в ГИС ЖКХ и РИАС ЖКХ. Файлы в РИАС ЖКХ могут храниться в двух видах: поле FileName (имя файла) целевой сущности или через единое хранилище файлов (таблица po.cmn\$File). Любой из этих

механизмов предполагает отличающиеся друг от друга способы сопоставления. В случае с единым хранилищем файлов всё просто: создаётся расширение таблицы po.cmn\$File с ГУИД ГИС ЖКХ, и таким образом в системе возникает абсолютное сопоставление файла РИАС ЖКХ и ГИС ЖКХ. При замене файла у целевой сущности меняется ссылка на запись из po.cmn\$File и, соответственно, изменяется и ссылка на файл в ГИС ЖКХ. В случае же с хранением только имени файла в целевой сущности (а не ссылки на общее файловое хранилище), необходимо сопоставлять ГУИД файла ГИС ЖКХ напрямую к записи сущности (договора, дома и т.п.). Для этих целей существует аналог таблицы fgis.GISEntityCompare – fgis.GISFileCompare, в которой также есть кортеж полей, идентифицирующих запись в БД (SchemaName, TableName, LocalId). Сложность синхронизации файлов таким методом заключается в том, что необходимо не забывать при смене файла в записи целевой сущности очищать связь с файлом в fgis.GISFileCompare, что, несомненно, может привести к десинхронизации информации о прикреплённых файлах. Например, в РИАС ЖКХ файл заменили на новый без замены записи в fgis.GISFileCompare. В ГИС ЖКХ новый файл не передался (так как у файла есть ГУИД ГИС ЖКХ), и система считает, что всё работает в штатном режиме.

3.1.4 Пользовательский интерфейс

Нет сомнения в том, что автоматический обмен РИАС ЖКХ с ГИС ЖКХ должен контролироваться пользователями. Такая возможность должна быть доступна как администратору системы, так и администраторам организаций-поставщиков информации в ГИС ЖКХ.

Для пользователя РИАС ЖКХ доступны следующие виды контроля за инфор-
мационным обменом с ГИС ЖКХ:

- информационная панель обмена;
 - журнал обмена;
 - флаг «Синхронизирован с ГИС ЖКХ» в некоторых таблицах и фильтрах; – описание результата предпроверки данных для выгрузки в ГИС ЖКХ.
- Информационная панель обмена помогает контролировать ход интеграции с ГИС ЖКХ в разрезе организаций (для администратора системы) и домах, закреплённых за организацией (для всех администраторов). Примерный вид страниц со списками организаций и домов организации представлен на рисунках 7 и 8.

Журнал обмена представляет собой подробные технические сведения об обмене: сколько каких запросов было, какой ответ был получен. Также можно просмотреть тело запроса и ответа. Снимок экрана со списком запросов в журнале обмена с ГИС ЖКХ представлен на рисунке 9.

Флаг «Синхронизирован с ГИС ЖКХ» помогает быстро оценить, сколько информации было синхронизировано с ГИС ЖКХ. Его примерный вид можно видеть на рисунке 10.

Просмотр результатов предпроверки данных для синхронизации с ГИС ЖКХ бывает полезен, если требуется узнать, чего конкретно не хватает для

выгрузки какого-либо объекта в ГИС ЖКХ. Например, требуется узнать, какие данные отсутствуют для раскрытия информации о договоре управления. Для этого после включения необходимой настройки в конфигурации системы в подробностях каждого собственного договора управления появляются подсказки, что необходимо добавить или изменить, чтобы договор был размещён в ГИС ЖКХ. Пример такой подсказки можно видеть на рисунке 11.

3.1.5 Проблемы информационного взаимодействия

При построении информационного обмена с ГИС ЖКХ первой сложностью является проведение тестовых испытаний, по результатам которых региональную систему могут допустить к промышленной эксплуатации. Несмотря на наличие веб-интерфейса подачи заявок на такие испытания, потребуется сперва заполнить заявку на подключение системы к информационному взаимодействию с СИТ ГИС ЖКХ, а затем направить уведомление об окончании тестовых испытаний. Это может сделать только руководитель организации через форму обращения в техническую поддержку ГИС ЖКХ. После обработки уведомления специалистами технической поддержки, а затем и техническими специалистами, система допускается к промышленной эксплуатации.

Рисунок 7 – Информационная панель обмена с ГИС ЖКХ в разрезе организаций

Следующей трудностью является установка соединения с ГИС ЖКХ.

Как упоминалось ранее, существует два вида таких площадок: тестовые (СИТ ГИС ЖКХ) и промышленные (ППАК ГИС ЖКХ).

Доступ до промышленной площадки осуществляется по протоколу SSL с обязательным предоставлением клиентского сертификата по ГОСТ 34.10-2012.

Со стороны веб-сервисов ГИС ЖКХ также возвращается информация о сертификате по ГОСТ. Для поддержки такого типа сертификатов требуется установка дополнительного платного программного обеспечения (в данном случае был выбран КриптоПРО CSP). Дополнительно требуется установить сертификат удостоверяющего центра, выдавшего сертификат ППАК ГИС ЖКХ, в список доверенных корневых центров сертификации.

Доступ до тестовых площадок ГИС ЖКХ также может быть осуществлён по протоколу HTTPS/SSL, однако при обмене будут использоваться недостоверные сертификаты, так что проверку безопасности канала данных необходимо будет отключить. Альтернативой является HTTP-соединение, в котором следует в заголов-

Рисунок 8 – Информационная панель обмена с ГИС ЖКХ в разрезе домов
ках присылать отпечаток сертификата. Также при установке соединения с СИТ ГИС ЖКХ требуется указание простой электронной подписи при помощи basic-авторизации. Все эти особенности накладывают дополнительные расходы на реализацию соединения до СИТ ГИС ЖКХ, так как описанные выше требования не покрываются стандартными алгоритмами установки соединения для SOAP в современных языках программирования и фреймворках.

Следующей сложностью является процесс обновления ГИС ЖКХ.

Промышленный стенд ГИС ЖКХ может быть выключен либо для технического обслуживания, либо для установки новой версии без изменения форматов информационного взаимодействия, либо с их изменением.

Для первого и второго случая рассылка о том, что веб-сервисы будут отключены, приходит обычно от часа до двух до намеченного срока выключения площадки. Так как выключение происходит обычно в вечернее время, письмо об этом приходит после окончания рабочего дня разработчиков. Соответственно, смысл такой рассылки теряется.

При обновлении версии ППАК ГИС ЖКХ с изменением форматов

информационного обмена рассылка приходит заранее: примерно за неделю до планируемо-

Рисунок 9 – Список запросов в журнале обмена с ГИС ЖКХ

Рисунок 10 – Пример использования флага «Синхронизирован с ГИС ЖКХ» в интерфейсе РИАС ЖКХ

го обновления. Однако, нередко случаи, когда обновление откладывали на срок до месяца. В таком случае уже обновлённый модуль интеграции РИАС ЖКХ с ГИС ЖКХ откатывался до старых версий форматов обмена. Иногда стенд СИТ-02 ГИС ЖКХ, предназначенный для проверки новых версий форматов обмена, не обновлялся своевременно, что также может создавать трудности для проверки правильности интеграционного взаимодействия.

Следующую трудность информационного обмена, которую стоит отдельно выделить, является сложность форматов информационного обмена. Согласно целевой схеме ГИС ЖКХ, практически все виды информации являются версионными. То есть у каждой записи есть ГУИД корневой и ГУИД версии. Одни операции с

Лист
50

Рисунок 11 – Пример предпроверки договора управления

данными требуют корневой идентификатор, другие операции – версионный. Также часто в запросах на выгрузку данных смешиваются операции и смена состояния записей. К примеру, договор управления можно создать, обновить, расторгнуть, аннулировать, подать заявку на его одобрение, пролонгировать. Прибор учёта можно создать, обновить до ввода первого показания, обновить после ввода первого показания, заменить и архивировать. Таким образом, усложняется логика синхронизации данных с ГИС ЖКХ.

3.1.6 Описание модели данных

Описание основных сущностей представлено в таблице 1.

Название сущности	Описание
1	2
Запрос	Обращение к веб-сервисам ГИС ЖКХ
Сопоставление сущностей	Связь записей в РИАС ЖКХ и ГИС ЖКХ
Версия сущности	Сведения о версиях сущности в ГИС ЖКХ
Сопоставление файлов	Связь файлов РИАС ЖКХ и ГИС ЖКХ
Адресный объект	Сведения об адресном объекте адресного плана ГИС ЖКХ
Дом	Сведения о доме адресного плана ГИС ЖКХ
Поставщик информации	Сводная информация по синхронизации организации с ГИС ЖКХ
Объекты ПИ	Объекты поставщика информации
Версия ПИ	Версия поставщика информации в ГИС ЖКХ

Таблица 1 – Основные сущности логической модели данных модуля обмена РИАС ЖКХ с ГИС

ЖКХ
1 2

Справочник НСИ Нормативно-справочная информация ГИС ЖКХ
Элемент справочника НСИ Элемент нормативно-справочной информации ГИС ЖКХ
Сопоставление справочников Связь НСИ со справочниками РИАС ЖКХ
Веб-сервис Сведения о веб-сервисах ГИС ЖКХ
Операция веб-сервиса Сведения об операциях (методах) веб-сервисов ГИС ЖКХ
Группа операций Объединение операций для удобства интерфейса
Тип шаблона Сведения о шаблонах ГИС ЖКХ
Связь ТШ и ПИ Связь типа шаблона ГИС ЖКХ и поставщика информации
Связи служебных таблиц модуля интеграции РИАС ЖКХ с ГИС ЖКХ пред-
ставлены на рисунке 12.

3.1.7 Объём самостоятельной разработки

Описание объёма самостоятельной разработки данного программного

модуля представлено в таблице 2.

Таблица 2 – Объём самостоятельной разработки модуля обмена РИАС ЖКХ с ГИС ЖКХ

Деятельность/Часть модуля Объём самостоятельной разработки

1 2

Проектирование Самостоятельно

Обмен файлами Самостоятельно

Приём данных Под руководством

Отправка данных Под руководством

Хранилище данных Самостоятельно

Анализ и решение проблем Под руководством

3.2 Платёжный шлюз РИАСЖКХ

Платёжный шлюз представляет собой часть модуля интеграции РИАС ЖКХ

с ГИС ЖКХ для передачи платежей (фактов оплат) и операций над ними

посредством

информационного взаимодействия.

Шлюз состоит из нескольких частей:

– сервер для сбора данных о платежах и операциях над ними;

– часть windows-службы модуля интеграции, обрабатывающая пришедшие

Рисунок 12 – Связи служебных таблиц модуля интеграции РИАС ЖКХ с ГИС ЖКХ

данные, отправляющая пакеты платежей в ГИС ЖКХ и обрабатывающая

результаты.

Сервер сбора данных представляет собой ASP.NET-приложение. Оно

написано с использованием технологии WebAPI. Для доступа к данным

используется Entity Framework 6. При построении интерфейсов взаимодействия

использует ар-

хитектурный стиль REST.

Сервер сбора данных реализует следующие операции:

– зарегистрировать платёж в собственную кассу;

– зарегистрировать платёж агента;

– аннулировать платёж;

– проверить состояние запроса.

Дополнительно для каждой операции существует групповая операция, пол-

ностью дублирующая функциональность единичного метода.

