

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

---

Институт Кибернетики \_\_\_\_\_  
Специальность 09.03.03. Прикладная информатика \_\_\_\_\_  
Кафедра Программной инженерии \_\_\_\_\_

**БАКАЛАВРСКАЯ РАБОТА**

| Тема работы   |
|---|
| «Проектирование блока конвертеров данных из стандартов LAS и WITS в стандарты WITSMML 1.3.1 и 1.4.1, в режиме реального времени.» |

УДК \_\_\_ 004.31.001.63:004.62.031.43 \_\_\_\_\_

**Студент**

| Группа | ФИО  | Подпись | Дата |
|--------|--|---------|------|
| 8К31   | Власкин Глеб Егорович<br>Прохоренко Ангелина Сергеевна |         |      |

**Руководитель**

| Должность                          | ФИО           | Ученая степень,<br>звание | Подпись | Дата |
|------------------------------------|---------------|---------------------------|---------|------|
| Заведующий<br>лабораторией каф. ПИ | Марчуков А.В. |                           |         |      |

**КОНСУЛЬТАНТЫ:**

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

| Должность | ФИО              | Ученая степень,<br>звание | Подпись | Дата |
|-----------|------------------|---------------------------|---------|------|
| Доцент    | Тухватулина Л.Р. | к.ф.н.                    |         |      |

По разделу «Социальная ответственность»

| Должность | ФИО                             | Ученая степень,<br>звание | Подпись | Дата |
|-----------|---------------------------------|---------------------------|---------|------|
| Доцент    | Пустовойтова<br>Марина Игоревна | к.х.н.                    |         |      |

**ДОПУСТИТЬ К ЗАЩИТЕ:**

| Зав. кафедрой | ФИО         | Ученая степень,<br>звание | Подпись | Дата |
|---------------|-------------|---------------------------|---------|------|
| ПИ            | Иванов М.А. | к.т.н.                    |         |      |

**Запланированные результаты изучения по основной образовательной программе подготовки бакалавров 09.03.03. «Прикладная информатика»**

| Код<br>результатов                  | Результат обучения<br>(выпускник должен быть готов)   |
|-------------------------------------|---|
| <i>Профессиональные компетенции</i> |   |
| Р1                                  | Применять глубокие естественнонаучные, математические и инженерные знания для создания и обработки новых материалов   |
| Р2                                  | Применять глубокие знания в области современных технологий машиностроительного производства для решения междисциплинарных инженерных задач  |
| Р3                                  | Ставить и решать инновационные задачи инженерного анализа, связанные с созданием и обработкой материалов и изделий, с использованием системного анализа и моделирования объектов и процессов машиностроения   |
| Р4                                  | Разрабатывать технологические процессы, проектировать и использовать новое оборудование и инструменты для обработки материалов и изделий, конкурентоспособных на мировом рынке машиностроительного производства   |
| Р5                                  | Проводить теоретические и экспериментальные исследования в области современных технологий обработки материалов, нанотехнологий, создания новых материалов в сложных и неопределенных условиях   |
| Р6                                  | Внедрять, эксплуатировать и обслуживать современные высокотехнологичные линии автоматизированного производства, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья и безопасности труда на машиностроительном производстве, выполнять требования по защите окружающей среды |

| <i>Универсальные компетенции</i> |   |
|----------------------------------|---|
| P7                               | Использовать глубокие знания по проектному менеджменту для ведения инновационной инженерной деятельности с учетом юридических аспектов защиты интеллектуальной собственности  |
| P8                               | Активно владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты инновационной инженерной деятельности   |
| P9                               | Эффективно работать индивидуально, в качестве члена и руководителя группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации |
| P10                              | Демонстрировать глубокие знания социальных, этических и культурных аспектов инновационной инженерной деятельности, компетентность в вопросах устойчивого развития   |
| P11                              | Самостоятельно учиться и непрерывно повышать квалификацию в течение всего периода профессиональной деятельности   |

## Форма задания на выполнение выпускной квалификационной работы

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

Институт кибернетики \_\_\_\_\_  
Направление подготовки (специальность) прикладная информатика \_\_\_\_\_  
Кафедра ПИ \_\_\_\_\_

УТВЕРЖДАЮ:  
Зав. кафедрой

\_\_\_\_\_  
(Подпись) (Дата) (Ф.И.О.)

### ЗАДАНИЕ на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы (в составе команды)

(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

| Группа | ФИО                                   |
|--------|---------------------------------------|
| 8К31   | 1. Власкин Г.Е.<br>2. Прохоренко А.С. |

Тема работы:

«Проектирование блока конвертеров данных из стандартов LAS и WITS в стандарты WITSML 1.3.1 и 1.4.1, в режиме реального времени.»

Утверждена приказом директора (дата, номер)

Срок сдачи студентами выполненной работы:

## ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

|  |  |
|--|--|
| <p><b>Исходные данные к работе</b></p> <p><i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономического анализ и т. д.).</i></p>       | <p>Объект исследования -«Агрегатор» -аппаратно-программный комплекс преобразования информации из стандартов LAS и WITS в стандарты WITSML 1.3.1 и WITSML 1.4.1</p> <p>Программные компоненты комплекса должны:</p> <ol style="list-style-type: none"><li>1. Осуществлять приём данных в стандартах LAS и WITS в режимах реального времени по запросу станций ГТИ;</li><li>2. Хранение принятых данных, в файловой базе данных, с привязкой по скважине и времени сеанса;</li><li>3. Передавать принятые файлы данных блоку конвертирования, сохраняя порядок приёма данных;</li><li>4. Осуществлять поиск файла стандарта, по запросу с помощью веб интерфейса;</li></ol>  |
| <p><b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b></p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p> | <ol style="list-style-type: none"><li>1. Назначение. Роль стандартов в разработке месторождений. Обзор станций управления бурением - описание. Сбор данных с буровой. Структура и организация стандартов LAS и WITS. Отчёты с буровых – периодичность, формы. Что такое технология XML- выгоды применения.</li><li>2. Режимы реального времени при построении геологической модели месторождения, выгода. Программные средства построения модели (Petrel, Геонафт). Современный стандарт передачи данных WITSML - краткий обзор. Сервера WITSML -функции. Структура сервера WITSML.</li><li>3. Преобразование из стандартов LAS и WITS в стандарт WITSML - назначение.</li><li>4. Зачем хранить данные в стандартах LAS и WITS в «агрегаторе» Обоснование выбора СУБД.</li></ol> |
| <p><b>Перечень графического материала</b></p> <p><i>(с точным указанием обязательных чертежей)</i></p>   | <ol style="list-style-type: none"><li>1. Блок схема схемы данных;</li><li>2. Блок схема программного комплекса поддержки стандарта (сервера);</li><li>3. Схему базы данных (фрагмент для бурения);</li><li>4. Алгоритмы формирования схемы данных;</li><li>5. Алгоритм обмена данными с агентами и моделями;</li><li>6. Схема применения стандарта- обзорная;</li></ol>  |
| <p><b>Консультанты по разделам выпускной квалификационной работы</b></p>   |  |

|  |                    |
|--|--------------------|
| <i>(с указанием разделов)</i>  |                    |
| <b>Раздел</b>  | <b>Консультант</b> |
|  |                    |
| <b>Названия разделов, которые должны быть написаны на русском и иностранном языках:</b>  |                    |
| <ol style="list-style-type: none"> <li>1. Анализ стандартов LAS, WITS, WITSML</li> <li>2. Цель создания «агрегатора»</li> <li>3. Анализ преимуществ использования агрегатора</li> <li>4. Организация очереди на конвертирование</li> </ol> |                    |

|   |  |
|---|--|
| <b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b> |  |
|---|--|

**Задание выдал руководитель:**

| Должность                          | ФИО                       | Ученая степень, звание | Подпись | Дата |
|------------------------------------|---------------------------|------------------------|---------|------|
| Заведующий лабораторией кафедры ПИ | Марчуков Артур Викторович |                        |         |      |

**Задание принял к исполнению студент:**

| Группа | ФИО   | Подпись | Дата |
|--------|---|---------|------|
| 8К31   | <ol style="list-style-type: none"> <li>1. Власкин Г.Е.</li> <li>2. Прохоренко А.С.</li> </ol> |         |      |

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

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

Форма представления работы:

|                     |
|---------------------|
| бакалаврская работа |
|---------------------|

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

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

|  |                 |
|--|-----------------|
| Срок сдачи студентом выполненной работы: | 15 июня 2017 г. |
|--|-----------------|

| Дата контроля | Название раздела (модуля) / вид работы (исследования)   | Максимальный балл раздела (модуля) |
|---------------|---|------------------------------------|
| 11.03.2017    | <i>Раздел 1. Цель разработки стандартов передачи данных и их роль в построении геологической и гидродинамической моделей месторождения. Стандарты передачи данных. Формирование и организация передачи данных с мест бурения скважин в режим реального времени.</i> | 10                                 |
| 13.04.2017    | <i>Раздел 2. Цель создания «агрегатора», назначение, основные функции, структура «агрегатора», блок схема. Передача данных в режиме реального времени.</i>  | 15                                 |
| 18.05.2017    | <i>Раздел 3. Анализ доступных для использования в блоке хранения БД. Обоснование выбора. Преимущества и недостатки. Реализация выбранной технологии.</i>  | 10                                 |
| 05.06.2017    | <i>Раздел 4. Организация очереди на конвертирование. Алгоритмы подготовки файлов на конвертирование.</i>  | 30                                 |
| 08.06.2017    | <i>Раздел 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.</i>   | 10                                 |
| 08.06.2017    | <i>Раздел 6. Социальная ответственность.</i>  | 10                                 |
| 14.06.2017    | <i>Заключение</i>   | 15                                 |

Составил преподаватель:

| Должность                       | ФИО           | Ученая степень, звание | Подпись | Дата |
|---------------------------------|---------------|------------------------|---------|------|
| Заведующий лабораторией каф. ПИ | Марчуков А.В. |                        |         |      |

**СОГЛАСОВАНО:**

| Зав. кафедрой | Иванов М.А. | Ученая степень,<br>звание | Подпись | Дата |
|---------------|-------------|---------------------------|---------|------|
| ПИ            | Иванов М.А. | К.Т.Н.                    |         |      |



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

Студентам:

| Группа | ФИО   |
|--------|---|
| 8К31   | Власкину Глебу Егоровичу<br>Прохоренко Ангелине Сергеевне |

|                            |                             |                                  |                        |
|----------------------------|-----------------------------|----------------------------------|------------------------|
| <b>Институт</b>            | <b>Институт кибернетики</b> | <b>Кафедра</b>                   | Программной инженерии  |
| <b>Уровень образования</b> | бакалавриат                 | <b>Направление/специальность</b> | Прикладная информатика |

| <b>Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:</b>                                  |  |
|--|--|
| 1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих | Система хранения данных-400522;<br>Сервер Depo Storage 3436 – 919182;<br>З/п руководителя – 13771,4;<br>З/п студента 1 – 7250,6;<br>З/п студента 2 – 7250,6. |
| 2. Нормы и нормативы расходования ресурсов   | Норматив доп. з/п – 12%  |
| 3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования                                  | Отчисления во внебюджетные фонды – 27,1  |

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

|  |  |
|--|--|
| 1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения | Оценка конкурентоспособности, рассмотрение альтернатив проведения НИ.                        |
| 2. Планирование и формирование бюджета научных исследований  | Планирование этапов разработки программы, определение трудоемкости, формирование бюджета НИ. |
| 3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования        | Сравнительный анализ интегральных показателей эффективности.                                 |

**Перечень графического материала (с точным указанием обязательных чертежей):**

|  |
|--|
| <ol style="list-style-type: none"> <li>1. Оценка конкурентоспособности технических решений</li> <li>2. SWOT анализ, QuaD технология</li> <li>3. График проведения и бюджет НИ</li> <li>4. Оценка ресурсной, финансовой и экономической эффективности НИ</li> </ol> |
|--|

|   |            |
|---|------------|
| <b>Дата выдачи задания для раздела по линейному графику</b> | 15.04.2017 |
|---|------------|

**Задание выдал консультант:**

| <b>Должность</b> | <b>ФИО</b>       | <b>Ученая степень,<br/>звание</b> | <b>Подпись</b> | <b>Дата</b> |
|------------------|------------------|-----------------------------------|----------------|-------------|
| Доцент           | Тухватулина Л.Р. | к.ф.н.                            |                |             |

**Задание приняли к исполнению студенты:**

| <b>Группа</b> | <b>ФИО</b>   |
|---------------|--|
| 8К31          | Власкин Глеб Егорович<br>Прохоренко Ангелина Сергеевна |

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»**

Студентам:

| Группа | ФИО   |
|--------|---|
| 8К31   | Власкину Глебу Егоровичу<br>Прохоренко Ангелине Сергеевне |

|                            |                      |                                  |                        |
|----------------------------|----------------------|----------------------------------|------------------------|
| <b>Институт</b>            | Институт кибернетики | <b>Кафедра</b>                   | Программной инженерии  |
| <b>Уровень образования</b> | бакалавриат          | <b>Направление/специальность</b> | Прикладная информатика |

| <b>Исходные данные к разделу «Социальная ответственность»:</b>  |  |
|---|--|
| 1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения  | Объект исследования – национальный стандарт передачи данных для нефтяной и газовой промышленности, совместимый с международным стандартом WITSML 2.0; Интерфейс взаимодействия пользователя с сервером хранения и обработки данных, использующим национальный стандарт.  |
| <b>Перечень вопросов, подлежащих исследованию, проектированию и разработке:</b>   |  |
| 1. Производственная безопасность.<br>1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения.<br>1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения. | Анализ рабочей среды:<br><ul style="list-style-type: none"> <li>• Освещенность рабочей зоны</li> <li>• Микроклимат</li> <li>• Уровень шума на рабочем месте</li> <li>• Электромагнитные излучения</li> </ul> Анализ выявленных опасных факторов:<br>- Статическое электричество<br>- Короткое замыкание<br>- Пожароопасность |
| 2. Экологическая безопасность.  | - Анализ негативного воздействия на окружающую природную среду:<br>утилизация компьютеров и другой оргтехники  |
| 3. Безопасность в чрезвычайных ситуациях.   | Возможные чрезвычайные ситуации:<br>- Пожар  |
| 4. Правовые и организационные вопросы обеспечения безопасности.   | Организация рабочих мест с электронно-вычислительными машинами регулируется СанПиНом 2.2.2/2.4.1340 – 03   |

|   |            |
|---|------------|
| <b>Дата выдачи задания для раздела по линейному графику</b> | 15.02.2017 |
|---|------------|

**Задание выдал консультант:**

| <b>Должность</b> | <b>ФИО</b>                      | <b>Ученая степень, звание</b> | <b>Подпись</b> | <b>Дата</b> |
|------------------|---------------------------------|-------------------------------|----------------|-------------|
| Доцент           | Пустовойтова<br>Марина Игоревна | к.х.н.                        |                |             |

**Задание приняли к исполнению студенты:**

| <b>Группа</b> | <b>ФИО</b>   | <b>Подпись</b> | <b>Дата</b> |
|---------------|--|----------------|-------------|
| 8К31          | Власкин Глеб Егорович<br>Прохоренко Ангелина Сергеевна |                |             |