Для доступа к API следует удостовериться, что для конкретной развёртки

РИАС ЖКХ настроен сервер сбора данных о платежах и связь с ГИС ЖКХ. При

отсутствии настройки такого сервера она может быть осуществлена отделом

поддержки Исполнителя или Заказчиком самостоятельно. Далее необходимо

передать права на раскрытие информации в ГИС ЖКХ для той информационной

системы, с которой установлена связь в РИАС ЖКХ. Если система не является

муниципальной или региональной, потребуется подтверждение права передачи

информации от владельца ИС в ГИС ЖКХ. После этого следует получить ГУИД

клиента API РИАС ЖКХ. Его

необходимо будет прикреплять ко всем запросам API.

Документация, ключи и конечные точки веб-сервисов регистрации платежей

находятся в закрытом доступе.

На текущий момент существует ряд реализаций клиентов для сервера сбора

данных о платежах: тестовый клиент от разработчиков сервера, две разработки

команд компании «АИС Город», как минимум одна реализация от сторонних

разра-

ботчиков.

Снимок экрана с журналом приёма информации о платежах представлена на

рисунке 13.

Для ускорения регистрации платежей веб-сервисом было принято решение

разделить факты оплат по нескольким слабо связанным друг с другом таблицам. В

одной таблице (fgis.GISPaymentServiceLog) содержится входящая информация от веб-сервиса. Затем оттуда windows-служба обмена при обработке платежей при помощи специальных хранимых процедур обрабатывает и заносит данные в таблицы финансовой подсистемы РИАС ЖКХ (fin.PaymentOrder и fin.RecipientInfo). Обмен с ГИС ЖКХ происходит уже со вторым наоборот таблиц. Таким образом, удалось отделить фронт платёжного шлюза от обработки пришедших данных, тем самым повысив отказоустойчивость системы.

Диаграмма деятельности по регистрации платежа в ГИС ЖКХ через РИАС ЖКХ представлена на рисунке 14.

3.2.1 Описание модели данных

Описание сущностей представлено в таблице 3.

Рисунок 13 – Снимок экрана с журналом приёма информации о платежах РИАС ЖКХ

Таблица 3 – Сущности логической модели данных платёжного шлюза РИАС ЖКХ

Название сущности		Описание
1	2	
Лог платёжного шлюза	Информация, зарегистрированная платёжным шлюзом	
Факт оплаты	Совершённый платёж за услуги ЖКХ	
Платёжные реквизиты	Платёжные реквизиты получателя платежа	
Атрибуты сущности [Лог платёжного шлюза] представлены в таблице 4.		
Таблица 4 – Атрибуты сущности [Лог платёжного шлюза] и поля таблицы		
{fgis.GISPaymentServiceLog}		
Название атрибута/поля		Тип Описание
1	2	3
Идентификатор записи лога		
{Id }		
Числовой		
{int }		
-		
Название метода		
{MethodName }		
Текстовый		
{nvarchar(50) }		
-		
Идентификатор клиента API		
{ApiClientId }		
Числовой		
{int }		
-		
Состояние записи		
{StatusId }		
Числовой		
{int }		
-		
Идентификатор платежа		
{PaymentOrderId }		
Числовой		
{int }		
-		
Идентификатор запроса ГИС ЖКХ		
{GisRequestId }		
Числовой		
{int }		
-		
Подробности состояния		
StatusDescription		

Текстовый	
nvarchar(max)	
Заполняются при ошибке	
{ } { }	
1 2 3	
ОГРН поставщика данных	
{DataProviderOgrn}	
Текстовый	
{nvarchar(max)}	
-	
Номер платежа	
{OrderNumber }	
Текстовый	
{nvarchar(64) }	
Уникальный номер из биллинговой системы	
Дата платежа	
{OrderDate }	
Дата и время	
{datetime2(0) }	
-	
Величина платежа	
{OrderAmount }	
Двойной числовой	
{bigint }	
В копейках	
Лицевой счёт	
{OrderAccount }	
Текстовый	
{nvarchar(max) }	
-	
Период оплаты	
{OrderPeriod }	
Дата	
{date }	
Первый день месяца	
Назначение платежа	
{OrderPurpose }	
Текстовый	
{nvarchar(1000) }	
-	
Произвольный комментарий	
{OrderComment }	
Текстовый	
{nvarchar(210) }	
-	
Наименование исполнителя	
{ExecutorName }	
Текстовый	
{nvarchar(200) }	
-	
Исполнитель ИП	
{ExecutorIsIP }	
Логический	
{bit }	
-	

ИНН исполнителя
{ExecutorInn }
Текстовый
{nvarchar(50) }
-
КПП исполнителя
{ExecutorKpp }
Текстовый
{nvarchar(50) }
Обязательно, если
исполнитель не ИП
Наименование получателя
{RecipientName }
Текстовый
{nvarchar(200) }
-
ИНН получателя
{RecipientInn }
Текстовый
{nvarchar(50) }
-
БИК получателя
{RecipientBik }
Текстовый
{nvarchar(50) }
-
Банк получателя
{RecipientBank }
Текстовый
{nvarchar(200) }
-
Расчётный счёт получателя
{RecipientAccount }
Текстовый
{nvarchar(50) }
Только цифры
Наименование плательщика
{PayerName }
Текстовый
{nvarchar(160) }
-
Плательщик ЮЛ
{PayerIsLegal }
Логический
{bit }
-
ОГРН владельца ЛС
{AccountOwnerOrganizationOgrn }
Текстовый
{nvarchar(max) }
Заполняется только для
метода SendOrderInfo
ИД ЖКУ ГИС ЖКХ
{AccountGisServiceId }
Текстовый
{nvarchar(max) }
-

Дата аннулирования платежа

{CancellationDate }

Дата и время

{datetime2(0) }

-

Причина аннулирования платежа

{CancellationComment }

Текстовый

{nvarchar(max) }

Обязательно, если указана

дата аннулирования

Дата и время создания записи

Created

Дата и время

datetimeoffset(2)

Включает часовой пояс

{ } { }

Лист

56

1 2 3

Дата и время обновления записи Дата и время Включает часовой пояс

{Updated} {datetimeoffset(2)}

Атрибут «Название метода» может содержать следующие значения:

– SendOrderInfo — размещение платежа в собственную кассу; –

SendOrderInfoAgent — размещение платежа через платёжного агента; –

SendOrderCancellation — аннулирование платежа.

Атрибут «Состояние записи» может содержать следующие значения:

– new – ожидает обработки;

– success – обработан успешно;

– error.service – ошибка веб-сервиса;

– error.rias – ошибка РИАС ЖКХ;

– error.gis – ошибка в ГИС ЖКХ;

– wait.rias – ожидает отправки в ГИС ЖКХ;

– wait.gis – ожидает ответа от ГИС ЖКХ;

– overtime – истекло время обработки.

Атрибуты сущности [Факт оплаты] представлены в таблице 5.

Таблица 5 – Атрибуты сущности [Факт оплаты] и поля таблицы {fin.PaymentOrder}

Название атрибута/поля Тип Описание

1 2 3

Идентификатор оплаты

{Id }

Числовой

{int }

-

Идентификатор платёжных реквизитов

{RecipientInfold }

Числовой

{int }

Ссылка на сущность

[Платёжные реквизиты]

Идентификатор поставщика данных

{DataProviderOrganizationId }

Числовой

{int }

Ссылка на базовый

справочник РИАС ЖКХ

Номер платежа
{OrderNumber }
Текстовый
{nvarchar(64) }
Уникальный номер из биллинговой системы
Дата платежа
{OrderDate }
Дата и время
{datetimeoffset(0) }
С часовым поясом
Период оплаты
{OrderPeriod }
Дата
{date }
Первый день месяца
Величина платежа
{OrderAmount }
Двойной числовой
{bigint }
В копейках
Назначение платежа
OrderPurpose
Текстовый
nvarchar(1000)
-
{ } { }
1 2 3
Произвольный комментарий
{OrderComment}
Текстовый
{nvarchar(210)}
-
ИД ЖКУ ГИС ЖКХ
{AccountGisServiceId }
Текстовый
{nvarchar(50) }
-
Лицевой счёт
{AccountNumber }
Текстовый
{nvarchar(50) }
-
Наименование исполнителя
{ExecutorName }
Текстовый
{nvarchar(200) }
-
Исполнитель ИП
{ExecutorIsIP }
Логический
{bit }
-
ИНН исполнителя
{ExecutorInn }
Текстовый
{nvarchar(50) }

-
КПП исполнителя
{ExecutorKpp }
Текстовый
{nvarchar(50) }
Обязательно, если
исполнитель не ИП
Наименование плательщика
{PayerName }
Текстовый
{nvarchar(160) }

-
Плательщик ЮЛ
{PayerIsLegal }
Логический
{bit }

-
Платёж отменён
{IsCancelled }
Логический
{bit }

-
Дата аннулирования платежа
{CancellationDate }
Дата и время
{datetime2(0) }

-
Причина аннулирования платежа
{CancellationComment }
Текстовый
{nvarchar(max) }
Обязательно, если указана
дата аннулирования
Транспортный ГУИД ГИС ЖКХ
{GisTransportGuid }
ГУИД
{uniqueidentifier }

-
Номер платежа в ГИС ЖКХ
{GisOrderNumber }
Текстовый
{nvarchar(50) }

-
ГУИД платежа в ГИС ЖКХ
{GisOrderGuid }
ГУИД
{uniqueidentifier }

-
Платёж отменён в ГИС ЖКХ
{GisIsCancelled }
Логический
{bit }

-
Дата и время создания записи
{Created }
Дата и время
{datetimeoffset(2) }

Включает часовой пояс

Дата и время обновления записи

Updated

Дата и время

datetimeoffset(2)

Включает часовой пояс

{ } { }

Атрибуты сущности [Платёжные реквизиты] представлены в таблице 6.

Таблица 6 – Атрибуты сущности [Платёжные реквизиты] и поля таблицы {fin.RecipientInfo}

Название атрибута/поля Тип Описание

1 2 3

Идентификатор платёжных реквизитов

{Id }

Числовой

{int }

-

БИК получателя

RecipientBik

Текстовый

nvarchar(9)

-

{ } { }

Лист

58

1 2 3

Банк получателя

{RecipientBank}

Текстовый

{nvarchar(255)}

-

Наименование получателя

{RecipientName }

Текстовый

{nvarchar(255) }

-

ИНН получателя

{RecipientInn }

Текстовый

{nvarchar(20) }

-

КПП получателя

{RecipientKpp }

Текстовый

{nvarchar(9) }

-

Расчётный счёт получателя

{RecipientAccount }

Текстовый

{nvarchar(20) }

Только цифры

Корреспондентский счёт

{CorrespondentBankAccountNumber }

Текстовый

{nvarchar(20) }

Только цифры

Дата и время создания записи

{Created }

Дата и время

{datetimeoffset(2) }

Включает часовой пояс

Дата и время обновления записи

Updated

Дата и время

datetimeoffset(2)

Включает часовой пояс

{ } { }

3.2.2 Объём самостоятельной разработки

Описание объёма самостоятельной разработки данного программного модуля представлено в таблице 7.
Таблица 7 – Объём самостоятельной разработки платёжного шлюза РИАС ЖКХ

Деятельность/Часть модуля Объём самостоятельной разработки

1 2

Проектирование Самостоятельно

Развёртка конечных точек Самостоятельно

Разработка методов API Самостоятельно

Хранилище данных Самостоятельно

Обработка входящих данных Самостоятельно

Отправка данных в ГИС ЖКХ Самостоятельно

3.3 Модуль интеграции РИАСЖКХ с «АИС Город.