## Оглавление

|   |    |
|---|----|
| Реферат .....   | 16 |
| Обозначения и сокращения .....  | 18 |
| Введение .....  | 19 |
| Раздел 1. Цель разработки стандартов передачи данных и их роль в построении геологической и гидродинамической моделей месторождения. Стандарты передачи данных. Формирование и организация передачи данных с мест бурения скважин в режим реального времени. .... | 21 |
| 1.1. Стандартизация передачи данных. Преимущества использования стандартов передачи данных для нефтегазовой промышленности. История стандартов. ....  | 23 |
| 1.2. Организация формирования и передачи данных с нефтегазовых месторождений России на данном этапе. Перспективы развития. ....   | 25 |
| 1.3. Описание существующей модели передачи данных с использованием международных стандартов LAS WITS на нефтегазодобывающих предприятиях. Недостатки использование текущей модели. ....   | 26 |
| 1.4. Описание стандартов LAS, WITS, WITSML. Преимущества и недостатки. Недостатки существующей технологии передачи данных с месторождений - отсутствие режима реального времени. ....   | 30 |
| 1.4.1. Описание стандарта LAS .....   | 31 |
| 1.4.2. Описание стандарта WITS. ....  | 32 |
| 1.4.3. Описание стандарта WITSML. ....  | 34 |
| 1.5. Пути решения существующей проблемы стандартизации данных. ....   | 36 |
| Раздел 2. Цель создания «агрегатора», назначение, основные функции, структура «агрегатора», блок схема. Передача данных в режиме реального времени. ....  | 38 |
| 2.1. Методы и технологии сбора, хранения и обработки информации в центрах моделирования месторождений. ....   | 38 |

|   |    |
|---|----|
| 2.1.1. Краткий обзор и описание применяемых в России станций ГТИ-функции, состав, стандарты. Недостатки.....  | 40 |
| 2.3. Цель создания «агрегатора». Анализ уже существующих решений.....   | 45 |
| 2.4. Описание процесса передачи данных с использованием «агрегатора», структура «агрегатора».....   | 48 |
| 2.5. Преимущества использования стандартов WITSML 1.3.1 и WITSML 1.4.1 при реализации передачи данных в режиме реального времени.....                   | 51 |
| 2.6. Основная идея использования блока приема и хранения данных в стандартах LAS и WITS.....  | 53 |
| 2.6.1 Программа и методы обмена данными между станциями ГТИ и «агрегаторами».....   | 56 |
| 2.7. Преимущества использования «агрегатора» при приёме данных с мест бурения.....  | 63 |
| Раздел 3. Анализ доступных для использования в блоке хранения СУБД. Обоснование выбора. Преимущества и недостатки. Реализация выбранной технологии..... | 65 |
| 3.1. Описание существующих СУБД.....  | 65 |
| 3.2. Выбор технологии MS SQL Server. Преимущества использования.....  | 68 |
| 3.3. Реализация БД в блоке хранения.....  | 71 |
| 3.4. Блок схема и алгоритм работы базы данных.....  | 74 |
| 3.5. Разработка запросов к БД.....  | 75 |
| 3.6. Веб интерфейс «агрегатора».....  | 79 |
| Раздел 4. Организация очереди на конвертирование. Алгоритмы подготовки файлов на конвертирование.....   | 83 |
| 4.1 Организация очереди на конвертирование (метод FIFO).....  | 83 |
| Раздел 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....  | 87 |
| 5.1. Введение.....  | 87 |

|  |     |
|--|-----|
| 5.2. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения..... | 87  |
| 5.2.1. Потенциальные потребители продукта исследования .....   | 87  |
| 5.2.2. Анализ конкурентов.....   | 88  |
| 5.2.3. Технология QuaD .....   | 90  |
| 5.2.4. SWOT-анализ.....  | 91  |
| 5.3. Нахождение альтернативных способов проведения научных исследований .....  | 95  |
| 5.4. Создание научно-исследовательских работ .....   | 97  |
| 5.4.1. Этапы планирования работ в рамках научного исследования.....  | 97  |
| 5.4.2. Расчет трудоемкости исполнения работ.....   | 98  |
| 5.4.3. Бюджет научно-технического исследования .....   | 103 |
| Вывод раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.....  | 110 |
| Раздел 6. Социальная ответственность .....   | 112 |
| 6.1. Введение.....   | 112 |
| 6.2. Производственная безопасность .....   | 113 |
| 6.2.1. Недостаточная освещенность рабочей зоны.....  | 113 |
| 6.2.2. Отклонение показателей микроклимата.....  | 116 |
| 6.3. Экологическая безопасность.....   | 121 |
| 6.4. Правовые и организационные вопросы обеспечения безопасности .....   | 122 |
| 6.5. Безопасность в чрезвычайных ситуациях.....  | 123 |
| 6.6. Организационные мероприятия обеспечения безопасности .....  | 124 |
| 6.7. Выводы раздела «Социальная ответственность» .....   | 125 |
| Заключение.....  | 126 |
| Список использованных источников: .....  | 127 |
| Приложения .....   | 129 |
| Приложение 1. Программный код конвертера.....  | 129 |

## Реферат

Выпускная квалификационная работа 143 с., 26 рис., 21 табл., 20 источников, 1 прил.

Ключевые слова: ГТИ, стандарт передачи данных, LAS, WITS, WITSML, блок конвертации данных, режим реального времени, бурение, месторождение скважин.

Объектом исследования выпускной квалификационной работы являются: методы приема, хранения, конвертации и передачи данных в режиме реального времени. Международные стандарты передачи данных, используемые в нефтегазодобывающей отрасли (LAS, WITS, WITSML). Технология преобразования LAS, WITS в WITSML. Методы передачи данных со станций ГТИ в систему моделирования месторождения.

Целью работы является создание программного решения, позволяющего принимать, хранить и конвертировать данные о бурении, поступающие со станции геолого-технологических исследований, в режиме реального времени. Разработка блока конвертера, реализующего хранение информации в БД. Реализация программного продукта, содержащего веб интерфейс базы данных.

В процессе исследования был проведен анализ стандартов передачи данных LAS, WITS, а также WITSML версией 1.3.1 и 1.4.1., изучены уже существующие программные решения проблемы конвертации данных о бурении в режиме реального времени.

С точки зрения экономики, создание отечественного блока конвертации данных позволит существенно сократить расходы денежных и временных ресурсов бурения и нефтедобычи. Помимо этого, уменьшится вероятность потери или задержки актуальных сведений с датчиков, что значительно снижает случаи несвоевременной реакции на произошедшие отклонения и, как следствие, сократятся затраты на ремонт и устранение последствий аварий.

В результате исследования была систематизирована информация о международных стандартах передачи данных, используемых в



нефтегазодобывающей отрасли, было создано программное решение, позволяющее принимать, хранить и обрабатывать поступающую информацию, а также организовывать очередь на конвертацию данных из стандартов LAS и WITS в стандарты WITSML 1.3.1. и 1.4.1. в режиме реального времени.

Данный продукт разрабатывался с помощью PyCharm, язык реализации python.

Областью применения исследования являются нефтегазодобывающие компании, бурильные сервисные компании, а также поставщики оборудования и программно-аппаратных решений для данных компаний.

В будущем планируется разработка дополнительных программных модулей, управляющие процессом конвертирования и обработки информации.

## Обозначения и сокращения

ИМ (интеллектуальное месторождение) - Интеллектуальное нефтегазовое месторождение — система автоматического управления операциями по добыче нефти и газа, предусматривающая непрерывную оптимизацию интегральной модели месторождения и модели управления добычей.

WITSML - стандартный язык разметки передачи информации с буровой.

ГТИ - станция геолого-технологических исследований скважин.

ПК – персональный компьютер.

БД – база данных.

LAS – Log ASCII Standart.

УИ – уникальный идентификатор.

ОС – операционная система сервера, контроллера и пр.

СУБД - система управления базой данных.

SCADA-система - (аббр. от англ. supervisory control and data acquisition, диспетчерское управление и сбор данных) — программный пакет, предназначенный для разработки или обеспечения работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления. SCADA может являться частью АСУ ТП, АСКУЭ, системы экологического мониторинга, научного эксперимента, автоматизации здания и т. д.

XML- спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров (программ, читающих XML-документы и обеспечивающих доступ к их содержимому).

АПК – аппаратно программный комплекс.

SOAP – протокол обмена структурированными сообщениями.

ОЗУ – оперативное запоминающее устройство.

## Введение

В настоящее время крупные нефтедобывающие мировые компании вкладывают огромные средства на поддержку и разработку единых стандартов передачи данных, так как это помогает обрабатывать и анализировать поступающие со станций ГТИ данные в режиме реального времени, таким образом получив несколько минут назад данные можно увидеть геологическую модель месторождения. Такая оперативность способствует делать процесс бурения скважин более точным и безопасным.

Благодаря консорциуму Energestics, который был создан для того, чтобы объединить людей, проблемы и идеи, чтобы облегчить обмен информацией по E&P и интеграцию бизнес-процессов на сегодняшний день имеется уникальный стандарт передачи данных WITSML, который может хранить до 7500 атрибутов данных о бурении, а используя специальные программные средства, такие как Petrel и Геонафт можно строить геологические модели месторождений в режиме реального времени, используя данные в формате WITSML.

Использование такого способа передачи данных снижает риски возникновения внештатных ситуаций за счет повышения точности построения геологических моделей месторождений путем реализации режима реального времени, а также минимизирует не только информационные, но и ресурсные потери компании. Одной из таких технологий, позволяющих проводить активный мониторинг процесса бурения в режиме реального времени с минимальными задержками отправки данных, является стандартизация передачи данных и переход компании на современные стандарты передачи данных, такие как WITSML версии 1.3.1 и 1.4.1.

Реализация режима реального времени требуется для оперативного анализа процесса бурения скважины и построения геологической модели месторождения, что в свою очередь актуализирует проблему стандартизации передачи данных и приведению их к современным стандартам передачи данных. Большинство

отечественных и зарубежных станций ГТИ используемых российскими нефтяниками работают с данными в стандартах WITS и LAS, которые имеют плоскую структуру и не могут быть использованы для построения математических моделей месторождения в режиме реального времени. В то же время, закупать станции ГТИ, которые имеют современный программный агент формирующий данные в стандартах WITSML версии 1.3.1 и 1.4.1.

у зарубежных поставщиков достаточно дорого, а также в настоящее время действуют санкции, не позволяющие сделать это. Поэтому возникла идея агрегировать данные со старых ГТИ и конвертировать их в современный стандарт WITSML, тем самым, не нарушая регламент обеспечивать режим реального времени. Отечественная реализация сервера WITSML выполнена сотрудниками лаборатории сетей ЭВМ ранее. На данный момент в России нет станций ГТИ, которые позволяли бы передавать данные на сервер в стандарте WITSML, а также обрабатывать их в кратчайшие сроки, обычно задержки в обработке данных составляют от 3 часов. Это достаточно большие временные потери, которые могут привести к различным негативным последствиям, а также к финансовым и ресурсным потерям. Исходя из этого обработка данных в режиме реального времени является актуальной задачей. Это позволит сократить финансовые потери, связанные с возможными рисками в процессе бурения, а также повысит точность построения геологической модели месторождения, так как будет минимизирован факт участия непосредственно человека в обработке поступающих с месторождения данных. Решение данной задачи заключается в разработке и реализации АПК «Агрегатор», который позволяет использовать старые стандарты передачи данных, такие как LAS и WITS для реализации режима реального времени, сохраняя тем самым существующие концепции обработки данных в российских компаниях, а также хранения полученных со станций ГТИ данных в стандартах LAS и WITS для аварийных случаев.

## **Раздел 1. Цель разработки стандартов передачи данных и их роль в построении геологической и гидродинамической моделей месторождения. Стандарты передачи данных. Формирование и организация передачи данных с мест бурения скважин в режим реального времени.**

В настоящее время интеллектуализация всех сфер деятельности человека развивается все быстрее, и газо- и нефтедобывающая отрасль не исключение. Россия занимает одно из первых мест среди всех стран в данной отрасли, однако и отечественный промысел не поспевает за новыми технологиями ввиду того, что замена оборудования на более современное очень затратно, и пока не все компании могут себе это позволить.

Существует три международных стандарта передачи данных о бурении скважин: LAS, WITS, WITSML. Наиболее популярными стандартами в России являются LAS версии 2.0 и 3.0 и WITS 1.0. В зарубежных странах в основном используется WITSML 1.3.1, однако существует все ещё множество станций ГТИ со стандартами обмена LAS версии 2.0 и 3.0 и WITS 1.0. Оба стандарта формируются в текстовых файлах и передаются на станцию управления бурением, где сотрудники компании вручную проверяют их, корректируют при необходимости, и отправляют в центр моделирования месторождения через почтовый ящик. Таким образом, передача информации с датчиков в центр моделирования месторождений происходит с частотой раз в 30-40 минут. Помимо этого, принцип записи в каждом из стандартов существенно отличается, необходимо время, чтобы найти необходимую информацию (учитывая, что текстовый формат стандартов не позволяет создать базу данных, которая бы существенно облегчила поиск). Поэтому время реакции на какие-либо изменения в процессе бурения составляет не менее 2-3 часов. За это время бур может уйти на 10-12 метров вглубь. В зависимости от ситуации коррекция направления бура может отнять различное количество финансовых и временных ресурсов.

С появлением современной цифровой технологии – оптоволоконных датчиков возникает проблема. Количество получаемых данных, как и значение

разрешающей способности значительно увеличились. С увеличением числа датчиков и количества параметров, старые форматы не способны формировать в несколько раз увеличившийся объем данных.

Следом возникла еще одна проблема – при использовании стандартов LAS или WITS очень сложно построить объемную модель месторождения, так как для этого необходимо большое множество различных параметров, при этом часть параметров должна повторяться. Для каждого набора параметров следует заново записывать значения параметров, коды, типы и т.д., что абсолютно не рационально. В сложившейся ситуации единственным выходом является использование иерархических структур данных.

Было предпринято несколько попыток усовершенствования WITS, однако проблемы не были решены. Тогда возникло понимание, что необходим совершенно новый стандарт, который удовлетворял бы современным условиям. Так был создан стандарт WITSML в формате .XML. Использование нового формата позволяет хранить данные в иерархическом виде и значительно увеличить количество передаваемых данных.

В связи с тем, что в России станции ГТИ в основном используют стандарты обмена данными LAS версии 2.0 и 3.0 и WITS 1.0, а моделирующие средства типа Petrel и Геонафт уже перешли на технологии обмена данными с ГТИ с использованием стандарта WITSML 1.3.1 и 1.4.1, возникла необходимость в программном комплексе, который будет принимать данные со станции ГТИ в стандартах LAS версии 2.0 и 3.0 и WITS 1.0 и конвертировать их в стандарты WITSML 1.3.1 и 1.4.1. Использование данного программного комплекса предоставит возможность построения абстрактной модели разрабатываемого месторождения, а также сократить времена передачи данных с 2-3 часов до 3-4 минут (режим real-time), при этом не нарушая регламент передачи данных, так как новое программное решение будет функционировать параллельно со старой схемой передачи данных. Это позволит сравнить два варианта передачи данных, выявить преимущества и недостатки каждой из них не только в теории, но и на практике, не нарушая при этом непрерывный процесс передачи данных.

## 1.1. Стандартизация передачи данных. Преимущества использования стандартов передачи данных для нефтегазовой промышленности. История стандартов.

Интерпретация геологических исследований предполагает интенсивный обмен информацией между программными системами различных производителей. Для успешного обмена информацией каждая из программных систем должна суметь открыть любой файл, созданный другой системой. Это возможно с помощью двух вариантов – либо каждая система имеет возможность открыть любой формат других систем, либо создается единый формат для всех программных систем нефтегазовой отрасли. Последний вариант наиболее удобный, так как легче создать единый стандарт передачи данных, чем каждый адаптировать систему для того, чтобы она могла понять новый формат.

Одним из первых стандартов для хранения, передачи и обмена каротажной информации для любой системы был стандарт LAS (Log ASCII Standard). Данный формат разработали под руководством Канадского общества каротажников (Canadian Well Logging Society's Floppy Disk Committee) в 90х годах. Стандарт LAS является текстовым файлом, в котором записываются необходимые геофизические данные по специальным правилам. На данный момент существует три версии стандарта: 1.2, 2.0 и 3.0. Наиболее популярными версиями являются 2.0, 3.0.

Еще одним стандартом в текстовом формате является WITS (Wellsite Information Transfer Specification) – телекоммуникационный протокол передачи данных бурения. Он был разработан в 1980х годах для обмена информацией с буровыми и станциями ГТИ. Существует 5 вариаций этого стандарта, однако наиболее популярной считается WITS 1.0.

Хотя оба стандарта и имеют довольно широкое распространение и долгую историю, у них существует ряд недостатков:

- Устаревший формат записи;
- Небольшое, по современным меркам, число записей;
- Отсутствие полей для записи новой номенклатуры;

- Практически абсолютное отсутствие справочной информации о разрабатываемой скважине;
- Использование устаревшего способа передачи данных точка-точка (RS-232);
- Невозможность реализации режима real-time;
- На основе стандартов LAS и WITS имеется возможность построения только «плоской» модели месторождения.

Номенклатура геофизических методов постоянно растет за счет появления современных датчиков, различной аппаратуры и технологий. В настоящее время самым популярным способом передачи геофизических данных являются стандартизированные текстовые файлы. Для успешного обмена файлами для каждой двух взаимодействующих систем необходимо соотнести схему данных источника со схемой данных получателя. Однако текстовые файлы имеют ограниченное число полей для записи данных с датчиков, а также возникают проблемы связи между различными системами, так как не все записи в каждом из стандартов являются строго регламентированными, каждая компания адаптирует стандарт для собственного использования.

Для решения различных проблем в нефтегазовой отрасли, одной из которых является проблема несовместимости структур данных различных производителей промышленного оборудования, в 1990 году был создан международный консорциум Energistics (POSC, с 2006 года - Energistics). Энерджистикс разработал стандарт WITSML, который основан на языке XML. Новый стандарт позволил снимать более 7500 различных параметров, а также имеет возможность неограниченно увеличивать количество номенклатуры. Это позволяет расширить область применения стандарта до месторождений, которые имеют очень сложную структуру. Использование стандарта WITSML позволяет реализовать режим реального времени, когда данные со скважины передаются каждые 3-4 минуты. Существует несколько версий стандарта: 1.3.1, 1.4.1, 2.0.

Помимо WITSML, предназначенного для передачи данных с буровой, было создано еще два стандарта, написанных на языке XML – PRODML, который



формирует данные о процессе добычи углеводородов и RESQML, который передает информацию для построения геологической модели месторождения. Эти три стандарта позволяют на порядок увеличить детализацию описания объектов, создавать объемную модель месторождения в режиме реального времени. Повышается скорость и качества принятия различных решений о процессе добычи.

## **1.2. Организация формирования и передачи данных с нефтегазовых месторождений России на данном этапе. Перспективы развития.**

Современное состояние нефтеразведки и добычи характеризуется множественной разработкой скважин, при разработке которых необходим постоянный контроль траектории бурения участка. Для контроля используется большое количество разнообразных датчиков, которые снимают различные показатели ствола скважины.

В России в 1990х годах было создано несколько систем контроля процесса бурения. Зачастую они состояли из датчиков, компьютера и информационного табло. Основной задачей системы было контролировать процесс бурения на месте. Компьютер проводил анализ полученной информации с датчиков, анализировал ее и сообщал, если происходили отклонения от нормальных показателей. Оператор имел возможность распечатать визуальный отчет, сформированный собранными данными.

Однако для обеспечения оперативного контроля разработки месторождения необходимо было организовать быструю передачу данных со скважины на предприятие. Таким образом, следующим шагом передачи данных стала реализация возможности передачи данных на локальные уровни иерархии управления бурением. Каналами передачи данных при этом были сотовая, спутниковая или телефонная связь. С появлением нового способа передачи данных специалисты, находящиеся за много километров от месторождения, могли получать информацию о процессе бурения, которые были получены со скважины всего несколько часов назад.

В настоящее время, с интенсивным развитием информационных технологий, возникла концепция «интеллектуального месторождения», которая подразумевает под собой создание высокоэффективных систем мониторинга, управления процессом разработки и отправки данных. Эта концепция включает в себя создание таких систем мониторинга, которые позволяли бы управлять разработкой и передачей данных, учитывать структурные особенности нефтяных и газовых месторождений и обеспечивать значительное сокращение производственных издержек.

Одной из таких функций является оперативная обработка данных, а также моделирование состояния резервуаров. Такая модель работы месторождения открывает новые возможности для переосмысления процесса построения геологических моделей. В большинстве российских компаний по разным причинам не получается воссоздать модель «интеллектуального» месторождения, зачастую это связано с устаревшим оборудованием и уже устаревающим внутренним правилами обработки данных. Создание «агрегатора» позволит решить проблему с заменой оборудования, позволит частично воплотить концепцию «интеллектуального» месторождения на давно работающих предприятиях.

### **1.3. Описание существующей модели передачи данных с использованием международных стандартов LAS WITS на нефтегазодобывающих предприятиях. Недостатки использования текущей модели.**

На текущий момент ведущие большинство отечественных компаний имеют сложившийся строгий регламент, определяющий порядок формирования, передачи, обработки и хранения данных со станций бурения. В большинстве случаев эти станции имеют устаревшее оборудование, которое не способно работать с современными стандартами передачи данных, поэтому они используют ставшие уже классическими форматы LAS и WITS, которые в свою очередь достаточно ограничены и в настоящее время считаются устаревшими, потому что не могут передавать большее количество информации, чем заложено в основе этих форматов.

Данные о бурении формируются на станции ГТИ, путем считывания каротажной информации с различных датчиков, которые находятся непосредственно внутри скважины. Эта информация постоянно обновляется, поэтому полученные данные станция ГТИ собирает и отправляет на дальнейшую обработку, используя установленный регламентом компании канал. Данные записываются в форматы LAS и WITS, которые представляют собой текстовый файл с установленными параметрами.

После установления соединения данные принимаются группой подготовки данных в стандарте LAS и WITS перед вводом в моделирующий пакет Petrel для дальнейшей обработки. Эта группа вручную занимается проверкой полноты этих файлов, а также их подготовкой для работы с пакетом и записью данных на сервер, который работает только с унифицированным стандартом WITSML. Примерами таких пакетов являются «Petrel» – программное обеспечение, разработанное компанией Schlumberger «от сейсмики до разработки», которое предлагает пользователям использование интегрированных рабочих процессов для коллективной работы, объединяющие в единую технологическую цепочку геофизику, геологию и разработку месторождений, и открывающие путь к описанию резервуаров в режиме реального времени. Еще одним примером программного комплекса, используемого для построения геологической модели является программный комплекс «Геонафт» - предназначенный для геологического сопровождения бурения наклонно-направленных и горизонтальных скважин с целью наиболее успешной проводки ствола в целевом горизонте. ПК «Геонафт» позволяет определить стратиграфическое местоположение скважины в процессе бурения, а также спрогнозировать изменение структуры пласта.

Современные программные комплексы позволяют строить детальные статические геологические 3D модели для планирования горизонтальных скважин. Но в условиях бурения, с появлением новой информации, особую важность приобретает возможность оперативно анализировать данные

и принимать решения о корректировке траектории скважины в режиме реального времени с целью минимизации потери ствола на бурение в непродуктивной зоне.

Данные программные средства позволяют выполнять следующие задачи:

1. Построение модели геонавигации с синтетическими кривыми ГИС, которые планируется записать в процессе бурения.
2. Построение отчетов о ходе геонавигации и формирование рекомендаций на бурение.
3. Принятие решений о траектории скважины по результатам сравнения синтетического и фактического каротажей.
4. Оптимизация и корректировка траектории горизонтальной / наклонной скважины.
5. Формирование финального отчета и геонавигационной модели по окончании бурения.

Все вышеописанные задачи «Геонафт» позволяет решать со скоростью, максимально приближенной к режиму реального времени (единственное ограничение – по скорости подгрузки новых данных пользователем).

После построения обработки данных в программном комплексе полученные результаты с помощью специально написанного API отправляются на WITSML сервер, где хранятся до востребования.

Данный сервер обеспечивает следующие функции:

- хранение данных о процессе бурения в стандарте WITSML v.1.4.1;
- приём данных от станций управления бурением в стандарте WITSML v.1.4.1;
- передачу данных в геологическую модель месторождения (Petrel), в режиме on-line;
- можно создать и зарегистрировать скважину с несколькими стволами;
- хранение данных о скважине, в базе данных до 5 лет;
- количество скважин, о которых храниться информация (до 500 шт.);
- управление своими скважинами;
- информация о финансовом состоянии оплаты за хранение.

На рисунке 1 представлена схема формирования и движения данных в структуре типовой нефтедобывающей организации.

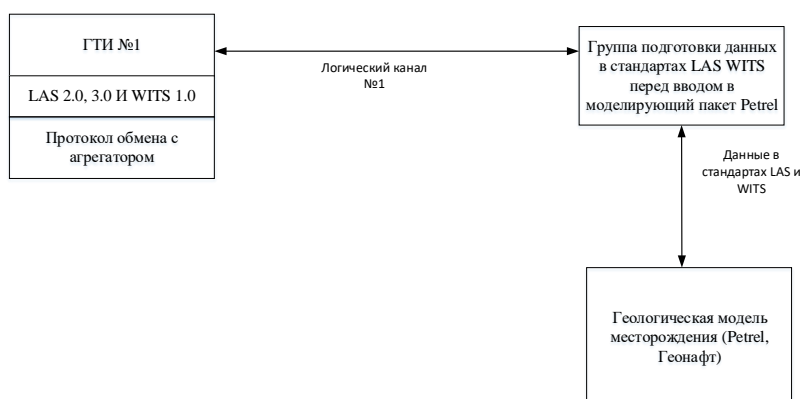


Рис. 1 - Формирование данных на станции бурения и их дальнейшая обработка

В условиях недостатка программного обеспечения, поддерживающего стандарт WITSML, приходится использовать метод конвертации данных из одного стандарта описания файла в другой. Это приводит к возникновению следующих проблем:

- 1) Сложность настройки интеграции написанного собственными силами конвертера в действующую технологическую цепочку;
- 2) Настройка конвертации данных в режиме реального времени, в условиях недостаточной скорости и стабильности сетевых каналов от месторождения до WITSML-сервера;
- 3) Соответственно, ресурсные потери, связанные с не преждевременным реагированием на различные рода внештатные ситуации, связанные с задержкой обработки данных. Задержки могут составлять от 3-4 часов.

#### **1.4. Описание стандартов LAS, WITS, WITSML. Преимущества и недостатки. Недостатки существующей технологии передачи данных с месторождений - отсутствие режима реального времени.**

Сегодня, при проведении каротажа, данные, поступающие из скважины, автоматически переводятся из аналогового в цифровой формат, и в таком виде сохраняются на носителе информации. В то же время довольно часто возникает необходимость обмена информацией. Примером такого обмена может быть передача результатов интерпретации от геофизической компании заказчику. В таких случаях очень остро встает вопрос о стандартизации предоставляемых материалов – ведь свои данные можно хранить в совершенно произвольной форме.

В связи с этим, ряд институтов по всему миру занялся разработкой стандартов для хранения, переноса и обмена каротажной информацией для всех желающих. В настоящее время известно несколько широко распространенных форматов хранения данных, как в России, так и за рубежом.

Стандарты LAS и WITS содержат наборы полей и параметров, некоторые из которых являются обязательными, а другие заполняются по желанию или требованию аналитического центра. Данные стандарты были разработаны достаточно давно, поэтому не имеют возможности работать с современными программными средствами для моделирования геологических месторождений, что в свою очередь делает невозможным работу анализа данных и построения моделей в режиме реального времени или режиме, приближенны к режиму реального времени. Также эти стандарты могут содержать в себе минимальное количество полей, отражающих только основные параметры, принимаемые с датчиков. Развитие технологий привело к тому, что такое количество полей, которое могут хранить в себе стандарты LAS и WITS становится недостаточным, потому как объемы данных возрастают, меняются требования к показателям, снимаемых с мест бурения. Проблемой является и взаимодействие программного комплекса для построения модели с современными серверами, которые хранят в себе информацию в стандарте WITSML. Поэтому данные в большинстве случаев

проходят множественные конвертации, что чревато временными потерями. Более современным стандартом является стандарт WITSML, который позволяет содержать в себе более 7500 полей, а также позволяет станции обрабатывать данные в режиме реального времени, сохраняя всю информацию на специальном WITSML сервере.

#### **1.4.1. Описание стандарта LAS**

Аббревиатура LAS является сокращением от Log ASCII Standard (ASCII – American Standard Code for Information Interchange). Формат Las был разработан под руководством специального комитета Канадского общества каротажников (Canadian Well Logging Society's Floppy Disk Committee). Изначально формат LAS создавался для переноса данных с использованием гибких магнитных дисков (1.44” и др.). Хотя в настоящее время использование подобных носителей информации уже не столь актуально, формат не утратил своей популярности. Более того, со времен его разработки в первоначальный вариант несколько раз вносились дополнения. Сегодня файл формата LAS можно записать в трех версиях: 1.2, 2.0 и 3.0.

Формат Las разрабатывался с той целью, чтобы данные, хранящиеся в файле были одинаково легко понятны рядовым пользователям и, в то же время с тем, чтобы программистам не составляло значительных затрат написание программ, способных читать и записывать информацию, следуя представленным правилам. Во-первых, файлы формата Las – это всегда текстовые файлы, которые можно легко открыть и просмотреть с помощью любого текстового редактора. Для удобства распознавания файла LAS среди всех остальных файлов, он должен иметь соответствующее расширение - “.LAS”. Las-файл состоит из нескольких разделов (секций) (рисунок 2). Порядок их размещения никак не регламентируется, за исключением того, что секция данных должна быть последней секцией в файле. Как правило, первым разделом записывается раздел “VERSION”, несущий в себе информацию об используемой версии формата. В

секции “WELL INFORMATION” заключена информация о скважине, ее имени, положении и т.п. Раздел “CURVE INFORMATION” перечисляет имена каротажных кривых, записанных в данном файле. Секции “PARAMETER” и “OTHER” не являются обязательными и не всегда присутствуют в файле. Первая из них описывает различные параметры, относящиеся к скважине (такие, как сопротивление и вязкость бурового раствора). Вторая применяется для записи комментариев. Последней секцией файла обязательно должна быть секция “ASCII LOG DATA”. Здесь записываются колонки данных, отделенных друг от друга пробелами.

```

~Version information|
  VERS.                1.20: CMES LAS - VERSION 1.20
  WRAP.                NO: One line per depth step
~Well information
# MNEM.UNIT           DATA TYPE INFORMATION
# =====|=====|=====
  STRT.M              1756.00 :
  STOP.M              1832.00 :
  STEP.M              0.20 :
  NULL.               -9999.00: Null values
  COMP.               COMPANY:
  WELL.               WELL:
  FLD .               FIELD:
  FLDN.               FIELD NAME :
  LOC .               LOCATION:
  CNTY.               COUNTY:
  STAT.               STATE:
  CTRY.               COUNTRY:
  SRVC.               SERVICE COMPANY:
  DATE.               LOG DATE: 9/13/2001
  TIME.               LOG TIME: 8:58
  METD.               METHOD: DS
~Curve information
# MNEM.UNIT           API CODE CURVE DESCRIPTION
# =====|=====|=====
  DEPTH.M .           :
  DS .                :
~Parameter information block
# MNEM.UNIT           VALUE DESCRIPTION
# =====|=====|=====
~Other information
# ----- REMARKS AREA -----
# -----
~A Depth             DS
1756.00              0.215
1756.20              0.215
1756.40              0.215
1756.60              0.215
1756.80              0.215
1757.00              0.215
1757.20              0.216
1757.40              0.223
  
```

Рис. 2 - Пример .Las файла

#### 1.4.2. Описание стандарта WITS.

WITS (Wellsite Information Transfer Specification) - телекоммуникационный протокол передачи данных бурения. Был разработан в середине 1980-х годов для



обеспечения обмена информацией на буровой, и предоставления данных станциям удалённого мониторинга бурения.