Система начислений»

Интеграция РИАС ЖКХ с программным продуктом «АИС Город. Система начислений» (далее – Система начислений) является примером прямого обмена меж-

Рисунок 14 – Диаграмма деятельности по регистрации платежа в ГИС ЖКХ через РИАС ЖКХ
ду БД. Этот метод информационного обмена применим в первую очередь потому, что и источник, и приёмник данных разрабатываются в одной организации. К тому же, взаимодействие происходит только в одну сторону (из Системы начислений в РИАС ЖКХ), что также допускается прямым обменом между БД. Схематично процесс прямого обмена РИАС ЖКХ с Системой начислений представлен на рисунке 15.

В Системе начислений реализован набор таблиц и открыт к ним доступ для хранилища данных РИАС ЖКХ. В РИАС ЖКХ создан такой же набор таблиц с до-полнительными столбцами.

Лист

60

Рисунок 15 – Схематичное представление процесса прямого обмена с Системой начислений
Нижеследующие таблицы для прямого обмена (для удобства на-звания таблиц были упрощены):

- ext.Organizations – справочник организаций;
- ext.Services – справочник групп услуг;
- ext.Addresses – справочник адресов (адресный план);
- ext.Apartments – помещения и комнаты;
- ext.Accounts – лицевые счета;
- ext.MeteringDevices – приборы учёта;
- ext.PaymentDocuments – платёжные документы.

Существует ряд зависимых от представленных таблиц, которые для текущего описания не являются ключевыми.

В каждой таблице для прямого обмена должны быть определены поля:

- Remoteld – идентификатор записи во внешней системе;
- Created – дата и время создания записи;
- Updated – дата и время обновления записи;
- IsDeleted – флаг «Запись является удалённой».

В региональной системе ЖКХ к столбцам в обменных таблицах

дополнитель-

но добавляются следующие:

- Id – идентификатор записи в РИАС ЖКХ;
- SystemId – идентификатор системы, откуда пришли данные;
- CheckMessage – текст ошибки проверки данных перед их синхронизацией;
- IsProcessed – флаг «Запись обработана»;
- набор иных идентификаторов РИАС ЖКХ.

Обмен данными с точки зрения источника данных (Системы начислений)

вы-

глядит следующим образом:

- а) персонал даёт команду системе сформировать данные для РИАС ЖКХ;
- б) специальный механизм обновляет и дополняет записи в обменных таблицах;
- в) персонал вручную проверяет корректность заполненных данных в РИАС ЖКХ.

С точки зрения приёмника данных обмен выглядит следующим образом:

- а) каждую ночь (либо по запросу Заказчика) выполняется сканирование обменных таблиц связанных систем;
- б) если какие-либо изменения были обранужены, они при помощи receiveпроцедур загружаются в РИАС ЖКХ. Соответствующая запись оставля-
- ется в логе обмена;
- в) после получения всех обновлённых сведений для каждого вида информации запускаются read-процедуры, которые синхронизируют данные обменный таблиц с хранилищем данных РИАС ЖКХ. Запись об этом остав-
- ляется в логе обмена;
- г) персонал по записям лога проверяет корректность прохождения процесса

получения данных из обменных таблиц.

Не все данные сразу могут быть добавлены в РИАС ЖКХ. К примеру, частая ошибка обмена – несоответствие адресных планов РИАС ЖКХ и Системы начислений. Адресный план РИАС ЖКХ содержит ФИАС, временные адреса ГИС ЖКХ и вручную добавленная адреса через саму систему. Адресный план Системы начислений сильно зависит от конкретной развёртки этой системы.

Соответственно, для этих целей в РИАС ЖКХ существует механизм сопоставления адресов. Он представляет собой связь записей адресных планов внешних систем или реестров с адресным планом РИАС ЖКХ. Если адреса отличаются так, что система сама не может принять решение о сопоставлении записей, она предоставляет это решение экспертам. Чаще всего в роли экспертов выступают представители Заказчика, потому что именно они заинтересованы в корректности передаваемой информации.

3.3.1 Объём самостоятельной разработки

Описание объёма самостоятельной разработки данного программного модуля представлено в таблице 8.

Таблица 8 – Объём самостоятельной разработки прямого обмена РИАС ЖКХ с Системой начислений

Деятельность/Часть модуля Объём самостоятельной разработки

1 2

Проектирование Принимал участие

Обмен реестрами Под руководством

Сопоставление адресов Не принимал участие

Проверка приходящих данных Самостоятельно

Настройка связи между СУБД Самостоятельно

Интерфейс Принимал участие

3.4 Подсистема обработки реестров РИАСЖКХ

Наряду с механизмами обмена при помощи API и заполнением данных при

помощи интерфейса пользователя в РИАС ЖКХ существует метод массового за-
полнения данных через реестры.

Реестры с точки зрения РИАС ЖКХ могут быть файлы следующих типов:

– DBF-файл в кодировке КОИ-8;

– CSV-файл в кодировке Win-1251 с разделителем «;» с наименованием
столбцов в первой строке;

– CSV-файл в кодировке Win-1251 с разделителем «;» только с данными.

Реестр в РИАС ЖКХ описывается при помощи основной информации о ре-
естре, списка столбцов и хранимой процедуры-обработчика реестра. Более
подробное описание основных сущностей по работе с реестрами можно найти в п.

3.4.1 и

таблице 9.

Для успешного заполнения реестра РИАС ЖКХ позволяет выгрузить его

шаблон. Можно выгрузить два типа шаблонов: CSV-файл и Excel-файл.

Последний отличается наличием у каждого столбца описания, флага
обязательности и списка возможных значений (если поле справочное). Вид
одного из шаблонов в формате Excel
представлен на рисунке 16.

Бывают случаи, когда поставщики информации (любые участники рынка ЖКХ
или Заказчик) не могут предоставить справочные данные в необходимом виде.

Например, в РИАС ЖКХ поле «Тип помещения» может принимать значение
«Комната в ИЖД», а во внешней системе этот же элемент может называться
«Комната в жилом доме». Для сопоставления неопределённых значений
справочных полей был разработан механизм сопоставления справочников. Суть
его работы схожа с сопоставлением адресов, который применяется для прямого
обмена с Системой начис-

лений. Интерфейс по сопоставлению справочников представлен на рисунке 17.

3.4.1 Описание модели данных

Описание сущностей представлено в таблице 9.

Название сущности Описание

1 2

Тип реестра Сведения о видах реестров, доступных в РИАС ЖКХ

1 2

Поля типа реестра Набор полей у типов реестров

Реестр Сведения о загруженных реестрах в систему

Строка реестра Записи о строках реестра

Сопоставление справочников Сведения о сопоставлении значений справочных полей
для каждой организации

Таблица 9 – Сущности логической модели данных подсистемы обмена реестрами РИАС ЖКХ

Атрибуты сущности [Тип реестра] представлены в таблице 10.

Таблица 10 – Атрибуты сущности [Тип реестра] и поля таблицы {ree.ReesterType}

Название атрибута/поля Тип Описание

1 2 3

Идентификатор типа реестра

{Id }

Числовой

{int }

-

Код

{Code }

Текстовый

{nvarchar(50) }

-

Наименование

{Name }
Текстовый
{nvarchar(100) }
-
Есть расширенный обработчик
{IsCustomProcessor }
Логический
{bit }
Реестр обрабатывается не
построчно

Название обработчика
{ProcessProcName }
Текстовый
{nvarchar(100) }
Хранимая процедура,
обрабатывающая реестр
Описание
{Description }
Текстовый
{nvarchar(1000) }
-
Является архивным
{IsArchived }
Логический
{bit }
-
Дата и время создания записи
{Created }
Дата и время
{datetimeoffset(2) }
Включает часовой пояс
Дата и время обновления записи
Updated
Дата и время
datetimeoffset(2)
Включает часовой пояс
{ } { }

Атрибуты сущности [Поля типа реестра] представлены в таблице 11.
Таблица 11 – Атрибуты сущности [Поля типа реестра] и поля таблицы {ree.ReesterField}

Название атрибута/поля	Тип	Описание
1	2	3
Идентификатор поля		
{Id }		
Числовой		
{int }		
-		
Идентификатор типа реестра		
{TypeId }		
Числовой		
{int }		
-		
Номер		
{Num }		
Числовой		
{int }		
-		
Является обязательным		

{IsRequired }

Логический

{bit }

-

Является информационным

IsInformational

Логический

bit

-

{ } { }

1 2 3

Код

{FieldName }

Текстовый

{nvarchar(256) }

-

Название

{FieldDisplayName }

Текстовый

{nvarchar(250) }

-

Описание

{FieldDescription }

Текстовый

{nvarchar(500) }

-

Тип данных

{DataType }

Текстовый

{nvarchar(50) }

В типах SQL Server 2008+

Набор возможных значений

{ReferenceViewName }

Текстовый

{nvarchar(100) }

Специальное представление

в

БД

Дата и время создания записи

{Created }

Дата и время

{datetimeoffset(2) }

Включает часовой пояс

Дата и время обновления записи

Updated

Дата и время

datetimeoffset(2)

Включает часовой пояс

{ } { }

Атрибуты сущности [Реестр] представлены в таблице 12.

Таблица 12 – Атрибуты сущности [Реестр] и поля таблицы {ree.Reester}

Название атрибута/поля Тип Описание

1 2 3

Идентификатор реестра

{Id }

Числовой

{int }
-
Идентификатор типа реестра
{TypeId }
Числовой
{int }
-
Идентификатор поставщика информации
{OrganizationId }
Числовой
{int }
-
Идентификатор состояния
{StatusId }
Числовой
{int }
-
Имя файла
{FileName }
Текстовый
{nvarchar(500) }
-
Период
{Period }
Дата
{date }
Первый день месяца
Число записей
{TotalRowNumber }
Длинное целое
{bigint }
-
Обрабатывается
{IsProcessing }
Логический
{bit }
-
Проверяется
{IsChecking }
Логический
{bit }
-
Начало обработки
{ProcessStart }
Дата и время
{datetimeoffset(2) }
Включает часовой пояс
Окончание обработки
{ProcessEnd }
Дата и время
{datetimeoffset(2) }
Включает часовой пояс
Дата и время создания записи
{Created }
Дата и время
{datetimeoffset(2) }

Включает часовой пояс

Дата и время обновления записи

Updated

Дата и время

datetimeoffset(2)

Включает часовой пояс

{ } { }

Рисунок 16 – Вид шаблона реестра Универсальный.7 РИАС ЖКХ в Excel

Рисунок 17 – Интерфейс по сопоставлению справочников РИАС ЖКХ

Атрибуты сущности [Строка реестра] представлены в таблице 13.

Таблица 13 – Атрибуты сущности [Строка реестра] и поля таблицы {ree.ReesterRecord}

Название атрибута/поля Тип Описание

1 2 3

Идентификатор строки реестра

{Id }

Числовой

{int }

-

Идентификатор реестра

{ReesterId }

Числовой

{int }

-

Значения столбцов

{Fields }

Текстовый

{nvarchar(max) }

-

Идентификатор состояния

{StatusId }

Числовой

{int }

-

Ошибка проверки

{CheckErrorMessage }

Текстовый

{nvarchar(max) }

-

Ошибка обработки

{ProcessErrorMessage }

Текстовый

{nvarchar(max) }

-

Комментарий

CommentMessage

Текстовый

nvarchar(max)

-

{ } { }

Лист

67

1 2 3

Идентификатор данных

{DataId }

Текстовый

{nvarchar(100) }

- Дата и время создания записи

{Created }

Дата и время

{datetimeoffset(2) }

Включает часовой пояс

Дата и время обновления записи

Updated

Дата и время

datetimeoffset(2)

Включает часовой пояс

{ } { }

Атрибуты сущности [Сопоставление справочников] представлены в таблице

14.

Таблица 14 – Атрибуты сущности [Сопоставление справочников] и поля таблицы {ree.Reference}

Название атрибута/поля Тип Описание

1 2 3

Идентификатор сопоставления

{Id }

Числовой

{int }

-

Идентификатор поля типа реестра

{FieldId }

Числовой

{int }

-

Сопоставляемое значение

{Value }

Текстовый

{nvarchar(1000) }

-

Значение сопоставлено

{IsMapped }

Логический

{bit }

-

Идентификатор сопоставленной записи

{ReferenceValueId }

Числовой

{int }

-

Ошибочное значение

{IsError }

Логический

{bit }

-

Сопоставленное значение

{ReferenceValue }

Текстовый

{nvarchar(200) }

-

Идентификатор поставщика

информации

{DataProviderOrgId }

Числовой

{int }

-	
Дата и время создания записи {	
Created }	
Дата и время	
{datetimeoffset(2) }	
Включает часовой пояс	
Дата и время обновления записи	
Updated	
Дата и время	
datetimeoffset(2)	
Включает часовой пояс	
{ } { }	

3.4.2 Объём самостоятельной разработки

Описание объёма самостоятельной разработки данного программного модуля представлено в таблице 15.

Деятельность/Часть модуля
Объём самостоятельной разработки

1	2
Проектирование	Не принимал участия
Написание обработчиков реестров	Принимал участие
Сопоставление справочников	Принимал участие
Автоматизация обработки	Под руководством
Интерфейс	Принимал участие
Таблица 15 – Объём самостоятельной разработки подсистемы обмена реестрами РИАС ЖКХ	

4 ОПИСАНИЕ ЭКСПЕРИМЕНТОВ

4.1 Нагрузочное тестирование веб-сервиса платежей ГИС

ЖКХ
Цели и задачи эксперимента
Целью данного вычислительного эксперимента является проверка корректности работы веб-сервиса регистрации платежей (фактов оплат) ГИС ЖКХ в момент пиковых нагрузок.
Данная цель достигается при помощи следующих задач:
– внедрение механизма мониторинга времени обработки сообщений в ГИС ЖКХ для платёжного веб-сервиса;
– получение от заказчиков контрольных значений по пиковым нагрузкам их биллинговых систем;
– получение платежей (фактов оплат) в РИАС ЖКХ и их последующая отправка в ГИС ЖКХ в момент пиковой нагрузки;
– сбор и обработка полученных временных интервалов обработки пакетов фактов оплат и аппроксимация результатов для большего числа платежей.
Исходные данные
Время и продолжительность пика оплат за ЖКХ с 10:00 до 14:00. В это время происходит свыше половины фактов оплат. Каждую минуту от заказчика приходит в среднем около 120 платежей. Наибольший поток платежей за 2017 год был зарегистрирован 27 февраля 2017 года в 14:03 и составил 533 платежа в минуту.
По требованию приказа No 74/114/пр [12] и технического задания на РИАС ЖКХ время отправки факта оплаты от его совершения до конца его регистрации в ГИС ЖКХ должно составлять 2 часа.
Необходимо узнать, будет ли превышен указанный интервал в конце пиковой нагрузки.
План эксперимента
а) сбор данных о пиковых нагрузках. На этом этапе необходимо выяснить у заказчика (или разработчика его биллинговой системы), в какие промежутки времени происходит самый большой поток оплат за услуги ЖКХ;

- б) разработка и внедрение механизма мониторинга времени обработки запроса в ГИС ЖКХ;
- в) получение фактов оплат от заказчика при пиковой нагрузке;
- г) регистрация фактов оплат в ГИС ЖКХ. На данном шаге следует отметить, что факты оплат в ГИС ЖКХ регистрируются пакетами от 1 до 1000 штук. Здесь следует отправить пакеты по 1000 платежей (пик), а также и более мелкие пакеты (полупик) для выявления зависимости числа времени обработки пакета от числа платежей в нём;
- д) моделирование и аппроксимация полученных задержек регистрации платежей на весь период пика. На данном шаге вычисляется максимальная задержка платежа под конец пиковой нагрузки;
- е) составление выводов о корректности работы веб-сервиса ГИС ЖКХ.

Выполнение эксперимента

Платежи были перенесены в очередь передачи в ГИС ЖКХ и сохранены в хранилище данных РИАС ЖКХ. Дальнейшая обработка данных и их передача в ГИС ЖКХ проводилась в автоматическом режиме модулем интеграции РИАС ЖКХ с ГИС ЖКХ, как если бы эти данные приходили в платёжный шлюз от систем заказчика в обычном режиме работы.

Платежи отправлялись в ГИС ЖКХ порциями от 1 до 1000 записей за раз. Результаты передачи данных и время реагирования ГИС ЖКХ приведены в таблице 16.

Таблица 16 – Результаты передачи данных и время реагирования ГИС ЖКХ

Количество платежей			
в запросе			
Дата и время			
Запроса Ответа Дельта, сек			
1	2	3	4
21	2017-05-28 20:15:22	2017-05-28 20:15:23	1
60	2017-05-28 17:55:34	2017-05-28 17:55:37	3
71	2017-05-28 17:18:00	2017-05-28 17:18:04	4
88	2017-05-28 14:04:12	2017-05-28 14:04:18	6
148	2017-05-27 19:33:13	2017-05-27 19:33:17	3
221	2017-05-27 18:39:00	2017-05-27 18:39:06	6
265	2017-05-27 18:28:06	2017-05-27 18:28:15	9
317	2017-05-27 17:06:29	2017-05-27 17:06:43	14
362	2017-05-27 17:28:15	2017-05-27 17:28:26	11
416	2017-05-27 15:55:45	2017-05-27 15:56:07	22
485	2017-05-27 15:06:35	2017-05-27 15:06:52	17
511	2017-05-27 13:39:14	2017-05-27 13:39:38	24
566	2017-05-26 17:26:26	2017-05-26 17:26:43	17
611	2017-05-26 12:53:16	2017-05-26 12:53:43	27
676	2017-05-26 12:31:22	2017-05-26 12:31:58	36
727	2017-05-25 10:53:13	2017-05-25 10:53:48	35
784	2017-05-24 18:00:57	2017-05-24 18:01:38	41
838	2017-05-24 19:06:48	2017-05-24 19:07:35	47
897	2017-05-24 19:28:44	2017-05-24 19:29:26	42
985	2017-05-24 19:50:42	2017-05-24 19:51:20	38
1000	2017-05-26 07:32:00	2017-05-26 07:32:49	49

График зависимости времени обработки пакета платежей от количества записей в пакете изображён на рисунке 18.

Ожидаемые и полученные результаты

Ожидается, что веб-сервис ГИС ЖКХ будет обрабатывать более 1000 платежей в минуту. Скорость обработки пакета платежей должна линейно

возрастать по

отношению к количеству фактов оплат в пакете.

По результатам вычислительного эксперимента скорость обработки

платежей в ГИС ЖКХ составила в среднем 1466 платежей (фактов оплат) в

минуту. Возраста-

Рисунок 18 – График зависимости времени обработки пакета платежей от количества записей в

пакете с линией линейного тренда

ние скорости обработки в зависимости от элементов в пакете проявляет линейную

зависимость.

Выводы

ГИС ЖКХ регистрирует платежи со скоростью, позволяющей иметь запас

по пиковым нагрузкам заказчика более чем в 2 раза. Линейная зависимость при

формировании пакетов данных свидетельствует о том, что формирование крупных

пакетов (с количеством платежей около предельных значений) выгодно только

благодаря накладным расходам на их формирование в РИАС ЖКХ и благодаря

затратам

на установку зашифрованного канала связи между РИАС ЖКХ и ГИС ЖКХ.

Лист

73

4.2 Нагрузочное тестирование веб-сервиса платежей

РИАСЖКХ

Цели и задачи эксперимента

Данный эксперимент связан с экспериментом по нагрузочному

тестированию

веб-сервиса платежей ГИС ЖКХ, описанном в п. 4.1.

Целью данного эксперимента является проверка корректности веб-сервиса

РИАС ЖКХ по регистрации фактов оплат от биллинговых систем заказчиков (т.н.

«платёжный шлюз») и подсистемы обработки платежей РИАС ЖКХ для их даль-

нейшей отправки в ГИС ЖКХ.

Задачи эксперимента:

– сбор сведений о предельной скорости регистрации фактов оплат в РИАС

ЖКХ;

– сбор сведений о скорости обработки фактов оплат для их последующей от-

правки в ГИС ЖКХ;

– сопоставление полученных данных со скоростью обработки платежей в ГИС

ЖКХ (более подробно описано в п. 4.1).

Исходные данные

Было получено разрешение от одного из заказчиков провести данный

эксперимент на промышленных стендах РИАС ЖКХ и ГИС ЖКХ в рамках

нагрузочных

испытаний РИАС ЖКХ.

Также из отчёта о нагрузочном тестировании биллинговой системы

заказчика известно, что данная система способна регистрировать до 120000

фактов оплат в час (2000 в минуту). Из п. 4.1 известно, что ГИС ЖКХ может

обрабатывать до 1466 фактов оплат в минуту.

План эксперимента

а) согласование с заказчиком времени и продолжительности проведения на-

грузочного тестирования;

б) настройка РИАС ЖКХ для проведения нагрузочного тестирования веб-

сервиса платежей;

в) разработка программного обеспечения для проверки времени ответа

вебсервиса платежей РИАС ЖКХ;

г) сбор статистики по времени ответа веб-сервиса платежей РИАС ЖКХ;

д) сбор статистики по времени обработки фактов оплат для их отправки в

ГИС ЖКХ;

е) сопоставление полученных данных с исходными;

ж) составление выводов.

Выполнение эксперимента

Для проведения нагрузочного тестирования на промышленном стенде заказчика в ночное время был развёрнут тестовый экземпляр веб-сервиса по приёму платежей (фактов оплат). Затем на платформе Microsoft Visual Studio был разработан нагрузочный тест, который параллельно в заданное число потоков отправлял запросы на регистрацию платежей. Конфигурация нагрузочного теста представлена в таблице 17.

Таблица 17 – Конфигурация нагрузочного теста для проведения испытаний платёжного шлюза

Параметр	Значение
1	2
ПО веб-сервиса	IIS Express
Количество одновременных клиентов API	10
Метод API	SendOrderInfoAgent (регистрация факта оплаты платёжным агентом)
Время тестирования	5 минут
Конфигурация веб-сервиса	Release
Лист	75

Снимок экрана с загрузкой процессора и оперативной памяти представлен на рисунке 19.

Иные показатели результата нагрузочного теста представлены на рисунке

20. Ожидаемые и полученные результаты

Ожидается, что веб-сервис РИАС ЖКХ будет обрабатывать более 1000 платежей в минуту. Процент ошибок должен быть менее 0,3%. По результатам нагрузочного теста платёжный шлюз РИАС ЖКХ смог обработать за 5 минут 21830 платежей (4366 в минуту), что больше ожидаемого результата в 4 раза. Ошибок было зарегистрировано 0, что меньше ожидаемого результата. Выводы

Платёжный шлюз РИАС ЖКХ, развёрнутый на площадке заказчика, успешно справляется с пиковыми нагрузками и имеет запас по скорости регистрации и обработки фактов оплат и не требует проведения дополнительной оптимизации про-

граммного обеспечения и улучшения аппаратного обеспечения серверов.