В создании принимали участие: Baker Hughes, INTEQ, Schlumberger, Anadrill, Halliburton, Sperry Sun, MD/Totco, Experience.

WITS является многоуровневым протоколом (рисунок 3), который предлагает простоту на низких уровнях и возрастающую гибкость формирования более высоких уровней репрезентации данных. На низких уровнях используется заранее определённый текстовый поток формата ASCII, в то же время на самом высшем уровне поток данных может быть настроен для любого вида представления информации.

Типы данных WITS

| №  | Наименование типа                     | Описание  |
|----|---------------------------------------|---|
| 1  | <u>General Time-Based</u>             | Данные собираемые с определённым временным интервалом         |
| 2  | <u>Drilling - Depth Based</u>         | Данные собираемые с определённым интервалом глубины           |
| 3  | <u>Drilling - Connections</u>         | Данные соединений собранные во время бурений                  |
| 4  | <u>Hydraulics</u>                     | Гидростатические данные во время циркуляции бурового раствора |
| 5  | <u>Trip - Time</u>                    | Данные собранные во время <u>спуско-подъёмных</u> операций    |
| 6  | <u>Trip - Connections</u>             | Данные собранные во время соединений при СПО                  |
| 7  | <u>Survey/Directional</u>             | Оценке направления и отклонения во время бурения              |
| 8  | <u>MWD Formation</u>                  | Оценка характеристик пласта во время бурения                  |
| 9  | <u>MWD Mechanical</u>                 | Механические данные во время бурения                          |
| 10 | <u>Pressure Evaluation</u>            | Данные оценки давлений  |
| 11 | <u>Mud Tank Volumes</u>               | Данные объёмов ёмкостей                                       |
| 12 | <u>Chromatograph Cycle-Based</u>      | Данные хроматографа по времени                                |
| 13 | <u>Chromatograph Depth-Based</u>      | Данные хроматографа по глубине(с отсрочкой получения)         |
| 14 | <u>Lagged Mud Properties</u>          | Данные оценки свойств шлама по глубине                        |
| 15 | <u>Cuttings / Lithology</u>           | <u>Шламограмма</u> , Литология и прочая информация.           |
| 16 | <u>Hydrocarbon Show</u>               | <u>Карбонатометрия</u>  |
| 17 | <u>Cementing</u>                      | Данные о цементировании                                       |
| 18 | <u>Drill Stem Testing</u>             | Данные оценки испытаний                                       |
| 19 | <u>Configuration</u>                  | Данные настройки буровой установки и бурильной установки      |
| 20 | <u>Mud Report</u>                     | Отчёт о шламе   |
| 21 | <u>Bit Report</u>                     | Отчёт о состоянии бура  |
| 22 | <u>Comments</u>                       | Комментарии в свободной форме                                 |
| 23 | <u>Well Identification</u>            | Данные идентификации скважины                                 |
| 24 | <u>Vessel Motion / Mooring Status</u> | Статус бурового судна (движение, швартовка)                   |
| 25 | <u>Weather / Sea State</u>            | Погода и состояние моря                                       |

Рис. 3 - Поля, используемые в стандарте WITS

Уровни WITS:

- Уровень 0

Также известен как IRTS(Intra Rig Transfer Specification), базовый уровень обмена текстовой информацией между сервисными компаниями.

- Уровень 1

На уровне 1 и выше, поток данных превращается в двоичный формат LIS, значения конвертируются согласно типам данных LIS. В дальнейшем данные инкапсулируются в WITS пакет, который помещается между физическими и логическими заголовками и концовками записей, что превращает его в пакет LIS.

- Уровень 2

Уровень 2 надстраивается на уровень 1 добавлением двусторонней коммуникации через записи, определённые под комментарий. Это даёт возможность обмена мгновенными текстовыми сообщениями между пользователями.

- Уровень 2b

Уровень 2b добавляет опцию буферизации переданных данных, что гарантирует доставку данных в случае их неполучения по какой-либо причине.

- Уровень 4

WITS четвертого уровня работает по совершенно другому формату чем предыдущие уровни начиная с того, что он основан на формировании передачи данных стандарта API RP66. Концепция предопределённых записей и двустороннего обмена остаются, но как механизм форматирования используется RP66

Несмотря на активное использование в настоящее время, дальнейшее развитие протокол WITS получил в основанном на XML протоколе WITSML, в 2003 году.

### **1.4.3. Описание стандарта WITSML.**

WITSML - стандарт для передачи технических данных между организациями в нефтяной промышленности. Он продолжает развиваться с помощью международного некоммерческого консорциума Energistics,

занимающегося разработкой XML - стандартов для бурения и обмена данными. Организации, которые применяют WITSML включают энергетические компании, сервисные компании, буровые подрядчики, поставщики приложений и регулирующие органы.

В 1990 году была создана Международная организация Petrotechnical Open Standards Consortium (POSC, с 2006 года - Energistics). Это международный некоммерческий консорциум, целью которого является предоставление специалистам нефтегазовых компании, занимающихся геологоразведкой и добычей, нейтральной площадки для совместной работы, продвижения открытых стандартов по обмену данными и решения вопросов, касающихся разведки и разработки.

Международный некоммерческий консорциум Energistics разрабатывает несколько направлений промышленных стандартов в области обмена данных:

- WITSML для бурения и ремонта
- PRODML для процесса добычи и передачи отчетов
- RESQML для построения геологической модели среды и пласта

Большая часть зарубежных добывающих компаний либо уже используют у себя технические решения WITSML для передачи скважинных данных в режиме реального времени или планируют это осуществить в ближайшее время. Количество продуктов, совместимых с WITSML, стремительно растет, и сама совместимость с WITSML становится требованием договоров, в особенности на новых месторождениях. WITSML позволяет обеспечить интегрирование данных и интеллектуальный сбор данных и является основополагающей технологией.

Стандарт WITSML (рисунок 4) поддерживает непрерывный поток данных сверления между источниками этих данных и их потребителями для ускорения и улучшения принятия решений в области бурения и доработки.

На данный момент существуют следующие версии стандарта:

- Версия WITSML версии 1.3.1 была выпущена в марте 2006 года. Она была заменена версией 1.3.1.1 (с исправлением ошибок), выпущенной в марте 2007 года.

- Версия WITSML 1.4.1 выпущена в сентябре 2011 года.
- Версия WITSML 2.0 выпущена в феврале 2017 года.

```

<?xml version="1.0" encoding="UTF-8"?>
<!--Example of Mud Log data From Lithology Element User Guide -->
<mudLogs xmlns = "http://www.witsml.org/schemas/1series" xmlns:xsi =
"http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation =
"http://www.witsml.org/schemas/1series ../xsd_schemas/obj_mudLog.xsd" version =
"1.4.1.0">
  <documentInfo>
    <documentName>mudLog</documentName>
    <fileCreationInformation>
      <fileCreationDate>2011-12-15T16:15:31.827Z</fileCreationDate>
      <fileCreator>John Smith</fileCreator>
    </fileCreationInformation>
  </documentInfo>
  <mudLog uidWell = "W-12" uidWellbore = "B-01" uid = "h45a">
    <nameWell>W-12</nameWell>
    <nameWellbore>B-01</nameWellbore>
    <name>Exmouth Marina #1</name>
    <geologyInterval uid = "00001">
      <typeLithology>interpreted</typeLithology>
      <mdTop uom = "ft">495</mdTop>
      <mdBottom uom = "ft">670</mdBottom>
      <chronostratigraphic kind = "era">Mesozoic</chronostratigraphic>
      <chronostratigraphic kind = "period">Triassic</chronostratigraphic>
      <chronostratigraphic kind = "epoch">Middle
Triassic</chronostratigraphic>
    </geologyInterval>
    <geologyInterval uid = "00002">
      <typeLithology>interpreted</typeLithology>
      <mdTop uom = "ft">495</mdTop>
      <mdBottom uom = "ft">645</mdBottom>

```

Рис. 4 - Пример WITSML файла

### 1.5. Пути решения существующей проблемы стандартизации данных.

Для решения проблемы конвертации данных и обработки данных в режиме реального времени без замены существующего оборудования и нарушения устоявшихся регламентов компании была разработана концепция введения дополнительного модуля, который позволял бы автоматически конвертировать данные из стандартов LAS и WITS в стандарт WITSML в режиме реального времени с минимальными задержками. При этом данный разрабатываемый модуль должен иметь функции проверки на целостность поступаемых данных, а также возможность хранить исходные файлы в базе данных, чтобы в случае необходимости была возможность делать выборку по этим файлам, используя определенные параметры.

Разрабатываемый модуль носит название «агрегатор», и выполняет функции получения, проверки, хранения и конвертации данных с последующей передачей

этих данных на WITSML сервер. На рисунке 5 показан вариант внедрения «агрегатора» без внесения существенных изменений в уже существующую схему передачи данных, то есть не нарушаются устоявшиеся методы обработки данных компании.

Таким образом использование «агрегатора» позволяет снизить общее время задержки с 3-4 часов до 4-5 минут, что приближает обработку данных к режиму реального времени (конвертация 50 мегабайт данных = 10 секунд, один las отчет в 50 мегабайт может содержать данные за 1 месяц бурения).

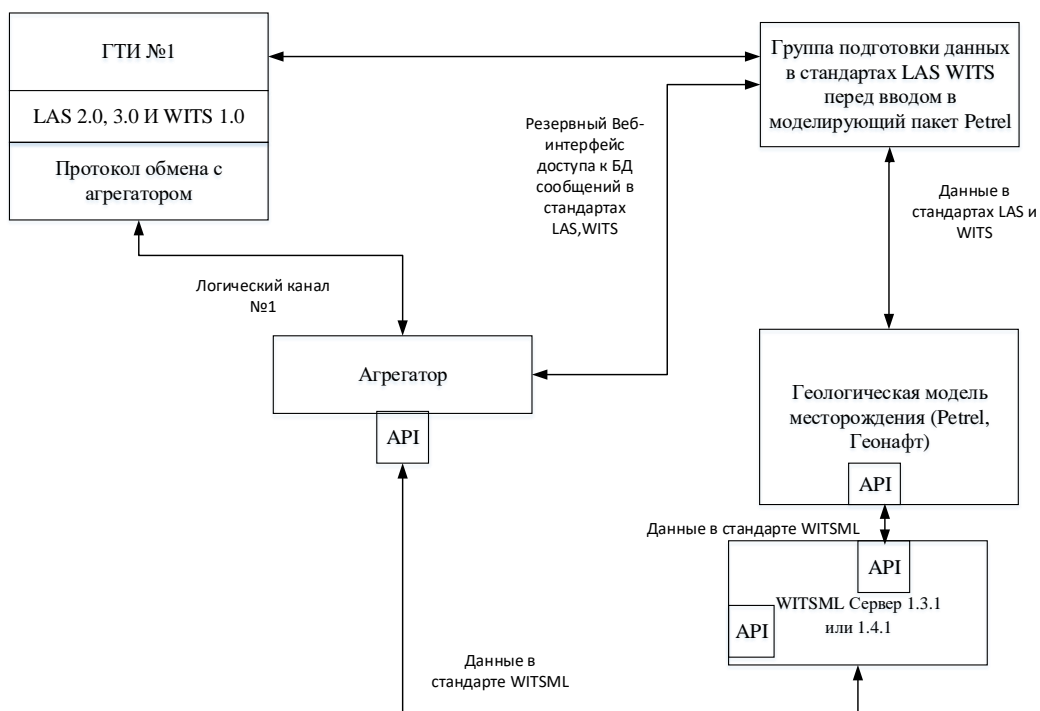


Рис. 5 - Новая модель передачи данных, с использованием "агрегатора"

## Раздел 2. Цель создания «агрегатора», назначение, основные функции, структура «агрегатора», блок схема. Передача данных в режиме реального времени.

### 2.1. Методы и технологии сбора, хранения и обработки информации в центрах моделирования месторождений.

В рамках исследуемой области нужно сравнить опыт сбора, хранения и обработки информации отечественных и зарубежных центров моделирования месторождений, чтобы выявить основные недостатки и преимущества каждого метода, выделить проблемные области и предложить решение существующих проблем.

Рассмотрим схему сбора, хранения и обработки информации отечественных центров моделирования месторождений, которая осуществляется и на сегодняшний день в большинстве российских компаниях (рисунок 6). Плюсом такой схемы является малый объем передаваемых данных, а главным недостатком – невозможность обработки данных в режиме реального времени.

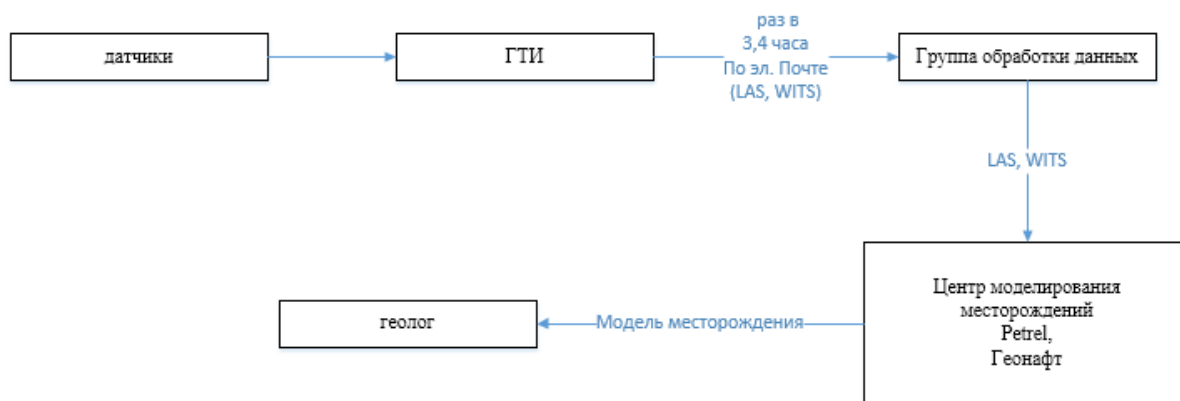


Рис. 6 - схема сбора, хранения и обработки информации отечественных центров моделирования месторождений

В некоторых зарубежных компаниях (рисунок 7) давно используется более совершенная схема сбора, хранения и обработки информации центров

моделирования месторождений, которая также имеет свои достоинства и недостатки. К плюсам такой схемы можно отнести оперативность передачи данных, а к минусам реализации схемы в отечественных компаниях - требование высокоскоростных каналов передачи данных, практически отсутствие датчиков, формирующих данные в формат WITSML в отечественных компаниях и высокая стоимость замены оборудования, а также различные политические запреты (эмбарго) на поставку оборудования.