4.3 Сравнение скорости передачи данных различными методами обмена

Цели и задачи эксперимента

Целью данного вычислительного эксперимента является составление общего и максимально полного понимания характеристик передачи данных при использовании методов информационного обмена, используемых в РИАС ЖКХ.

Цель достигается при помощи следующих задач:

– выделение основных типовых видов информации, по которым осуществля-

Рисунок 19 – Снимок экрана с загрузкой процессора и оперативной памяти. ЦП – красным, ОЗУ – зелёным

Рисунок 20 – Иные показатели результата нагрузочного теста

ется информационный обмен;

– разработка тестового стенда для проверки различных методов информационного взаимодействия;

– выполнение нагрузочных испытаний и сбор статистических данных для

- последующей их обработки;
- формирование выводов, на каких типичных данных какой метод информационного взаимодействия использовать целесообразнее.

Лист

77

Исходные данные

Существует три типовых вида информации:

- много записей с малым числом данных (например, показания приборов учёта);
- записи с большим числом данных (например, сведения об общедомовом приборе учёта);
- записи с вложенными множественными данными (например, платёжный документ со списком начислений по каждой услуге).

План эксперимента

- а) выбор площадки для проведения эксперимента;
- б) генерация данных;
- в) написание прикладного ПО, позволяющее проводить информационный обмен;
- г) проведение измерений скорости, времени и объёма передаваемой информации;
- д) формирование графиков;
- е) составление выводов.

Выполнение эксперимента

Протокол проведения вычислительного эксперимента:

TestSmall [2017-06-06 20:00]

Записей: 2 174 410.

Значащих полей: 3.

Размер таблицы: 71.38 Мб (индексов 21.38 Мб).

Прямойобмен. Время передачи (секунд): 60, 63, 57, 55, 63 (среднее 59.6).

Использование С# EF6. Время вычитывания из БД (секунд): 55, 43, 47, 42, 47 (среднее 46.8).

Скорость передачи информации: 5.2 Мб/с.

Реестры. Объём выгрузки: 92.60 Мб. Формирование (секунд): 55, 64, 86, 54, 75 (среднее 66.8).

REST. Объём выгрузки: 177.46 Мб. Формирование (секунд): 56, 79, 79, 72, 74 (среднее 72.0).

SOAP. Объём выгрузки: 329.25 Мб. Формирование (секунд): 65, 74, 79, 78, 78 (среднее 75.0).

TestLarge [2017-06-06 21:00]

Записей: 434 890.

Значащих полей: 15.

Размер таблицы: 168.88 Мб (индексов 0.65 Мб).

Прямойобмен. Время передачи (секунд): 98, 73, 58, 70, 55 (среднее 70.8).

Использование С# EF6. Время вычитывания из БД (секунд): 19, 16, 18, 19, 19 (среднее 18.2).

Скорость передачи информации: 5.2 Мб/с.

Реестры. Объём выгрузки: 157.79 Мб. Формирование (секунд): 25, 26, 23, 25, 23 (среднее 24.4).

REST. Объём выгрузки: 221.31 Мб. Формирование (секунд): 29, 26, 30, 29, 26 (среднее 28.0).

SOAP. Объём выгрузки: 290.24 Мб. Формирование (секунд): 29, 28, 28, 27, 29 (среднее 28.2).

TestMany [2017-06-06 23:00]

Записей: 108 724 + 1 087 240 * 2.

Значащих полей: 15.

Размер таблиц: 124.89 Мб (индексов 30.40 Мб).

Прямойобмен. Время передачи (секунд): 35, 35, 49, 40, 47 (среднее 41.2).

Использование С# EF6. Время вычитывания из БД (секунд): 273, 337, 343, 250, 264 (среднее 293.4).

Скорость передачи информации: 5.2 Мб/с.

Реестры. Объём выгрузки: 164.24 Мб. Формирование (секунд): 340, 465, 299, 292, 272 (среднее 333.6).

REST.Объёмвыгрузки:253.97Мб.Формирование(секунд):299,308,285,268, 254 (среднее 282.8).

SOAP. Объём выгрузки: 454.22 Мб. Формирование (секунд): 605, 710, 686, 385, 445 (среднее 566.2).

Ожидаемые и полученные результаты

Ожидалось, что самым быстрым и меньшим по объёму на всех наборах данных будет метод прямого обмена. Самым медленным и большим по объёму на всех наборах данных будет обмен при помощи SOAP. Также ожидалось, что во вложенных данных время и объём выгрузок для REST и SOAP по отношению к прямому обмену будут ниже, чем при других тестовых данных.

График отношения объёма выгрузки к объёму таблиц в базе данных представлен на рисунке 21.

Логарифмический график сравнения времени обмена представлен на рисунке 22.

Выводы

Самым быстрым практически во всех случаях оказался прямой обмен. Это обусловлено закрытым бинарным протоколом передачи данных между СУБД Microsoft SQL Server, а также отсутствием затрат на возможную сериализацию данных. Однако, в случае с большим числом столбцов в таблице (тест TestLarge) время передачи данных по прямому обмену затратилось больше, чем у любого другого метода. Это может быть связано с задержками от передачи метаданных в связи с разными версиями СУБД источника и приёмника данных.

На всех тестовых данных передача при помощи SOAP показала худший результат как по объёму, так и по времени передачи данных. Связано это, прежде всего, со сложностью протокола формирования и передачи информации. Передача информации при помощи архитектурного стиля REST оказалось менее объёмной, чем SOAP, но более объёмной, чем реестры. По времени передачи информации лишь в тесте с большим числом данных в записи REST справился

быстрее реестров.

Также стоит отметить резкое увеличение времени передачи данных при использовании вложенности. Это связано с особенностями используемой технологии доступа к данным: Entity Framework 6. Данная технология генерирует не самые оптимальные запросы при выборке связанных записей из БД.

Рисунок 22 – Логарифмический график сравнения времени обмена

Лист

82

ЗАКЛЮЧЕНИЕ

В рамках данной работы автор формально описал механизм информационного обмена, определил характеристики и ограничения типичных механизмов для сферы жилищно-коммунального хозяйства, доработал региональную информационно-аналитическую систему ЖКХ, провёл ряд вычислительных экспериментов в подтверждение выявленной модели и рекомендаций.

Работа является актуальной, так как в 70 сфере ЖКХ до сих пор 70 не существует

единого формата информации, который необходимо заносить в единое место.

Также определённая предметная область является достаточно широкой, чтобы в ней могли сочетаться информационные системы разного назначения: справочные, расчётные, финансовые, аналитические и прочие.

Работа над изучением информационного обмена в сфере ЖКХ заняла у автора более полутора лет. Более года автор работает над созданием модуля интеграции региональной системы ЖКХ с государственной информационной системой. Автор участвовал в технических сессиях по информационному обмену в сфере ЖКХ, проводимых Министерством Строительства Российской Федерации и ЗАО «ЛАНИТ» (разработчик ГИС ЖКХ).

Работа имеет научные и инженерные перспективы. Дальнейшая формализация принципов информационного обмена может быть полезна для приложения к методам обмена теории управления или анализа данных и систем.

Программные решения (два модуля региональной системы ЖКХ), созданные в рамках написания работы, были зарегистрированы как программы для ЭВМ.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Альбом ТФФ v.11.2.0.8 / ГИС ЖКХ. – [Б. м. : б. и.], 2017. – Режим доступа: <https://dom.gosuslugi.ru/#/regulation> (дата обращения: 06.06.2017)
2. Альбом ТФФ v.11.2.0.8 Приложение 1. Форматы электронных сообщений / ГИС ЖКХ. – [Б. м. : б. и.], 2017. – Режим доступа: <https://dom.gosuslugi.ru/#/regulation> (дата обращения: 06.06.2017)
3. ГОСТ 19.701-90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. – М. : Стандартинформ, 2010.
4. ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам. – М. : Стандартинформ, 1996.
5. ГОСТ Р 34.10-2012. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. – М. : Стандартинформ, 2012. - Режим доступа: <http://www.altell.ru/legislation/standards/gost-34.10-2012.pdf> (дата обращения: 06.06.2017)
6. ГОСТ 34.601-90. Автоматизированные системы. Стадии создания. – М. : Изд-во стандартов, 1997.
7. ГОСТ 7.1-2003. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления. М. : Стандартинформ, 2010.
8. Дутиков, Д. Н. Применение SOAP-сервисов для обеспечения взаимодействия внутри распределённой информационной системы // Вестник Южно-Уральского государственного университета. - Челябинск, 2010. - No22 (198). - С. 9-14
9. Заленский, Д. А. Разработка универсальной модульной системы удалённой обработки геоданных, основанной на технологии SOAP (Simple Object Access Protocol) // Вестник Южно-Уральского государственного университета. - Ханты-Мансийск, 2006. - No4. - С. 31-35
10. Порядок проведения тестирования v.11.1.0.6 / ГИС ЖКХ. – [Б. м. : б. и.], 2017. – Режим доступа: <https://dom.gosuslugi.ru/#/regulation> (дата обращения: 06.06.2017)
11. Плотников А. В. Реализация транзакций в RESTful веб-службах / А. В. Плотников, С. В. Золотарев // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов международной науч.-тех. конференции, Воронеж, 12-15 сент. 2016 г. – Воронеж. : Изд-во ВГУ, 2016. – С. 165–167

12. Приказ Министерства связи и массовых коммуникаций РФ и Министерства

строительства и жилищно-коммунального хозяйства РФ от 29 февраля 2016 г. 2

№ 74/11 4/ 43 пр « 2 Об утверждении состава, сроков и периодичности размещения

информации поставщиками информации в государственной информационной

системе жилищно-коммунального хозяйства» / 5 ГАРАНТ.РУ. – [Б. м. : б. и.],

2016. – Режим доступа: <http://www.garant.ru/products/ipo/prime/doc/71311946/>

(73 дата обращения: 06.06.2017)

13. 70 Регламент взаимодействия внешних систем с ГИС ЖКХ v.11.1.0.7 / ГИС

ЖКХ. – [Б. м. : б. и.], 2017. – Режим доступа:

<https://dom.gosuslugi.ru/#!/regulation> (дата

обращения: 06.06.2017)

14. Родионов, В. В. Дипломное проектирование: учебно-методическое пособие

для студентов специальности 23020165 «Информационные системы и

технологии» / В. В. Родионов. – Ульяновск : УлГТУ, 2008. – 98 с.

15. Справочники ГИС ЖКХ v.11.2.0.8 / ГИС ЖКХ. – [Б. м. : б . и.], 2016. – Режим

доступа: <https://dom.gosuslugi.ru/#!/regulation> (дата обращения: 06.06.2017)

16. Троелсен, Э. Язык про 6 граммирован 8 ия C# 2010 и платформа .NET 4 / Э. Троел-

сен. – 5-е изд. – М. : Вильямс, 2010. – 1392 с.

17. Федеральный закон от 21 июля 2014 г. N 209-ФЗ «О государственной

информационной системе жилищно-коммунального хозяйства» / 17 Российская

Газета. – [Б. м. : б. и.], 2014. – Режим доступа: <https://rg.ru/2014/07/23/gkh->

[dok.html](#) (дата

обращения: 06.06.2017) 15

18. Федеральный закон от 29.12.2004 N 188- 15 ФЗ « 46 Жилищный кодекс 46 Российской

Федерации» [Электронный ресурс] / 86 КонсультантПлюс. – [Б. м. : б. и.], 2004. –

Режим доступа: 48 http://www.consultant.ru/document/cons_doc_LAW_171389/ (дата

обращения: 08.05.2015).

19. 52 Фленов, М. Е. Библия C# / М. Е. Фленов. – 2-е изд. – СПб. : БХВ-Петербург,

2011. – 560 с.

20. Чертовской, В. Д. Базы и банки данных: Учебное пособие / В. Д. Чертовской.

–

СПб. : Изд-во МГУП, 2001. – 220 с.

21. API НА ОСНОВЕ SOAP И REST / Поскребышев Р. С., Тарасов В. Г. //

МОЛОДЫЕ УЧЕНЫЕ - УСКОРЕНИЮ НАУЧНО-ТЕХНИЧЕСКОГО

ПРОГРЕССА В

XXI ВЕКЕ : сборник материалов IV Всероссийской научно-технической

конференции аспирантов, магистрантов и молодых учёных с международным

уча-

стием. - Ижевск. : ИННОВА, 2016. - С. 404-410

22. Availability of Features in Visual Studio Versions // Microsoft Developer Network 15 .

– [Б. м. : б. и.], 2015. – Режим доступа: 15 <https://msdn.microsoft.com/en->

[us/library/ 15 ee519072.aspx](#) (дата обращения: 24.05.2015)

23. RESTful Web APIs / Leonard Richardson, Mike Amundsen, Sam Ruby : O'Reilly

Media, 2013. – 406 с.

24. Rogriguez, A. RESTful Web services: The basics / IBM developerWorks. - [Б. м. :

б. и.], 2015. – Режим доступа:

<https://www.ibm.com/developerworks/library/wsrestful/> (дата обращения:

06.06.2017)

25. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) / W3C

Recommendation. - [Б. м. : б. и.], 2007. – Режим доступа:

<https://www.w3.org/TR/soap12-part1/> (дата обращения: 06.06.2017)

26. Web-протоколы. Теория и практика. / Б. Кришнамурти, Дж. Рэксфорд – М.:

ЗАО

«Издательство БИНОМ», 2002. – 22 592 с.

Лист

Приложение А

(обязательное)

Текст программы

Пример обвязки над клиентом веб-сервиса ГИСЖКХ

```
using System;

using System.Collections.Generic;

using System.Linq; using

System.Text; using

System. 35 Threading.Tasks;

using AIS.HM.Integration.GIS.Core.Wrapper; using

AIS.HM.Integration.GIS.Services.DeviceMetering; using

AIS.HM.Integration.GIS.Core.Utils; namespace

AIS.HM.Integration.GIS.Core.ServiceProviders

{

public sealed class DeviceMeteringServiceProvider : ServiceProvider

{

private DeviceMeteringPortTypesAsyncClient client; public

DeviceMeteringServiceProvider()

{

client = Core.Makers.ClientFactory<DeviceMeteringPortTypesAsyncClient,

DeviceMeteringPortTypesAsync>.Get(this) ;

}

private RequestHeader GenerateRequestHeader(Guid orgPpaGuid, Guid?

messageGuid)

{

return new RequestHeader()

{

MessageGUID = (messageGuid ?? Guid.NewGuid()).ToLowerString(),

Date = DateTime.Now,

IsOperatorSignatureSpecified = true,

IsOperatorSignature = true,

ItemElementName = ItemChoiceType1.orgPPAGUID, Item =

orgPpaGuid.ToLowerString()

};

}

#region api methods

public AckRequest importMeteringDeviceValues(

importMeteringDeviceValuesRequest request, Guid orgPpaGuid, Guid? messageGuid = null)

{

OnApiCalling("importMeteringDeviceValues"); try

{

if ( ConfigurationBase.Current.IsProduction )

{

var xmlRequest = Core.Utils.XmlSigner.Form(request); xmlRequest =

Core.Utils.XmlSigner.Sign(xmlRequest);

request = XmlHelper.Deserialize<importMeteringDeviceValuesRequest

>( xmlRequest);

}

AckRequest result = null;

client.importMeteringDeviceValues(GenerateRequestHeader(orgPpaGuid,

messageGuid), request, out result);

return result;

}

catch (Exception ex)

{

OnApiException(ex);
```

```

throw new GisServiceException(ex);
}
}

public AckRequest exportMeteringDeviceHistory(
    exportMeteringDeviceHistoryRequest request, Guid orgPpaGuid, Guid? messageGuid = null)
{
    OnApiCalling("exportMeteringDeviceHistory"); try
    {
        if ( ConfigurationBase.Current.IsProduction )
        {
            var xmlRequest = Core.Utils.XmlSigner.Form(request);
            xmlRequest = Core.Utils.XmlSigner.Sign(xmlRequest); request =
                XmlHelper.Deserialize<
                    exportMeteringDeviceHistoryRequest>( xmlRequest);
        }

        AckRequest result = null;

        client.exportMeteringDeviceHistory(GenerateRequestHeader(orgPpaGuid,
            messageGuid), request, out result); return
            result;
        }

        catch (Exception ex)
        {
            OnApiException(ex);
            throw new GisServiceException(ex);
        }
    }

    public getStateResult getState(getStateRequest request, Guid orgPpaGuid)
    {
        OnApiCalling("getState"); try
        {
            getStateResult result = null;

            client.getSta te(GenerateRequestHeade 24 r(orgPpaGuid, null), request, out
                result); result.Signature = null;

            return result;
        }

        catch (Excepti 31 on ex)
        {
            OnAp 20 iException(ex);
            throw new GisServiceException(ex);
        }
    }

    #endregion
}

}

Обмен с файлами в ГИСЖКХ

using Newtonsoft.Json; using System;

using S 25 ystem.Collections.Generic; using
System. 35 Collections. 35 Specialized; using
System.Diagnostics; 34 using System.IO; 35
using System.Linq; using System. 35 l Net;

using System.Security.Cryptography; using System. 35 Text; using System. 35 Threading.Tasks; using
System.Web; namespace AIS.HM.Integration.GIS.Core.Utils
{
    /// <summary>

    /// Абстрактный класс, реализующий обмен файлами между ГИС ЖКХ и файловой системой.

    /// </summary>

    public abstract class FileServiceBase

```

```

{
private const int MAX SIZE FILE PART = 5242880; //максимальный размер файла,

когда его можно загружать и выгружать на ГИС ЖКХ одной частью

#region Upload

public FileDescriptorGis UploadToGis(string fileName, FileStorageType
storageType, string orgPpaGuid, out string hash, string fileDisplayName = null)

{
try
{
var fileInfo = new FileInfo(fileName); if (fileInfo.Length > MAX SIZE FILE
PART) return UploadToGisBigFile(fileName, fileDisplayName, storageType,
orgPpaGuid, fileInfo, out hash); return UploadToGisSmallFile(fileName,
fileDisplayName, storageType,
orgPpaGuid, fileInfo, out hash);
}

catch (WebException ex)
{
if (ex.Status == WebExceptionStatus.ProtocolError)
{
37
var response = ( HttpWebResponse)ex.Response;

if (response == null) throw new Exception("Ошибкаполучения файла от
ГИС ЖКХ.", ex);

var headerText = response.Headers["X-Upload-Error"]; throw new
Exception(headerText ?? "Ошибка получения файла от ГИС
ЖКХ.", ex);
}

throw new Exception("Ошибкаполучения файла от ГИС ЖКХ.", ex);
}

catch (Exception ex)
{
throw new Exception("Ошибкаполучения файла от ГИС ЖКХ.", ex);
}
}

private FileDescriptorGis UploadToGisSmallFile(string fileName, string
fileDisplayName, FileStorageType storageType, string orgPpaGuid, FileInfo fileInfo, out
string hash)
{
var wp = $"{ConfigurationBase.Current.GisFileEndpoint}{storageType.
ToString()}/"; var wc = new SecureWebClient(); hash = getGostHash(fileName);
wc.Headers.Add("Content-MD5", Md5toBase64( getMd5(fileName)));
wc.Headers.Add("X-Upload-OrgPPAGUID", orgPpaGuid);
wc.Headers.Add("X-Upload-Filename", HttpUtility.UrlEncode(fileDisplayName
?? fileInfo.Name));

wc.UploadFile(wp, "PUT", fileName);

return new FileDescriptorGis { Guid = wc.ResponseHeaders["X-Upload- UploadID"],
Storage = storageType };
}

private FileDescriptorGis UploadToGisBigFile(string fileName, string
fileDisplayName, FileStorageType storageType, string orgPpaGuid, FileInfo fileInfo, out
string hash)
{
var wp = $"{ConfigurationBase.Current.GisFileEndpoint}{storageType.
ToString()}/"; var wc = new SecureWebClient(); hash = getGostHash(fileName); int
fileOffset = 0, fileParts = (fileInfo.Length / MAX SIZE FILE PART) % 10 == 0
? (int)(
fileInfo.Length / MAX SIZE FILE PART) : (int)(fileInfo.Length /

```

MAX SIZE FILE PART) + 1 ;

#region Инициализация сессии

/*

Запрос на инициализацию сессии

POST /ext-bus-file-store-service/rest/<upload-context>/?upload HTTP

/1.1

Host: host

Date: date

X-Upload-Datapvider: GUID X-Upload-Filename:

filename X-Upload-Length: length

X-Upload-Part-Count: number

*/ wc.Headers.Add("X-Upload-OrgPPAGUID", orgPpaGuid);

wc.Headers.Add("X-Upload-Filename", HttpUtility.UrlEncode(fileDisplayName

?? fileInfo.Name)); wc.Headers.Add("X-Upload-Length",

fileInfo.Length.ToString()); wc.Headers.Add("X-Upload-Part-Count",

fileParts.ToString());

wc.UploadString(wp + "?upload", string.Empty); var uploadID

= wc.ResponseHeaders["X-Upload-UploadID"]; #endregion

#region Загрузка части файла

/*

PUT /ext-bus-file-store-service/rest/<upload-context>/<uploadID> HTTP

/1.1

Host: host

Date: date

Content-Length: size Content-MD5:

hash

X-Upload-Partnumber: partnumber

*/

FileStream fs = new FileStream(fileName, FileMode.Open);

byte[] data = new byte[fileInfo.Length]; fs.Read(data, 0,

data.Length); for (int i = 0; i < fileParts; i++)

{

int dataItemLength = data.Length - fileOffset >= MAX SIZE FILE PART ?

MAX SIZE FILE PART : data.Length - fileOffset;

byte[] dataItem = new byte[dataItemLength];

Array.Copy(data, fileOffset, dataItem, 0, dataItemLength); wc =

new SecureWebClient();

wc.Headers.Add("X-Upload-OrgPPAGUID", orgPpaGuid);

wc.Headers.Add("X-Upload-Partnumber", (i + 1).ToString());

wc.Headers.Add("Content-MD5", Md5toBase64(getMd5byByte(dataItem)));

wc.Headers.Add("Content-Type", "application/x-www-form-urlencoded");

wc.UploadData(string.Format("{0}{1}", wp, uploadID), "PUT", dataItem)

;

fileOffset += dataItem.Length;

}

fs.Close(); #endregion

#region Завершение сессии загрузки

/*

POST /ext-bus-file-store-service/rest/<upload-context>/<uploadID>?

completed HTTP/1.1

Host: host

Date: date

*/ wc = new SecureWebClient(); wc.Headers.Add("X-Upload-OrgPPAGUID",

orgPpaGuid); wc.UploadString(string.Format("{0}{1}?completed", wp, uploadID),

string.