Рис. 7 - схема сбора, хранения и обработки информации зарубежных центров моделирования месторождений

Для решения основной проблемы, а именно: передачи данных в режиме реального времени, что очень важно для современной компании, при этом не меняя основной концепции передачи данных и учитывая специфику отечественных компаний без замены существующего и используемого на сегодняшний день оборудования была предложена дополненная схема сбора, хранения и обработки информации зарубежных центров моделирования месторождений (рисунок 8), включающая в себя использование специального преобразователя «агрегатора», а также современного WITSML сервера, которые приблизят отечественные компании к обработке данных в режиме реального времени.

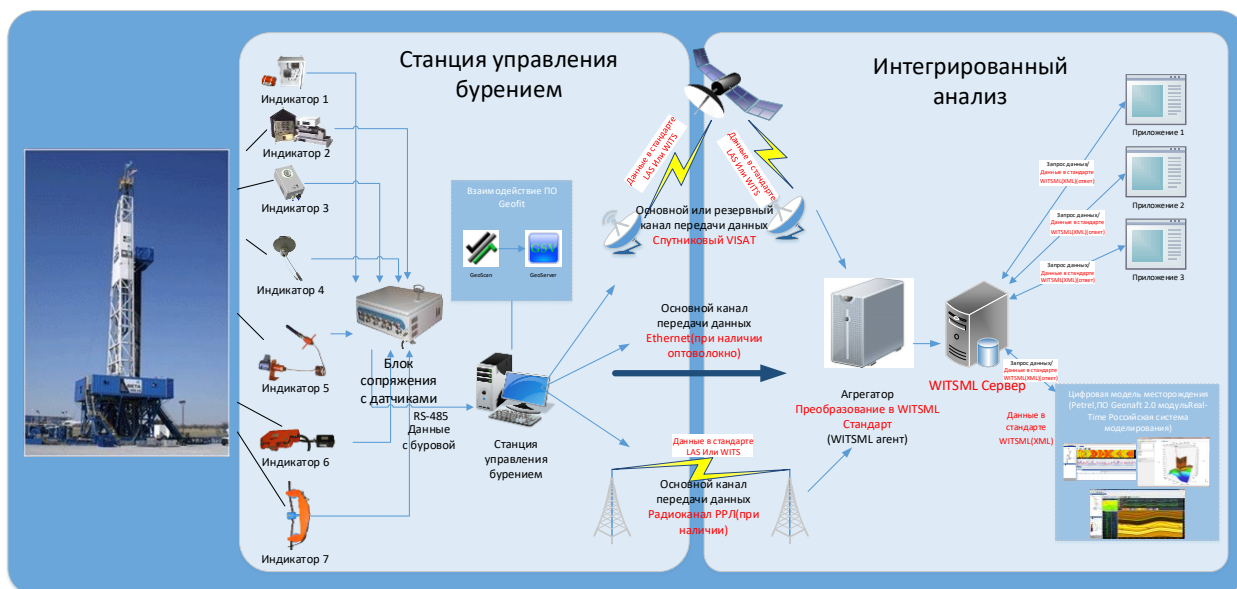


Рис. 8 - Дополненная схема сбора, хранения и обработки данных, с использованием "агрегатора"

### 2.1.1. Краткий обзор и описание применяемых в России станций ГТИ-функции, состав, стандарты. Недостатки.

ГТИ (геолого-технологические исследования) осуществляют контроль за состоянием скважин на протяжении всех этапов ее разработки и ввода в эксплуатацию. Геолого-технологические исследования являются одной из частей исследования нефтяных и газовых скважин. Исследования проводятся непрерывно во время процесса бурения скважины, без простоя оборудования и бригады. Это позволяет оперативно собирать информацию о состоянии скважины, наличии перспективных на нефть и газ участков месторождения, а также передавать информацию в центр моделирования месторождений.

Операторы ГТИ, находящиеся на станции, получают отчеты с датчиков скважины, в которых находится вся информация о процессе бурения и состоянии ствола скважины. Они обрабатывают и анализируют полученные данные, на основании которых корректируют процесс бурения, предотвращают различные аварийные ситуации, предотвращают осложнения процесса проводки скважин, производят корректировку плотности промывочной жидкости.



Неоднократно ГТИ открывали неизвестные ранее залежи углеводорода. Теперь ГТИ это не только вспомогательные инструменты для геолога, но и самый сложный автоматизированный комплекс, оперативно решающий множество различных задач. Одной из главных задач ГТИ является снижение затрат на процесс бурения за счет предотвращения аварий. Станции позволяют не только узнавать информацию непосредственно на месторождении, но и передавать отчеты и проводить мониторинг удаленно.

С целью увеличения эффективности интерпретации информации данные формируются по двум величинам: глубина (в метрах) и время (точное время формирования данных). Формирование информации в функции времени открывает новые возможности для определения и оценки коллекторов в газовом каротаже. Это результаты диффузионного газового каротажа (ГКПБ), привязка газовой аномалии к определенной глубине после наращиваний, промывки, перерывов в циркуляции, анализ поглощений промывочной жидкости и проявлений флюидов, который невозможно провести без «временных» замеров. Помимо вышперечисленного, данные, сформированные по времени, производят контроль сформированного архива по глубине.

Задачи, которые должна решать современная станция ГТИ, весьма обширны. Они прописаны в Национальном стандарте Российской Федерации ГОСТ Р 53375–2009.:

- обеспечение безопасности проведения работ при строительстве скважины;
- обеспечение соблюдения правил недропользования;
- геологические задачи;
- технологические задачи;
- расширение комплекса ГТИ за счет новых технологий;
- технико-экономические задачи;
- научно-исследовательские задачи;
- информационные задачи.

В СССР процесс контроля за бурением развивался независимо и параллельно с зарубежными странами. Первой станцией была станция ГТИ «Разрез», изготовленная в 1988 году. В 1995 году была изготовлена первая компьютеризированная станция «Разрез-2». В период с 1996 года было изготовлено более 240 таких станций. «Разрез-2» производит автоматизированный сбор, обработку и интерпретацию полученной информации в процессе разработки скважин. Оборудование для сбора параметров бурения состоит из датчиков, блоков питания и устройств, предназначенных для обработки сигналов. Аппаратура непосредственно станции делится на два вида: стационарная (Рис. 9) и выносная (рис. 10). Стационарное оборудование находится в кузове станции, выносная располагается на буровой.

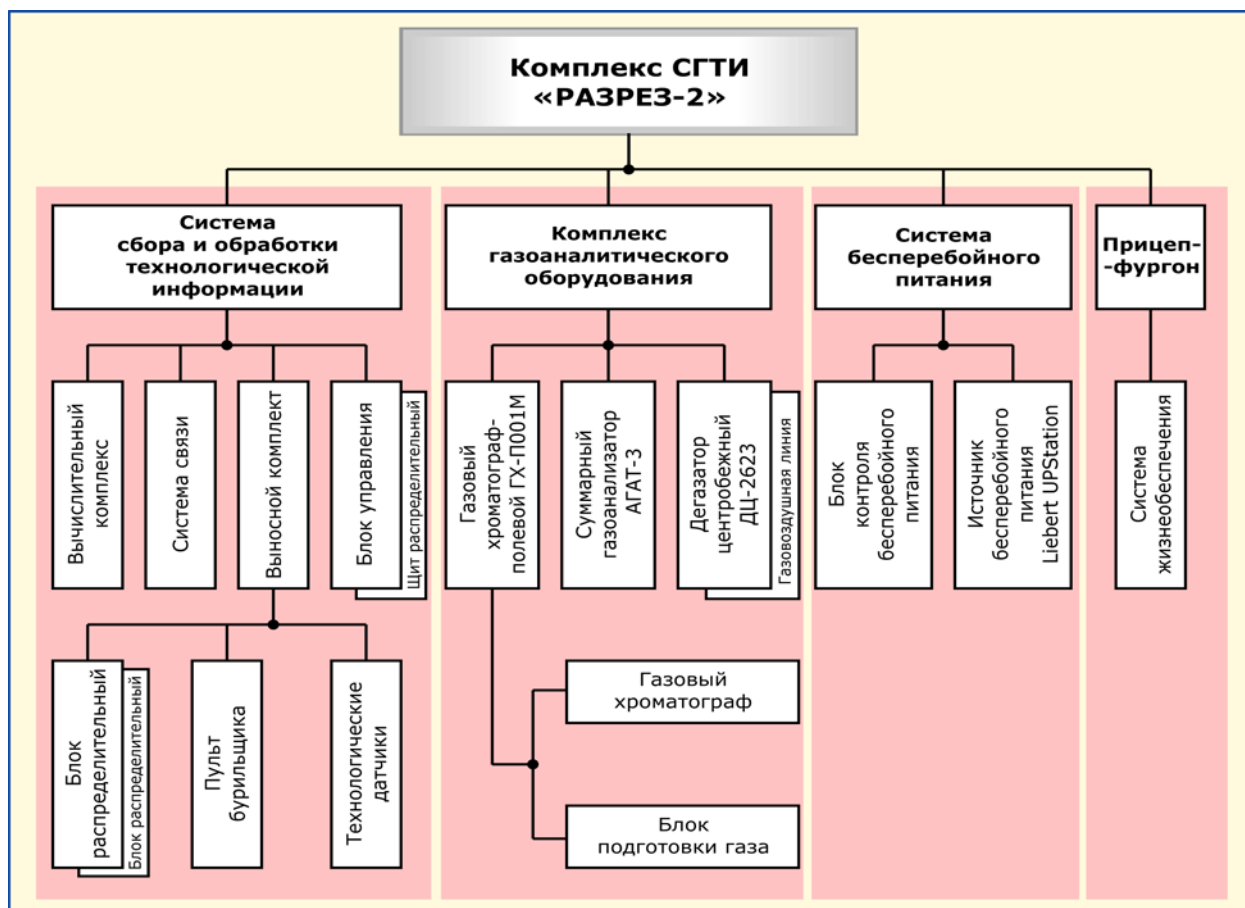


Рис. 9 - Стационарная станция геолого-технологических исследований СГТИ «РАЗРЕЗ-2»

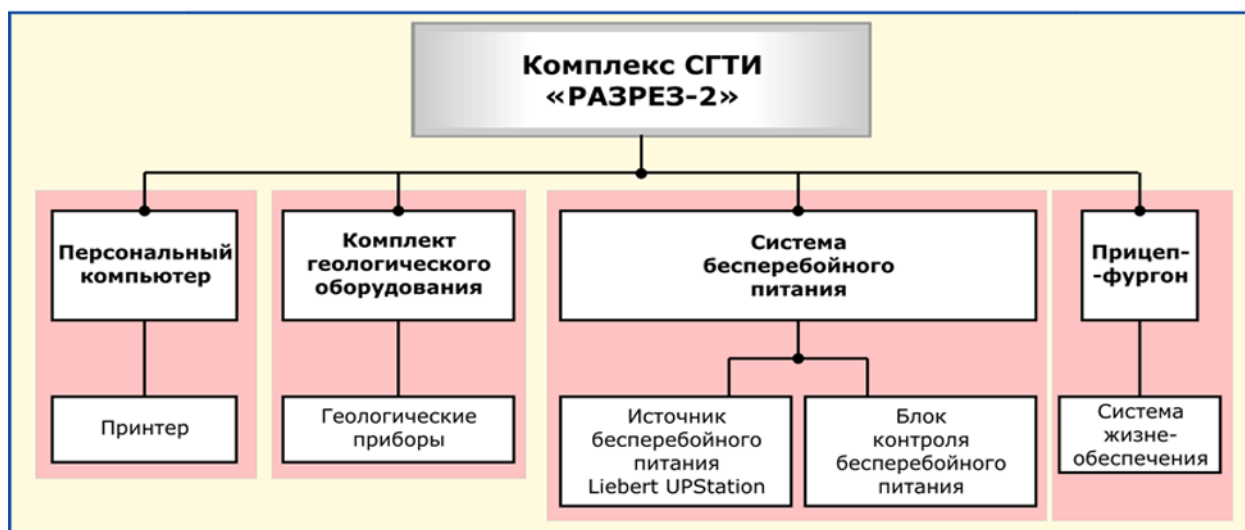


Рис. 10 - Выносная станция ГТИ «Разрез-2». Вариант использования – геологическая кабина.

Данная станция ГТИ позволяет решить следующие задачи:

Геологические:

- литологическое расчленение разреза;
- выделение пластов-коллекторов;
- определение характера насыщения пластов-коллекторов;
- оценка фильтрационно-емкостных свойств пластов-коллекторов;
- выявление реперных горизонтов.

Технологические:

- раннее обнаружение газонефтеводопроявлений и поглощений при бурении;
- оптимизация процесса углубления скважины;
- выбор и поддержание рационального режима бурения с контролем отработки долот;
- контроль гидродинамических давлений в скважине;
- контроль пластовых и поровых давлений, прогнозирование зон АВПД и АВПоД;
- диагностика предаварийных ситуаций в реальном масштабе времени.

Информационные задачи:

- передача по требованию заказчика получаемой информации по каналам связи, сбор, обработка и накопление геолого-технологической информации в базе данных для ее дальнейшего использования.

Однако, «Разрез-2» является устаревающей станцией, не адаптированной к высоким требованиям в связи с быстроразвивающимися информационными технологиями. Более информационно развитой станцией является СГТИ «Геотест-5».

Станция «Геотест-5» состоит из технологического модуля, в котором находится расширенный набор датчиков, которые входят в состав станции «Леуза-2», а также геологического модуля и блока газового каротажа, которые, в свою очередь, входят в состав «Геогаз-1». Схема станции приведена на рисунке 3.

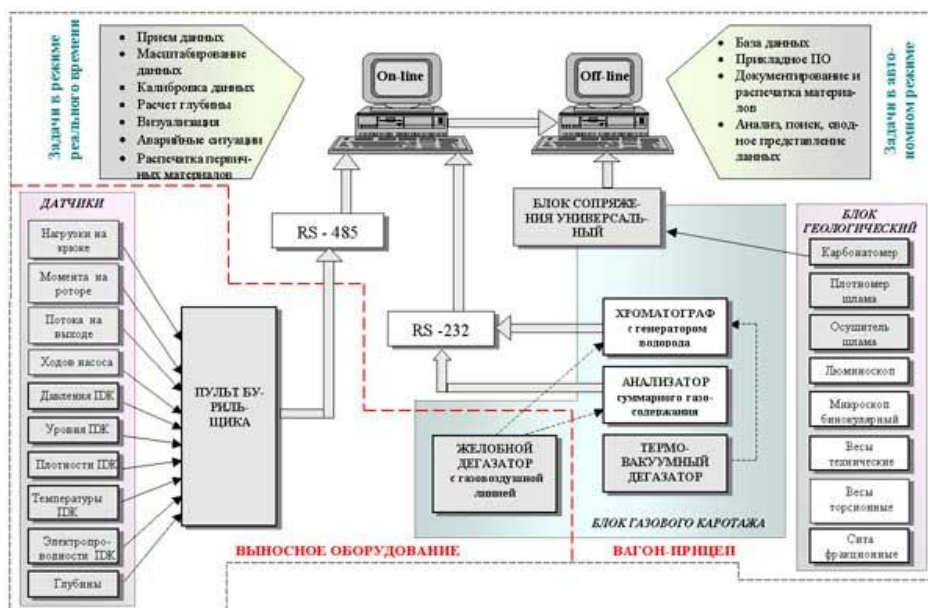


Рис. 11 - Функциональная схема СГТИ «Геотест-5».