Empty);

```

#endregion return new FileDescriptorGis { Guid = uploadID, Storage =
storageType };

}

protected string downloadFromRias(string url)

{

// костыль для localhost if ( url.StartsWith("/") ) url =
ConfigurationBase.Current.HmWebPath + url.Substring(1);

// создание временного файла var
tempFile = Path.GetTempFileName();

// загрузка во временный файл

using (SecureWebClient client = new SecureWebClient())

{

client.DownloadFile(url, tempFile);

}

return tempFile;

}

#endregion #region

Download

public string DownloadFromGis(FileDescriptorGis gisFile, string orgPpaGuid,
out string gisFileName)

{

try

{

// variables initialize var wp = string.Format("{0}/{2}?getfile",
ConfigurationBase.
Current.GisFileEndpoint, gisFile.Storage.ToString(), gisFile.Guid); var wc
= new ExWebClient();

// get info of downloaded file

wc.Headers.Add("X-Upload-OrgPPAGUID", orgPpaGuid);

wc.Method = "HEAD"; wc.DownloadString(wp);

// handle info

var fileLength = Convert.ToInt32(wc.ResponseHeaders["X-Upload-Length"
]); gisFileName = MimeKit.Utlis.Rfc2047.DecodeText(Encoding.UTF8.GetBytes
(wc.ResponseHeaders["X-Upload-Filename"]));

string extension = Path.GetExtension(gisFileName); if
(fileLength > MAX SIZE FILE PART)

{

var completedPartsCount = wc.ResponseHeaders["X-Upload-Completed-
Parts"].Split(',').Count(); return DownloadBigFileFromGis(orgPpaGuid, wp,
fileLength,
completedPartsCount, extension);

}

return DownloadSmallFileFromGis(orgPpaGuid, wp, extension);

}

catch (WebException ex) { 37

{

if (ex.Status == WebExceptionStatus.ProtocolError)

{ 37

var response = ( HttpWebResponse)ex.Response;

if (response == null) throw new Exception("Ошибказагрузки файла
в РИАС ЖКХ.");

var headerText = response.Headers["X-Upload-Error"]; throw new
Exception(headerText ?? "Ошибка загрузки файла в РИАС
ЖКХ.");

}

throw new Exception("Ошибказагрузки файла в РИАС ЖКХ.", ex);

}

}

```

```

catch (Exception ex)
{

throw new Exception("Ошибка загрузки файла в РИАС ЖКХ.", ex);

}

}

private string DownloadSmallFileFromGis(string orgPpaGuid, string
serviceAddress, string extension)

{

string tempFileName = GenerateTempFile(extension);

// download file into temporary firectory var wc =
new ExWebClient(); wc.Method = "GET";

wc.Headers.Add("X-Upload-OrgPPAGUID", orgPpaGuid);

wc.DownloadFile(serviceAddress, tempFileName); return
tempFileName;

}

/// <summary>
/// Генерирует уникальное имя файла
/// </summary>
/// <param name="extensionТребуемое"> расширение</param>
/// <returns></returns>

private static string GenerateTempFile(string extension)

{

var path = Path.GetTempPath(); var tempFileName =
Guid.NewGuid().ToString() + extension; tempFileName =
Path.Combine(path, tempFileName); return tempFileName;

}

private string DownloadBigFileFromGis(string orgPpaGuid, string
serviceAddress, int fileLength, int completedPartsCount, string extension)

{

string tempFileName = GenerateTempFile(extension);

byte[] data = new byte[ fileLength]; int offset = 0 ;

for (int i = 0; i < completedPartsCount; i++)

{

int currentPartLength = fileLength - offset > MAX SIZE FILE PART ?
MAX SIZE FILE PART : fileLength - offset;

var wc = new ExWebClient(); wc.Method
= "GET";

wc.Headers.Add("X-Upload-OrgPPAGUID", orgPpaGuid);

wc.RangeFrom = offset;

wc.RangeTo = offset + currentPartLength - 1 ; var
dataltem = wc.DownloadData(serviceAddress);

Array.Copy(dataltem, 0, data, offset, currentPartLength); offset +=
currentPartLength;

}

File.WriteAllBytes(tempFileName, data); return
tempFileName;

}

#endregion

#region Utils

/// <summary>
/// Получает MD5сумму- файла.
/// </summary>
/// <param name="fileNameПолное"> и абсолютное имя файла.</param>
/// <returns></returns>

private static string getMd5(string fileName)

{


```



```

using (var md5 = MD5.Create())

{

using (var stream = File.OpenRead(fileName))

{

return BitConverter.ToString(md5.ComputeHash(stream)).Replace("-",
, "").ToLower();

}

}

}

}

/// <summary>

/// Получает MD5сумму- набора byte данных

/// </summary>

/// <param name="dataArrayНабор"> данных</param>

/// <returns></returns>

private static string getMd5byByte(byte[] dataArray)

{

using (var md5 = MD5.Create())

{

using (var stream = new MemoryStream(dataArray))

{

return BitConverter.ToString(md5.ComputeHash(stream)).Replace("-",
, "").ToLower();

}

}

}

}

/// <summary>

/// Преобразует MD5сумму- к формату Base64.

/// </summary>

/// <param name="hexStringИсходная"> MD5строка-</param>

/// <returns></returns>

private static string Md5toBase64(string hexString)

{

byte[] buffer = new byte[hexString.Length / 2] ;

for (int i = 0; i < hexString.Length; i++)

{

buffer[i / 2] = Convert.ToByte(Convert.ToInt32(hexString.Substring(i,
2), 16)); i += 1 ;

}

string res = Convert.ToBase64String(buffer);

/// Получение хешсуммы- файла согласно ГОСТ 34.11-94.

/// <param name="fileNameПолное"> и абсолютное имя файла.</param>

/// <returns></returns> private static string getGostHash(string fileName)

var asseblyFile = System.Reflection.Assembly.GetExecutingAssembly().

var cpverifyPath = Path.GetDirectoryName(asseblyFile) + @"\Libraries\"; var

cpverifyFile = cpverifyPath + @"cpverify.exe"; var arguments =

string.Format("-mk \"{0}\"", fileName);

Process p = new Process();

p.StartInfo.UseShellExecute = false;

p.StartInfo.RedirectStandardOutput = true;

p.StartInfo.FileName = cpverifyFile;

p.StartInfo.Arguments = arguments;

string output = p.StandardOutput.ReadLine(); p.WaitForExit();

if (output.Length != 64) throw new Exception(string.Format("Wrong hash

from cpverify.exe\

nFileName: {0}\nArgs:{1}\nOutput:{2}", cpverifyFile, arguments, output));

string result = string.Empty;

for (int i = 1; i < 64; i += 2) // 1, 3, 5, ..., 63

```

```
result += output[i]; result
+= output[i - 1]; return

result.ToLower();

protected class JsonResponse

public dynamic data { get; set; } public

string message { get; set; } public bool

success { get; set; }

Структура, описывающая файл в ГИС ЖКХ

public struct FileDescriptorGis

public string Guid { set; get; } public

FileStorageType Storage { set; get; }

/// Тип хранилища ГИС ЖКХ.

public enum FileStorageType { homemanagement, rki, voting, inspection, informing, bills,
licenses, agreements, nsi }

/// Структура, описывающая файл в РИАС ЖКХ.

public struct FileDescriptorRias

{ publicpublicpublic intstringstring?

OrganizationIdFileNameFileGroup{ {set; set;{

set;set;set;set;}} } }

/// <summary>

/// Хитрый класс для описания файлов, пришедших из vw GISFileCompare.

/// </summary>

public class FileEntityDescription

{ publicpublic GuidFileStorageType Guid { get;Storage;set; } { get; set; }

publicpublicpublic intstringstring? OrganizationId FileNameFileGroup{ {set;

set;set;set;set;}} }

{ get; }

publicpublicpublic intstringstringLocalId TableNameSchemaName{ get;{

{set;set;set;stringset;set;fileGroup,} } int? organizationId, string public

FileEntityDescription(

fileName, string tableName, string schemaName, int localId)

{ FileGroup = fileGroup;

OrganizationId = organizationId;

FileName = fileName; TableName

= tableName;

SchemaName = schemaName; LocalId

= localId; publi c}

FileEntityDescription() { }

}

}

Обработка новых платежей от ПШРИАСЖКХ

using AIS.HM.Integration.GIS.Core.Utills; using

AIS.HM 31 .Integration.GIS.Production.Classes; using

AIS.HM.Integration.GIS.Services.Payment; using

AIS.HM.Model; using System;

using System.Collections.Generic; using System.Linq; using

System.Text; using System. 35 Threading.Tasks; namespace

AIS.HM.Integration.GIS.Production.Activities.Impl

{ /// <summary>

/// Обрабатывает запросы от платёжного сервиса.

/// </summary> public class PaymentProcessActivity

: IActivity

{ public void Run()

{ ProcessOvertimed();

ProcessNew();

}
```

```
public void Run(object obj)
{
    throw new NotImplementedException();
}

private void ProcessOvertime()
{
    try
    {
        using (var db = new DB())
        {
            db.Database.CommandTimeout = 3600 ; db.GISPaymentServiceLog
            processOvertime();
        }
    }
    catch (Exception ex)
    {
        Log.Exception(ex, "Ошибкаобработкипросроченныхплатежей");
    }
}

private void ProcessNew()
{
    try
    {
        using (var db = new DB())
        {
            db.Database.CommandTimeout = 3600 ;
            db.GISPaymentServiceLogprocessInfo();
            db.GISPaymentServiceLogprocessInfoAgent();
            db.GISPaymentServiceLogprocessCancellation();
        }
    }
    catch (Exception ex)
    {
        Log.Exception(ex, "Ошибкаобработки новых платежей");
    }
}
}
```

Приложение Б

(обязательное)

SQL-команды генерации данных для вычислительных экспериментов

В текущем приложении приводится код на языке T-SQL, при помощи которого были сгенерированы тестовые наборы данных для проведения вычислительного эксперимента из п. 4.3.

Набор данных D1

--- Создание таблицы

```
IF 1=1 BEGIN

CREATE TABLE sandbox.MoiseevTestSml (

[TestSmlId] INT PRIMARY KEY IDENTITY(1,1),

[ItemId] INT NOT NULL INDEX IX TestSml ItemId(TestSmlId),

[Date] DATETIME2 NOT NULL,

[Value] DECIMAL(18,4) NOT NULL )

END
```

--- Заполнение данными

```
IF 1=1 BEGIN

CREATE TABLE #dates (d DATETIME2) CREATE

TABLE #ids (i INT)

;WITH cte AS (

SELECT CAST('2017-01-01 00:00' AS DATETIME2) AS dt

UNION ALL

SELECT DATEADD(MINUTE, 1, dt)

FROM cte s

WHERE DATEADD(MINUTE, 1, dt) <= '2017-06-01 00:00')

INSERT INTO #dates ( d )

SELECT s.dt

FROM cte s
```

```
OPTION (MAXRECURSION 0)

;WITH cte AS (

SELECT 1 AS dt

UNION ALL

SELECT 1+ dt

FROM cte s WHERE

1+dt <= 10)

INSERT INTO #ids ( i )

SELECT s.dt

FROM cte s


OPTION (MAXRECURSION 0)

INSERT INTO sandbox.MoiseevTestSml (ItemId, Date, Value)

SELECT i.i, d.d, CAST(CAST(NEWID() AS VARBINARY) AS INT)

FROM #dates d

CROSS JOIN #ids i

DROP TABLE #dates

DROP TABLE #ids

END

-- Проверка числа записей

SELECT COUNT(*) FROM sandbox.MoiseevTestSml WITH(NOLOCK) SELECT

TOP 100 * FROM sandbox.MoiseevTestSml

-- Удаление таблицы

IF 1=0 BEGIN

TRUNCATE TABLE sandbox.MoiseevTestSml

DROP TABLE sandbox.MoiseevTestSml END

Набор данных D2

-- Создание таблицы

IF 1=1 BEGIN

CREATE TABLE sandbox.MoiseevTestLarge (

[TestLargeId] INT PRIMARY KEY IDENTITY(1,1),

[Date] DATETIME2 NOT NULL,

[Value01] DECIMAL(18,4) NOT NULL,

[Value02] DECIMAL(18,4) NOT NULL,

[Value03] DECIMAL(18,4) NOT NULL,

[Value04] DECIMAL(18,4) NOT NULL,

[Value05] NVARCHAR(MAX) NOT NULL,

[Value06] NVARCHAR(MAX) NOT NULL,

[Value07] NVARCHAR(MAX) NOT NULL,

[Value08] NVARCHAR(MAX) NOT NULL,

[Value09] UNIQUEIDENTIFIER NOT NULL,

[Value10] UNIQUEIDENTIFIER NOT NULL,

[Value11] INT NOT NULL,

[Value12] INT NOT NULL,

[Value13] INT NOT NULL,

[Value14] INT NOT NULL

)

END

-- Заполнение данными

IF 1=1 BEGIN

CREATE TABLE #dates (d DATETIME2) CREATE

TABLE #ids (i INT)

;WITH cte AS (

SELECT CAST('2017-01-01 00:00' AS DATETIME2) AS dt

UNION ALL

SELECT DATEADD(MINUTE, 5, dt)

FROM cte s

WHERE DATEADD(MINUTE, 5, dt) <= '2017-06-01 00:00')
```

```
INSERT INTO #dates (d)

SELECT s.dt

FROM cte s

OPTION (MAXRECURSION 0)

;WITH cte AS (

SELECT 1 AS dt

UNION ALL

SELECT 1+ dt

FROM cte s WHERE

1+dt <= 10)

INSERT INTO #ids ( i )

SELECT s.dt

FROM cte s

OPTION (MAXRECURSION 0)

INSERT INTO sandbox.MoiseevTestLarge

( Date,

Value01,

Value02,

Value03,

Value04,

Value05,

Value06,

Value07,

Value08,

Value09,

Value10,

Value11,

Value12,

Value13,

Value14

)

SELECT d.d

, CAST(CAST(NEWID() AS VARBINARY) AS INT), CAST(CAST(NEWID() AS VARBINARY) AS INT

), CAST(CAST(NEWID() AS VARBINARY) AS INT), CAST(CAST(NEWID() AS VARBINARY) AS

INT)

, CAST(NEWID() AS NVARCHAR(MAX)), CAST(NEWID() AS NVARCHAR(MAX)), CAST(NEWID() AS

NVARCHAR(MAX)), CAST(NEWID() AS NVARCHAR(MAX))

, NEWID(), NEWID()

, CAST(CAST(NEWID() AS VARBINARY) AS INT), CAST(CAST(NEWID() AS VARBINARY) AS INT

), CAST(CAST(NEWID() AS VARBINARY) AS INT), CAST(CAST(NEWID() AS VARBINARY) AS INT)

FROM #dates d

CROSS JOIN #ids i

DROP TABLE #dates

DROP TABLE #ids

END

-- Проверка числа записей

SELECT COUNT(*) FROM sandbox.MoiseevTestLarge WITH(NOLOCK) SELECT

TOP 100 * FROM sandbox.MoiseevTestLarge

-- Удаление таблицы

IF 1=0 BEGIN

TRUNCATE TABLE sandbox.MoiseevTestLarge

DROP TABLE sandbox.MoiseevTestLarge END

Набор данных D3

-- Создание таблиц

IF 1=1 BEGIN

CREATE TABLE sandbox.MoiseevTestMany1 (
```

```
[Id] INT PRIMARY KEY IDENTITY(1,1),
[Date] DATETIME2 NOT NULL,
[Value01] DECIMAL(18,4) NOT NULL,
[Value02] DECIMAL(18,4) NOT NULL,
[Value03] DECIMAL(18,4) NOT NULL,
[Value04] DECIMAL(18,4) NOT NULL,
[Value05] NVARCHAR(MAX) NOT NULL,
[Value06] NVARCHAR(MAX) NOT NULL,
[Value07] NVARCHAR(MAX) NOT NULL,
[Value08] NVARCHAR(MAX) NOT NULL
```

```
)

CREATE TABLE sandbox.MoiseevTestMany2 (
[Id] INT PRIMARY KEY IDENTITY(1,1),
[Many1Id] INT NOT NULL,
[Value01] UNIQUEIDENTIFIER NOT NULL,
[Value02] UNIQUEIDENTIFIER NOT NULL
```

```
)

CREATE TABLE sandbox.MoiseevTestMany3 (
[Id] INT PRIMARY KEY IDENTITY(1,1),
[Many1Id] INT NOT NULL,
[Value01] INT NOT NULL,
[Value02] INT NOT NULL,
[Value03] INT NOT NULL,
[Value04] INT NOT NULL
```

```
)

ALTER TABLE sandbox.MoiseevTestMany2 ADD CONSTRAINT MoiseevMany2FK FOREIGN KEY ( Many1Id)
REFERENCES sandbox.MoiseevTestMany1(Id)

ALTER TABLE sandbox.MoiseevTestMany3 ADD CONSTRAINT MoiseevMany3FK FOREIGN KEY ( Many1Id)
REFERENCES sandbox.MoiseevTestMany1(Id)
```

```
CREATE NONCLUSTERED INDEX IXMoiseevMany2 ON sandbox.MoiseevTestMany2(Many1Id)

CREATE NONCLUSTERED INDEX IXMoiseevMany3 ON sandbox.MoiseevTestMany3(Many1Id) END
```

```
-- Заполнение данными

IF 1=1 BEGIN

CREATE TABLE #dates (d DATETIME2) CREATE

TABLE #ids (i INT)

;WITH cte AS (

SELECT CAST('2017-01-01 00:00' AS DATETIME2) AS dt

UNION ALL

SELECT DATEADD(MINUTE, 5, dt)

FROM cte s

WHERE DATEADD(MINUTE, 5, dt) <= '2017-06-01 00:00')

INSERT INTO #dates (d)

SELECT s.dt

FROM cte s

OPTION (MAXRECURSION 0)

;WITH cte AS (

SELECT 1 AS dt

UNION ALL

SELECT 1+ dt

FROM cte s WHERE

1+dt <= 10)

INSERT INTO #ids ( i )

SELECT s.dt

FROM cte s

OPTION (MAXRECURSION 0)

INSERT INTO sandbox.MoiseevTestMany1

( Date,
```

```
Value01,
Value02,
Value03,
Value04,
Value05,
Value06,
Value07,
Value08
)

SELECT d.d
, CAST(CAST(NEWID() AS VARBINARY) AS INT), CAST(CAST(NEWID() AS VARBINARY) AS INT
), CAST(CAST(NEWID() AS VARBINARY) AS INT), CAST(CAST(NEWID() AS VARBINARY) AS
INT)
, CAST(NEWID() AS NVARCHAR(MAX)), CAST(NEWID() AS NVARCHAR(MAX)), CAST(NEWID() AS
NVARCHAR(MAX)), CAST(NEWID() AS NVARCHAR(MAX))
FROM #dates d
CROSS JOIN #ids i

INSERT INTO sandbox.MoiseevTestMany2
( Many1Id, Value01, Value02 )
SELECT m1.Id, NEWID(), NEWID()
FROM sandbox.MoiseevTestMany1 m1
CROSS JOIN #ids i

INSERT INTO sandbox.MoiseevTestMany3
( Many1Id, Value01
,
Value02,
Value03,
Value04
)
SELECT m1.Id, i.i+1, i.i+2, i.i+3, i.i+4
FROM sandbox.MoiseevTestMany1 m1 CROSS
JOIN #ids i

DROP TABLE #dates
DROP TABLE #ids

END
```

```
-- Проверка числа записей

SELECT COUNT(*) FROM sandbox.MoiseevTestMany1 WITH(NOLOCK)

SELECT COUNT(*) FROM sandbox.MoiseevTestMany2 WITH(NOLOCK)

SELECT COUNT(*) FROM sandbox.MoiseevTestMany3 WITH(NOLOCK)
```

```
-- Удаление таблиц

IF 1=0 BEGIN

TRUNCATE TABLE sandbox.MoiseevTestMany1

TRUNCATE TABLE sandbox.MoiseevTestMany2

TRUNCATE TABLE sandbox.MoiseevTestMany3

DROP TABLE sandbox.MoiseevTestMany1

DROP TABLE sandbox.MoiseevTestMany2

DROP TABLE sandbox.MoiseevTestMany3 END
```

Приложение В

(обязательное)

Апробация работы

Свидетельство о регистрации программы для ЭВМ№01:

Свидетельство о регистрации программы для ЭВМ№01:

Письмо от автора программных модулей о том, что автор магистерской

дис-

сертации принимал участие при их разработке:

Список публикаций по теме диссертации

а) Тронин В.Г., Моисеев В.В., Храмов Е.М., ИСПОЛЬЗОВАНИЕ
ПРИНЦИПОВ ГИБКИХ МЕТОДОЛОГИЙ УПРАВЛЕНИЯ ПРОЕКТАМИ П
РИ РАЗРАБОТКЕ РЕГИОНАЛЬНОЙ СИСТЕМЫ ЖКХ УЛЬЯНОВСКОЙ
ОБЛА-
СТИ // Вестник УлГТУ. — No2 (74). — Ульяновск, 2016. — С. 38-42

б) Куркина С.В., Моисеев В.В., Прохоров Е.Э и др., АНАЛИЗ ПРИМЕНИ-
МОСТИ SAAS-МОДЕЛИ РАСПРОСТРАНЕНИЯ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ ПРОЦЕССОВ ЖКХ / 15 / При-
кладные информационные системы. 15 — Ульяновск, 2016. — С. 240-245

в) Куркина С.В., Моисеев В.В., Прохоров Е.Э., ИЗУЧЕНИЕ
ВОЗМОЖНОСТЕЙ ПЛАТФОРМЫ TELERIK KENDO ДЛЯ ОТОБРАЖЕНИ
Я ДАННЫХ
В ВЕБ-ОРИЕНТИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ
// Прикладные информационные системы. — Ульяновск, 2016. — С. 232-
239

г) Воронина В.В., Моисеев В.В., ОРГАНИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ
МОДУЛЕЙ ВЕБ-ОРИЕНТИРОВАННОГО ПРОГРАММНОГО КОМПЛЕК-
СА // Вузовская наука в современных условиях : сборник материалов 50й
научно-технической конференции. — Ульяновск: УлГТУ, 2016. — 271 с.
— С. 244-247

д) Моисеев В.В., МЕТОДЫ ИНФОРМАЦИОННОГО ВЗАИМОДЕЙСТВИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ ЖИЛИЩНО-КОММУНАЛЬНОГО
ХОЗЯЙСТВА // Прикладные информационные системы. — Ульяновск,
2017
Лист
106