Станция находится в вагон-прицепе, который раздел на три отсека: аппаратный, геологический и бытовой. В аппаратном отсеке находятся два ПК, один из них регистрирует данные с буровой в режиме реального времени, второй обрабатывает данные в автономном режиме. Также в аппаратном отсеке находится блок газового каротажа.

Пульт бурильщика основной процессор. Аналоговый сигнал поступает от датчика давления, здесь декодируется и оцифровывается, а затем отсылается в Рабочую станцию. Любые изменения в программировании во время бурения являются критичны для пульта бурильщика. Ноутбук предоставляет оператору средства мониторинга. Это также жизненно важно для программирования Пульта бурильщика на буровой площадке и передатчика (MWD электроника). Рабочая станция служит для отображения и сохранения данных и не играет никакой роли в расшифровке данных. Радио модем используется для передачи данных с пульта бурильщика на рабочую станцию. Вся информация и данные, передаваемые между двух компонентов, зашифрованы в целях безопасности.

Данные с любых ГТИ отправляются в Центр Обработки Информации. Отправка происходит одним пакетом примерно 6 раз в сутки. Такая малая скорость позволяет отслеживать информацию уже постфактум, когда данные уже изменились. Такая задержка данных не позволяет говорить об оперативном централизованном контроле, и уж тем более об управлении.

Помимо этого, данные с датчиков зачастую формируются в стандартах LAS и WITS, которые не могут позволить реализовать режим реального времени, и, как следствие, невозможно смоделировать объемное месторождение.

### **2.3 Цель создания «агрегатора». Анализ уже существующих решений.**

С каждым годом требования к аппаратно-программному комплексу в отрасли нефтедобычи все больше повышаются. Возникает необходимость разрабатывать более сложные месторождения, где требуется большая точность и осторожность. Ошибки, аварии при бурении могут повлечь за собой потери, исчисляемые в огромных суммах. Все это способствует поиску решений, увеличивающих контроль процесса бурения. Идеальная схема – данные передаются с множества датчиков сразу же прямо в центр моделирования месторождений, где создается объемная модель пластов, прогнозируются возможные аварийные ситуации, проблемы, происходит мгновенная корректировка процесса.

На практике все выглядит иначе. Данные не передаются в режиме real-time, месторождение моделируется в «плоском» варианте. Для того, чтобы заменить всю систему получения, обработки и передачи данных потребуются огромные суммы, не каждая компания имеет такую возможность. Переход на новую схему должен иметь место, однако этот переход должен быть постепенным.

Именно для этой цели создается «агрегатор» – аппаратно-программный решение, позволяющее принимать данные с ГТИ в стандартах LAS и WITS в режиме реального времени, конвертировать их в новый формат WITSML 1.3.1 или 1.4.1, сохраняя при этом в базе данных старые отчеты на случай неполадок при конвертации или передачи данных на WITSML сервер. При этом новая схема передачи данных будет работать параллельно со старой, что позволит, не нарушая регламент, внедрить новые технологии, протестировать их на предмет ошибок, не вызывая сбоев в отлаженной, хотя уже и устаревшей, системе.

В России пока нет аналогов данному устройству. Лишь за рубежом существует ряд решений. Например - компания PetroDAQ разработала продукт PetroDAS, который поддерживает множество типов датчиков, различных протоколов. «Агрегатор» PetroDAS реализует режим реального времени, буферизацию данных, а также конвертирует отчеты в формат WITSML (рисунок 1).

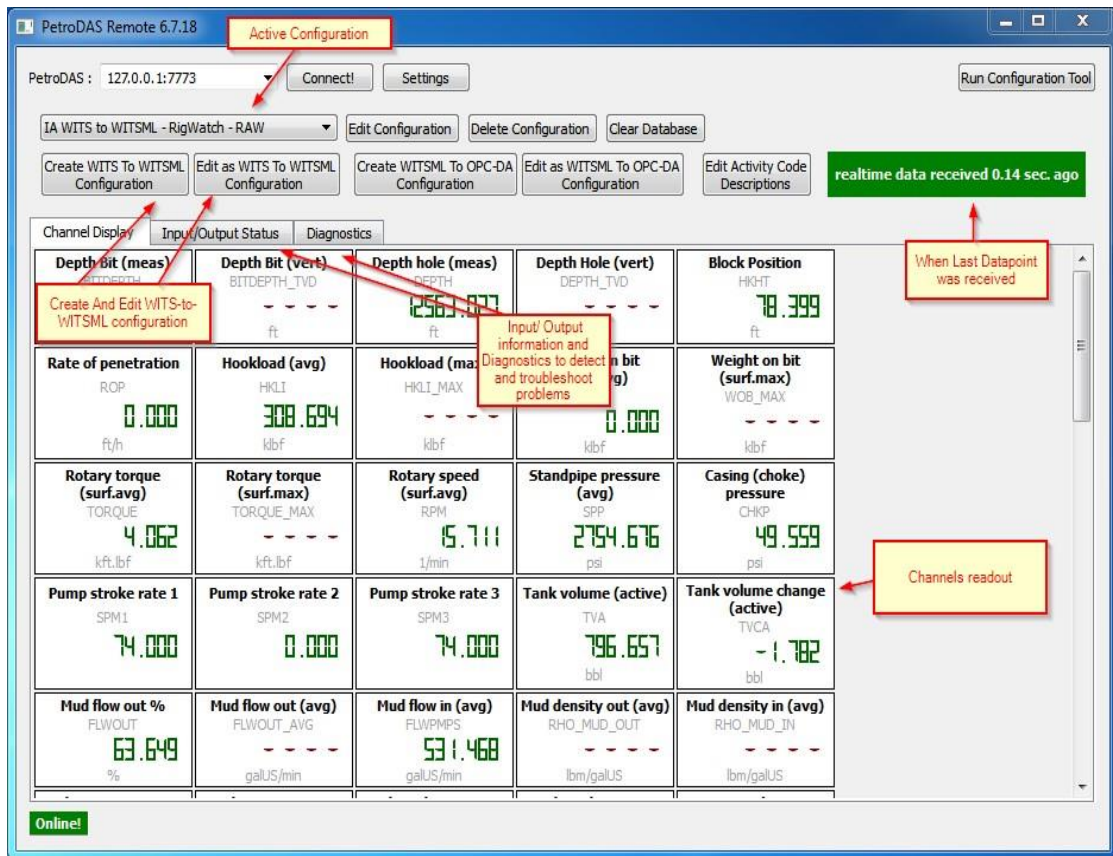


Рис. 12 - Конвертация стандарта WITS в WITSML.

### Функции PetroDAS:

- Входы из одновременных источников данных;
- Поддержка большого количества форматов данных и протоколов, используемых при бурении нефтяных скважин;
- Конвертация стандартов;
- Обработка быстрых потоков данных;
- Буферизация исходных отчетов;
- Мгновенные вычисления;
- Одновременные выходы в несколько систем назначения, действующие в качестве реле;
- Обмен данными между несколькими системами;
- Калибровка сигнала датчика;
- Повторная калибровка цифрового канала.

PetroDAS работает как системная служба в фоновом режиме и не имеет графического интерфейса пользователя. Конфигурирование и визуализация информации, полученной PetroDAS, выполняются клиентскими приложениями.

PetroDAS поддерживает стандарт Serial, TCP / IP и цифровой ввод / вывод на дисках.

Таблица 1. Транспортные протоколы, поддерживаемые PetroDAS

| Протокол                     | Вход | Выход |
|------------------------------|------|-------|
| Последовательный порт RS-232 | •    | •     |
| Последовательный порт RS-485 | •    | •     |
| Последовательный порт RS-422 | •    | •     |
| Клиент TCP / IP              | •    | •     |
| Сервер TCP / IP              | •    | •     |
| Файл диска                   | •    | •     |
| Беспроводная связь - 900 МГц | •    | •     |
| Беспроводная связь - 2,4 ГГц | •    | •     |

Кроме того, модули PetroDAQ реализуют и другие виды передачи данных, такие как оптоволоконная передача данных, а также сигналы с датчиков и цифровые данные, передающиеся по беспроводной сети свыше 900 МГц.

#### 2.4. Описание процесса передачи данных с использованием «агрегатора», структура «агрегатора».

При внедрении «агрегатора» процесс передачи данных от датчиков к центру моделирования месторождения распараллелится (рисунок 3).



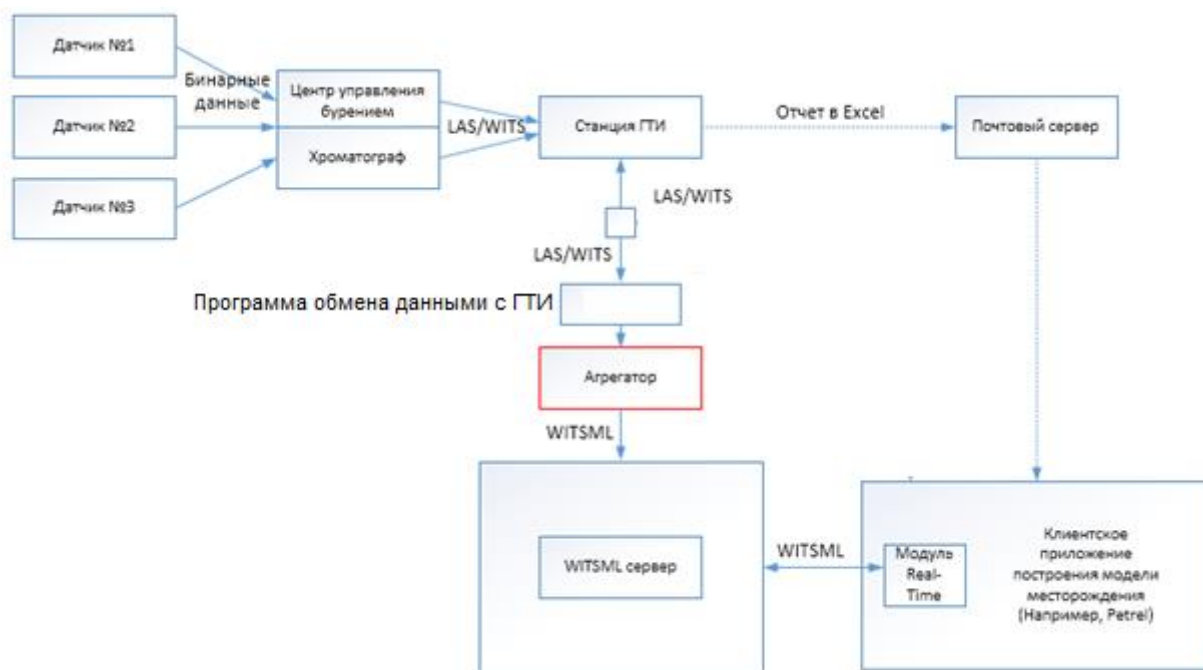


Рис. 13 - Схема передачи данных с «агрегатором».

Процесс передачи данных выглядит следующим образом:

1. Данные с датчиков передаются в центр управления бурением;
2. С центра управления бурения данные отправляются на станцию ГТИ в виде двоичного кода, в станции ГТИ преобразуются в форматы LAS или WITS;

Далее происходит распараллеливание процесса. По старой схеме:

3. Пакет отчетов через 3-4 часа отправляется со станции ГТИ в центр моделирования месторождений через почтового клиента (классическая схема, применяемая в большинстве случаев в отечественной технологии сбора данных).

И одновременно с этим происходит новый процесс передачи:

4. Станция ГТИ делает запрос на установление связи и проверяет готовность АПК (аппаратно-программный комплекс) принять данные.
5. Модуль управления и аутентификации проверяет регистрацию скважины, формирует приёмный буфер для приёма сообщения и отправляет готовность принять данные в стандарте LAS или WITS.
6. После приёма сообщения проверка на правильность приёма данных по сравнению контрольных сумм. Если сообщение правильное – то запись в БД и одновременно в блок конвертации.

7. После конвертации, сообщение в стандарте WITSML одновременно с помощью API посылается на сервер WITSML.

8. С сервера WITSML данные отправляются в центр моделирования месторождения.

Так как отчеты передаются не 6 раз в сутки, а раз в 3-4 минуты модель будет строиться точнее и эффективнее.

На рисунке 4 представлена схема передачи данных с датчиков в агрегатор и центр моделирования месторождений.

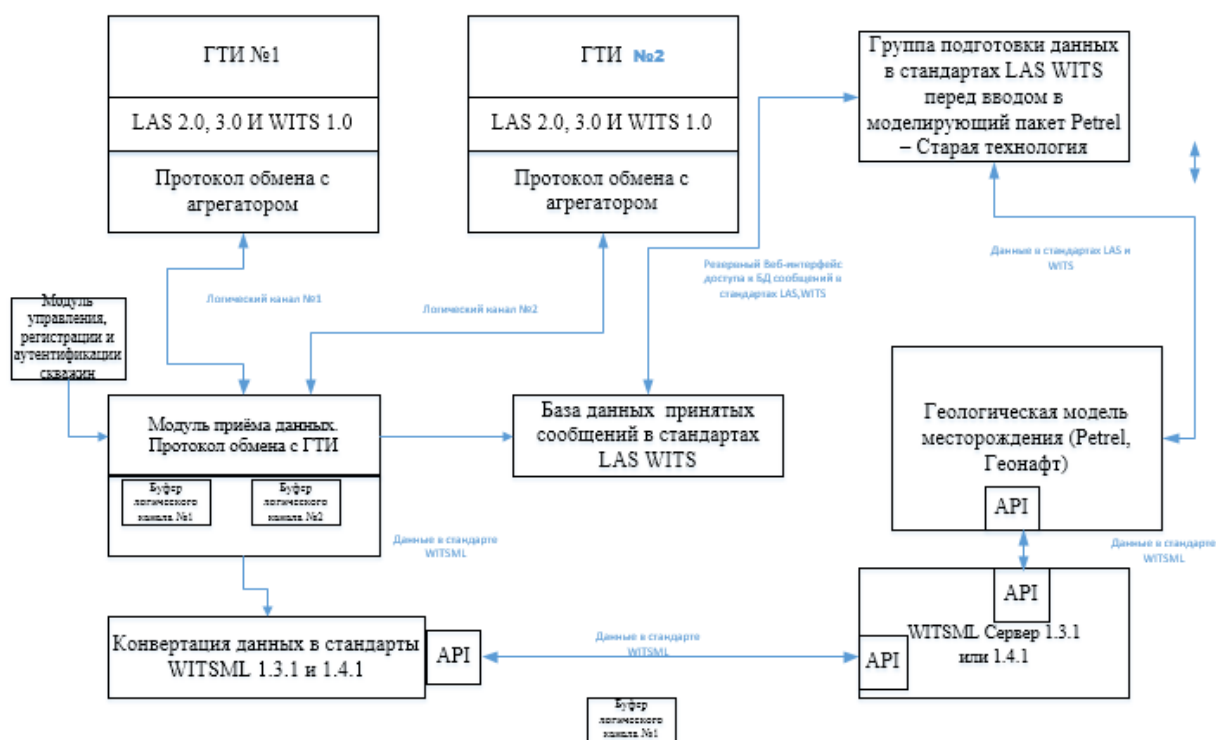


Рис. 14 - Архитектура системы сбора данных и передача их в модель месторождения с применением «Агрегатора».

Разрабатываемый агрегатор состоит из нескольких частей:

- Модуль управления регистрацией и аутентификацией скважин. Он необходим для того, чтобы принимать данные только от идентифицированной скважины.
- Модуль приема данных. Данный модуль позволяет взаимодействовать со станцией ГТИ – принимать и передавать сообщения.

- База данных принятых сообщений в формате LAS и WITS. В этой базе данных хранятся исходные файлы. В случае ошибки конвертации файлы можно будет снова взять в базе и сконвертировать заново.
- Блок конвертации данных в стандарт WITSML 1.3.1 и 1.4.1. Блок преобразует стандарты LAS 2,3 и WITS 1 в стандарт WITSML 1.3.1 и 1.4.1.

## **2.5 Преимущества использования стандартов WITSML 1.3.1 и WITSML 1.4.1 при реализации передачи данных в режиме реального времени.**

WITS является промышленным стандартом с середины 1980-х годов, который использует двоичный формат для передачи данных бурения на буровых и на сегодняшний день считается устаревшим для использования стандартом. Поэтому на смену ему в 2000 г. было положено начало развитию стандартов WITSML в целях передачи данных в последовательной форме и их интеграции с разных буровых установок в формате XML. Первопроходцами в этом начинании стали компании BP и Statoil. Компания Energinor предоставляет независимую поддержку стандартов WITSML с начала 2003 г., когда была сформирована специализированная группа - WITSML Special Interest Group (SIG). Плановое развитие позволило включить в группу представителей из всех основных и многих других нефтяных и сервисных компаний.

WITSML базируется на веб-технологии и построен на XML, который является одновременно платформой и независимым языком.

Большинство зарубежных нефтяных компаний или уже внедрило технологию WITSML для передачи данных в режиме реального времени с буровых и из их архивов, или планируют сделать это в ближайшее время. Объем программного обеспечения, совместимого с WITSML, растет в ответ на эти требования, а сам стандарт в большей мере становится уже договорным обязательством, особенно это относится к новым месторождениям. На данный момент ряд причин обуславливают развертывание WITSML. Стандарт представляет собой интеллектуальный инструментарий для интеграции данных и лежит в основе технологии буровых центров. На данный момент последняя версия WITSML 1.4.1 поддерживает оптимизацию автоматического

использования сценариев для минимизации вмешательства пользователя в обработку потоков данных. В нее добавлены новые объекты данных для более гибкого описания системы координатных ссылок (Coordinate Reference Systems), в то время как оператор (в отличие от вендора - компания, которая разрабатывает и распространяет программное обеспечение) сосредоточен на ежедневном отчете по бурению. В процессе подготовки версии было обработано более сотни запросов по оптимизации.

Стандарт WITSML представляет важный вклад в управление данными по добыче и позволяет осуществлять обмен данными с программным обеспечением разного рода. С его помощью осуществляется информационный обмен в режиме реального времени между платформой и головным офисом, и автоматически происходит передача данных в приложения по геологии и геофизике. Таким образом, значительно улучшается поддержка и процесс принятия решений. Данные становятся доступными всем последующим бригадам, работающим на скважине, и результатом такого подхода передачи информации является более высокий конечный результат.

Сервисным компаниям, выполняющим обработку и интерпретацию данных, принятие стандарта WITSML значительно упрощает интеграцию данных, получаемых от разных поставщиков. Таким образом, можно увеличить количество обслуживаемых конечных точек, которым предоставляются услуги по интерпретации и оптимизации данных, что позволяет более эффективно использовать задействованные процессы и приложения. Стандарт WITSML также предоставляет возможность выхода на рынок небольшим сервисным компаниям. Они могут предоставлять услуги по интеграции данных с инфраструктурой WITSML добывающей компании. В итоге получаем инновации в нефтегазовой отрасли.

С точки зрения разработки программного обеспечения, четкие стандарты получения данных с буровой упрощают процесс сбора информации при разработке новых решений, что может значительно сократить временной цикл разработки с момента концептуального видения до поставки. Используя объекты

данных стандарта WITSML и доступ к данным через API, разработчики могут создавать приложения и интегрировать их со сторонними приложениями, или собирать большое количество наборов данных от других служб и вендоров в новые приложения для анализа, визуализации и совместной работы. Стандарт WITSML находит все более широкое применение, что позволяет оптимизировать рабочие процессы, как в компаниях, так и между ними, чем обуславливается необходимость создания разработчиками дополнительного программного обеспечения, независимого от вендора.

## **2.6 Основная идея использования блока приема и хранения данных в стандартах LAS и WITS.**

Для решения проблемы обработки данных в режиме реального времени в томском политехническом университете была разработана и частично реализована концепция информационной системы «агрегатор», которая позволяет приблизить время обработки данных к режиму реального времени. «Агрегатор» представляет собой компьютер с установленной на него операционной системой семейства windows, а также уникальным прикладным программным обеспечением, которое выполняет основные функции «агрегатора», а именно: принятие данных, их хранение и конвертация из стандартов LAS и WITS в стандарты WITSML 1.3.1 и WITSML 1.4.1.

Схема, по которой предлагается внедрение «агрегатора» в систему передачи данных не подразумевает кардинальных изменений этой системы, так как «агрегатор» представляет собой вычислительную машину с установленным уникальным программным обеспечением и не зависит от месторасположения. Тем не менее информационная система «агрегатора» достаточно обширна и включает в себя несколько ключевых блоков, каждый из которых отвечает за выполнение определенной функции. Подробная структурная схема работы «агрегатора» в процессе передачи данных со станций ГТИ представлена на рисунке 4.

Основным отличием от типичной схемы передачи данных является не только наличие «агрегатора», несущего в себе функции принятия, проверки,

хранения и конвертации данных, но также и специального WITSML сервера, который делает процесс обработки данных с помощью любого из программных комплексов более эффективным, так как позволяет хранить множество значений в уже стандартизированном формате WITSML.

Сам же «агрегатор» представляет собой набор последовательных программных блоков, каждый из которых отвечает за определенные действия (рисунок 2).

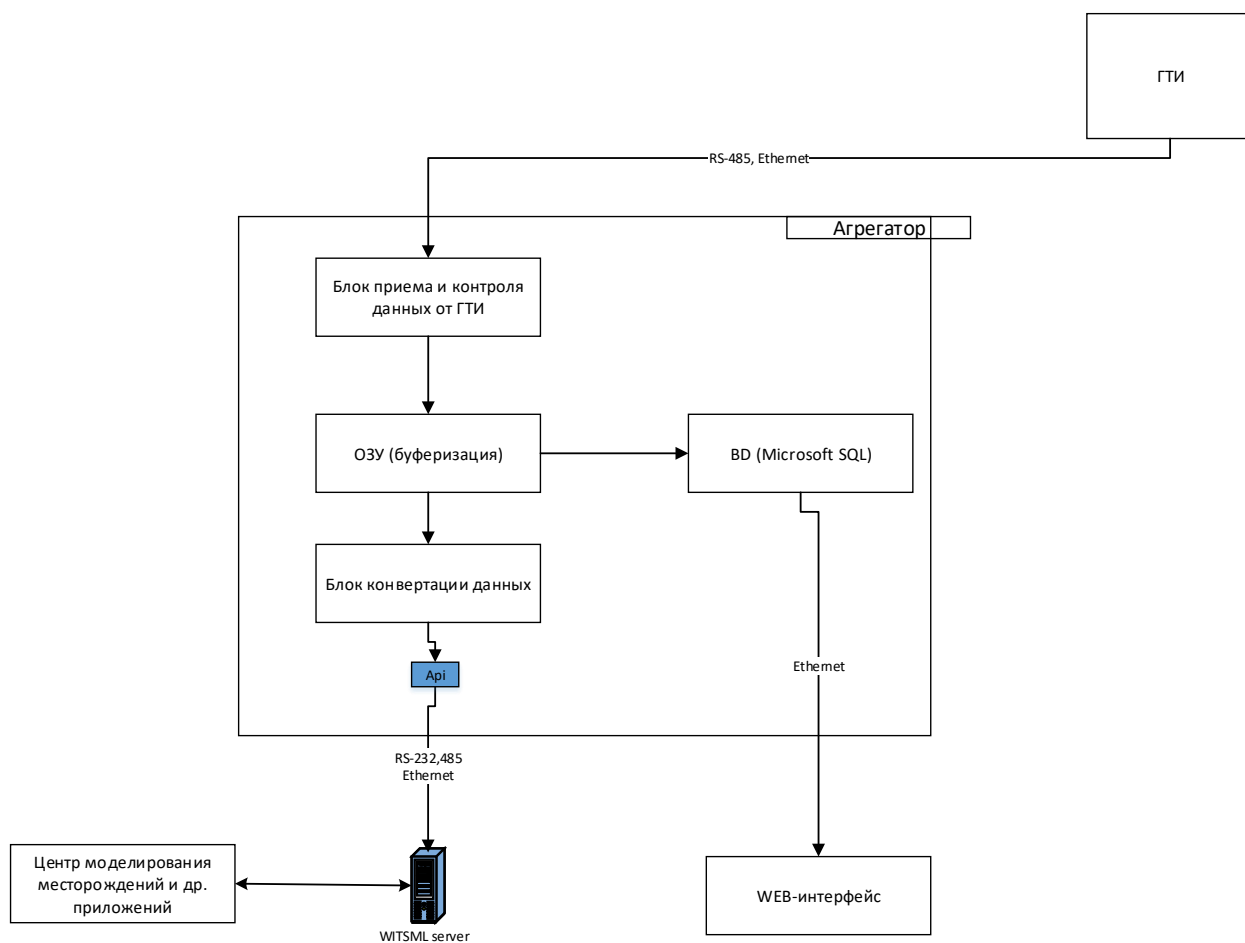


Рис. 15 - Блок схема "агрегатора"

Между станцией ГТИ и «агрегатором» устанавливается соединение по логическому каналу. Модуль управления, регистрации и аутентификации скважин проверяет соединение со станцией ГТИ. Далее (блок приёма и управления анализирует запрос- зарегистрирован ли данный номер скважины, её координаты для того, чтобы соотнести поступающие данные с определенной станцией ГТИ,

так как их может быть несколько. После того как установлено соединение и станция ГТИ прошла аутентификацию создаются виртуальные каналы (рисунок 3), по которым ведется прием данных в буфер. Данные поступают в формате LAS и WITS, при этом с каждым файлом отправляется его контрольная сумма. Затем блок, отвечающий за правильность поступающих данных высчитывает контрольную сумму, используя алгоритм CRC-32 и сверяет рассчитанное значение с тем, которое было получено в ходе передачи. В случае если значения не совпадают следует запрос на ГТИ с просьбой повторить передачу файла, или подтверждение о приёме файла, если значения файлов абсолютно совпадают, то файлы отправляются на конвертацию и записываются на хранение в базу данных, которая в свою очередь для более удобного доступа пользователя имеет веб интерфейс. Важной функцией базы данных является возможность осуществлять выборку этих данных по определенным критериям. Такой функционал определяет возможность хранения файлов, в случаях ошибок в процессе конвертации, нарушения логических или физических каналов связи и других внештатных ситуациях, которые могут произойти во время приема, обработки и последующей отправки информации. Вместе с этим данные проходят конвертацию по стандарту WITSML 1.3.1 или WITSML 1.4.1 и с помощью специально написанного API отправляются на WITSML сервер до востребования программным комплексом, который будет строить по полученным данным геологическую модель месторождения для последующего анализа.

## 2.6.1 Программа и методы обмена данными между станциями ГТИ и «агрегаторами».

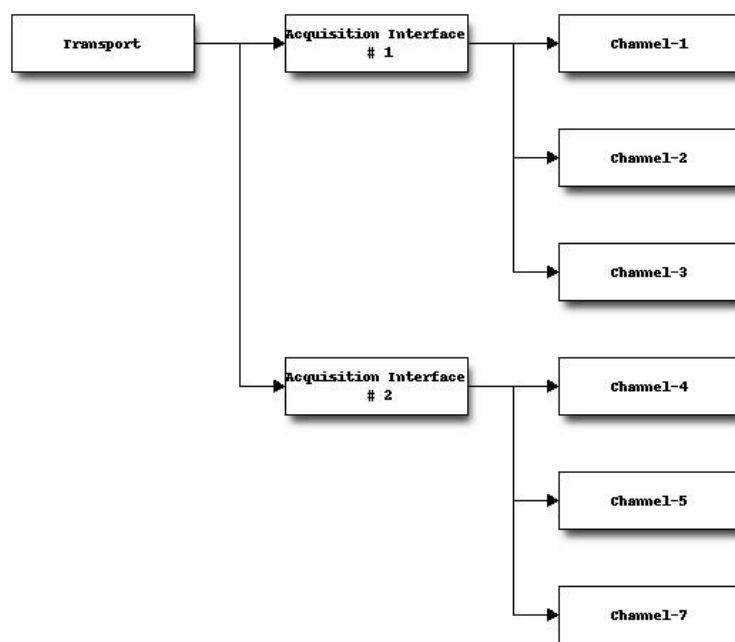


Рис. 16 - Создание виртуальных каналов на примере продукта PetroDAS, разработанного компанией PetroDAQ

«Агрегатор» использует концепции каналов, интерфейсов сбора данных и транспортов.

Каналы являются дискретными единицами информации, которые представляют непрерывный поток данных. Примером канала является входной сигнал датчика или столбец данных файла базы данных.

Transport - это физические и логические уровни, используемые для передачи данных внешним программам. Серийные соединения, TCP-клиенты и TCP-серверы являются примерами обычных транспортов.

Acquisition Interface (интерфейс сбора данных) является обработчиком данных в PetroDAS, который использует Transport для получения информации, анализа информации и создания виртуальных каналов. С Transport может взаимодействовать не один Acquisition Interface.

«Агрегатором» принимаются только файлы LAS WITS, данные с датчиков обрабатываются ГТИ и переводятся в LAS и WITS.



Все функции, предоставляемые Python, доступны на уровне описания сценариев PetroDAS, включая библиотеки и расширение.

Каналы доступны на уровне Scripting. В коде Scripting следующие логические функции управляют следующими функциями.

Выполнение скрипта выполняется следующими функциями:

```
def on_initialize( context ):  
    #  
    # Этот блок кода выполняется один раз при инициализации  
    # Времени. Это полезно для установки начального значения  
    # Глобальных переменных  
    # def on_acquire( context ,  
acquisition_interface ):  
    #  
    # Этот блок кода выполняется каждый раз, когда новые  
#данные  
  
    # принимаются для любого из каналов, в соответствии с  
  
    # установленным интервалом сбора данных.  
    def on_finalize( context  
    ):  
        #  
        # Этот блок кода выполняется в конце  
        # Исполнение (остановка программы). Это полезно для очистки  
        # Задач, таких как: очистка вывода и закрытие файлов
```

При запуске выполняется процедура `on_initialize (context)` и получает ссылку на класс `CScriptContext`.

Сразу после этого создается экземпляр объекта `varRepo`. Этот объект используется для восстановления всех сохраненных переменных.

Когда любые данные принимаются по какому-либо каналу выполняется функция `on_acquire (контекст, получение_интерфейс)`. Эта функция передается ссылкой на классы `CScriptContext` и `CInterfaceToAcquisitionInterface`.

Для целей доступа к истории данных и обеспечения механизма восстановления после перерывов связи в системах, «конвертер» реализует буферизацию данных. Эти буферы данных оптимизированы для быстрой скорости передачи данных.

Буферизация обеспечивается в двух режимах: режиме реального времени.

В режиме реального времени данные хранятся в базе данных и позже считываются выходными потоками. Скорость выходных данных управляется настройками в выходном потоке. Этот режим используется для асинхронной доставки данных, также известной как режим «real time» или почти в режиме

реального времени. Этот режим используется, когда к получателям обращаются по прерывистой транспортировке, например, к спутниковым каналам связи, или, когда время обработки и доступная пропускная способность приводят к задержкам в передаче данных.

В случае прерывания канала передачи данных с приемником данные сохраняются в базе данных. Как только связь восстановлена, данные передаются на конвертацию.

Конвертация происходит по определенному уникальному алгоритму, например, для наиболее часто используемого стандарта описания данных каротажа в России (LAS) сделана программа (в виде простой библиотеки классов), которая конвертирует las-файл в xml-файл, соответствующий смехе WITSML.

Анализ стандарта LAS:

Las-файл состоит из секций. Каждая секция начинается со знака (~). Всего секций 6:

- 1) ~V (обязательно) - информация о версии файла
- 2) ~W (обязательно) - информация о скважине
- 3) ~C (обязательно) - информация о каротажных кривых
- 4) ~P - информация о параметрах и константах
- 5) ~O - прочее и комментарии
- 6) ~A (обязательно) - цифровые данные по скважине

Специальные символы Las-файла:

- ~ : обозначение начала секции
- # : обозначение строки комментария (Строки комментария могут встречаться в любом месте выше секции данных. Использование комментариев внутри секции данных запрещено)

Анализ алгоритма работы программы конвертации

Для более наглядного представления работы программы составлена блок-смеха (рис. 17).

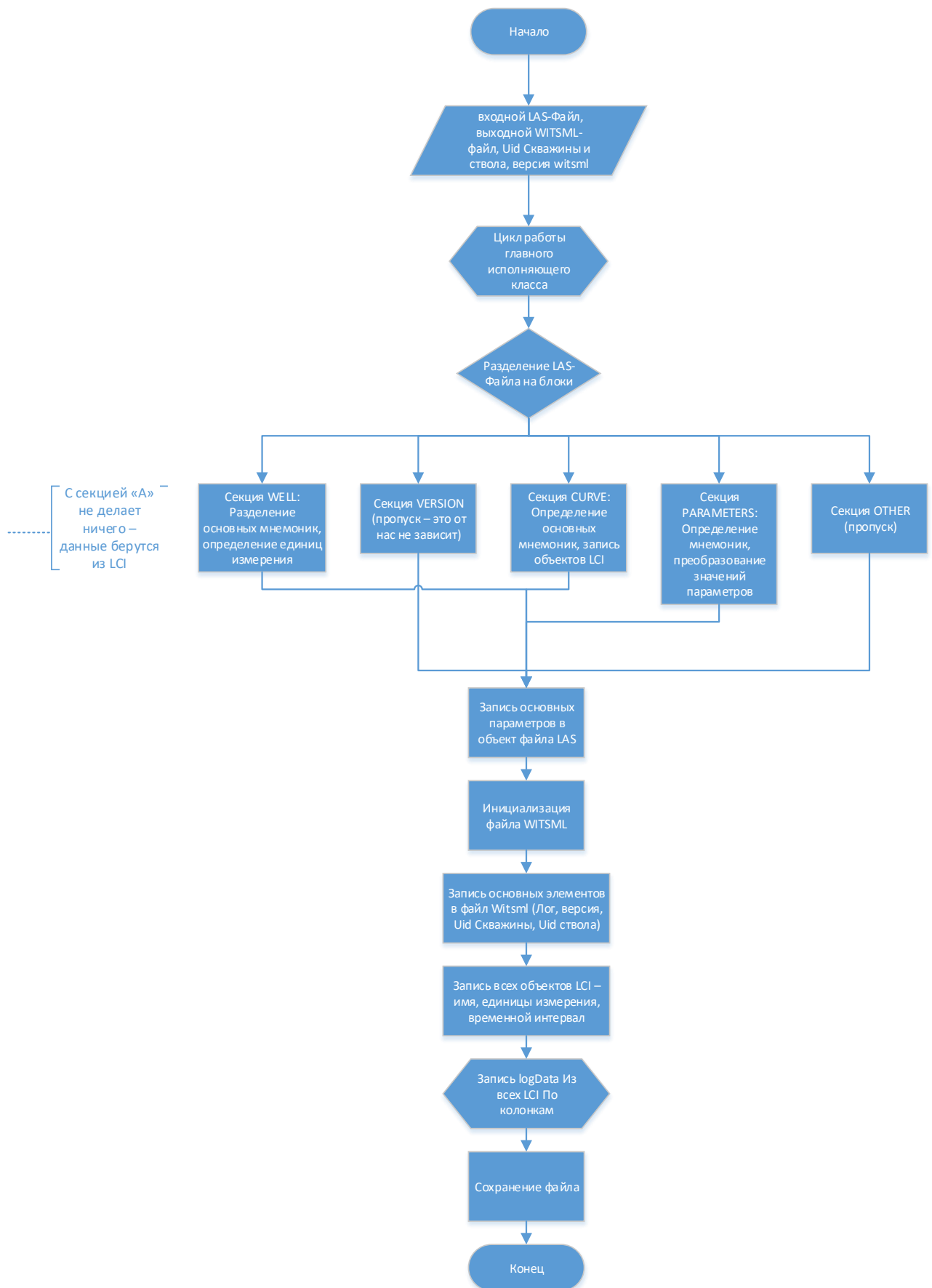


Рис. 17 - Блок-схема алгоритма работы программы конвертации

Краткое описание алгоритма работы программы:

- Программа берет на вход следующие параметры: входной файл (las), выходной файл (xml), uid скважины, uid ствола скважины, uid лога, версию witsml (1.3.1 или 1.4.1)

Работа с LAS

- Программа разделяет las-файл на секции (w,v,c,p,o,a).
- В секции W: разделяет основные мнемоники на составляющие, определяет единицы измерения, записывает начало и конец записи параметров (STRT, STOP, STEP, NULL, SRVC).
- С секцией V ничего не делает.
- В секции C программа разделяет мнемоники на составляющие, а также делает записи в логи кривых.
- В секции P также разделяет параметры на мнемоники; преобразует значение некоторых мнемоник (в целочисленные или с плавающей точкой).
- С секцией O ничего не делает.

Работа с WITSML

- Инициализация файла заданной версии witsml (по умолчанию 1.4.1).
- Вызов метода конвертации. Данный метод берет выбранный las-файл, uid скважины, uid ствола, name - для задания имени скважины и ствола.
- Далее происходит предварительная подготовка las-файла: вычленение и разделение информации о логах глубины и времени.
- Далее идет определение версии witsml.
- Далее идет запись xml объектов : логов, информации о скважине, компании.
- Далее следует цикл по добавлению строк во временный файл.
- Когда подготовлены все строки и logCurveInfo в цикле LCI записываются в файл
- После этого в файл записываются построчно logData.
- Файл witsml сохраняется.

Пример пользовательского интерфейса для конвертера представлен на рисунке 18.

Для того, чтобы пользоваться библиотекой необходим интерфейс. В windows forms сделан простой интерфейс:

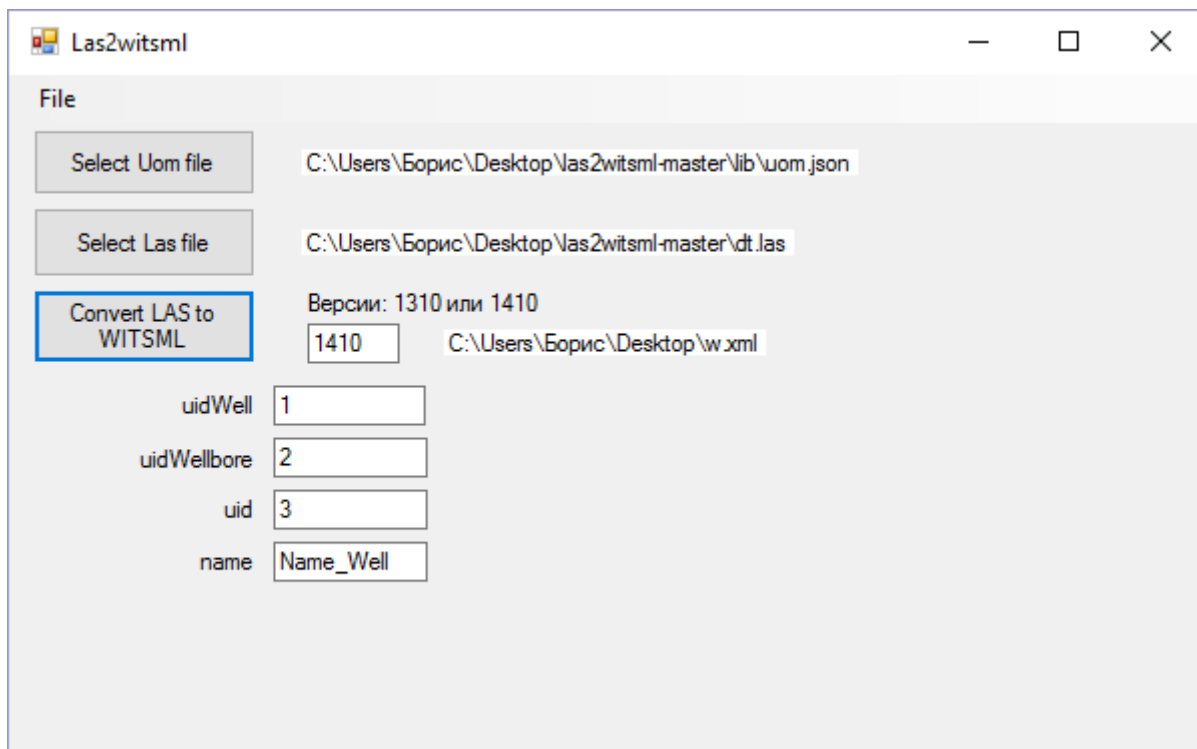


Рис. 18 - Простой интерфейс взаимодействия с конвертером

Пошаговая работа с интерфейсом:

Выбор файла Uom (необходим для корректной конвертации единиц измерения из las в witsml);

Выбор файла las;

Заполнение параметров для witsml файла (версия, id скважины, имя скважины);

Выбор места сохранения witsml файла.

Начальный файл в формате LAS представлен на рисунке 19. Результаты выполнения программы представлен на рисунке 20.

```

1 ~V
2 VERS . 2.0 :
3 WRAP . NO :
4 ~W
5 STRT .sec 0.0000 : 10/26/08 21:30:00.00 (Central Daylight Time)
6 STOP .sec 3015000.0000 : 11/30/08 18:00:00.00 (Central Standard Time)
7 STEP .sec 10.0000 :
8 NULL . -999.0000 :
9 COMP . Oxy :
10 WELL . University 20-26 #1H :
11 FLD . Bone Spring :
12 LOC . Wink :
13 SRVC . Scientific Drilling :
14 DATE . 01/05/2009 :
15 CNTY . Winkler County :
16 STAT . Texas :
17 CTRY . United States :
18 API . 42-495-33311 :
19 ~C
20 Time .sec : Time
21 Weight-on-Bit(surf,avg).kip :
22 TopDrvTorque.kip-ft :
23 TopDrvSpeed.rpm :
24 Temperature.degF :
25 StandpipePressure.psi :
26 ROP(MWD).ft/hr :
27 MudFlowIn.gal/min :
28 MergedcurvefromECD,ECDoLogAnnulusPress,ECDoLogAnnulusPress(MWD).lb/gal :
29 LogRMSVibrations.G :
30 LogPipePress.psi :
31 LogGammaRay.aapi :
32 LogAnnulusPress.psi :
33 HookLoad(MWD).lbf :
34 Gas(avg).% :
35 BlockPosition(MWD).ft :
36 BitDepth(MWD).ft :
37 ~P
38 ~O
39 ~A
40 0.0000 -999.0000 -999.0000 -999.0000 114.0375 59.1979 -999.0000 -999.0000 -999.0000 5.2121 3035.3391
41 10.0000 -999.0000 -999.0000 -999.0000 114.0356 59.1979 -999.0000 -999.0000 -999.0000 5.2171 3035.3398
42 20.0000 -999.0000 -999.0000 -999.0000 114.0338 59.2095 -999.0000 -999.0000 -999.0000 5.2221 3031.3288
43 30.0000 -999.0000 -999.0000 -999.0000 114.0319 59.2679 -999.0000 -999.0000 -999.0000 5.2272 3028.9770
44 40.0000 -999.0000 -999.0000 -999.0000 114.0301 59.9730 -999.0000 -999.0000 -999.0000 5.2322 3035.3361
45 50.0000 -999.0000 -999.0000 -999.0000 114.0283 59.5009 -999.0000 -999.0000 -999.0000 5.2372 3040.4018
46 60.0000 -999.0000 -999.0000 -999.0000 114.0264 61.1389 -999.0000 -999.0000 -999.0000 5.2422 3038.6768
47 70.0000 -999.0000 -999.0000 -999.0000 114.0246 64.4570 -999.0000 -999.0000 -999.0000 5.2472 3038.5400
48 80.0000 -999.0000 -999.0000 -999.0000 114.0227 57.1668 -999.0000 -999.0000 -999.0000 5.2522 3046.7419
49 90.0000 -999.0000 -999.0000 -999.0000 114.0209 60.4357 -999.0000 -999.0000 -999.0000 5.2572 3060.5460
50 100.0000 -999.0000 -999.0000 -999.0000 114.0190 90.3270 -999.0000 -999.0000 -999.0000 5.2622 3103.7619

```

Рис. 19 - Начальный las-файл

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <logs xmlns="http://www.witsml.org/schemas/1series" version="1.4.1.0">
3    <log uidWell="1" uidWellbore="2" uid="3">
4      <nameWell>Name_Well</nameWell>
5      <nameWellbore>Name_Well</nameWellbore>
6      <name>Name_Well</name>
7      <serviceCompany>Scientific Drilling</serviceCompany>
8      <description>Created by lab212</description>
9      <indexType>date time</indexType>
10     <startDateTimeIndex>26.10.2008 21:30:00</startDateTimeIndex>
11     <endDateTimeIndex>30.11.2008 18:00:00</endDateTimeIndex>
12     <indexCurve>Time</indexCurve>
13     <nullValue>-999</nullValue>
14     <logCurveInfo uid="Time">
15       <mnemonic>Time</mnemonic>
16       <unit>sec</unit>
17       <minDateTimeIndex>26.10.2008 21:30:00</minDateTimeIndex>
18       <maxDateTimeIndex>30.11.2008 18:00:00</maxDateTimeIndex>
19       <curveDescription>Time</curveDescription>
20       <typeLogData>float</typeLogData>
21     </logCurveInfo>
22     <logCurveInfo uid="Weight-on-Bit(surf,avg)">
23       <mnemonic>Weight-on-Bit(surf,avg)</mnemonic>
24       <unit>kip</unit>
25       <minDateTimeIndex>26.10.2008 21:30:00</minDateTimeIndex>
26       <maxDateTimeIndex>30.11.2008 18:00:00</maxDateTimeIndex>
27       <curveDescription />
28       <typeLogData>float</typeLogData>
29     </logCurveInfo>
30     <logCurveInfo uid="TopDrvTorque">
31       <mnemonic>TopDrvTorque</mnemonic>
32       <unit>kip-ft</unit>
33       <minDateTimeIndex>26.10.2008 21:30:00</minDateTimeIndex>
34       <maxDateTimeIndex>30.11.2008 18:00:00</maxDateTimeIndex>
35       <curveDescription />
36       <typeLogData>float</typeLogData>
37     </logCurveInfo>
38     <logCurveInfo uid="TopDrvSpeed">
39       <mnemonic>TopDrvSpeed</mnemonic>
40       <unit>rpm</unit>
41       <minDateTimeIndex>26.10.2008 21:30:00</minDateTimeIndex>
42       <maxDateTimeIndex>30.11.2008 18:00:00</maxDateTimeIndex>
43       <curveDescription />
44       <typeLogData>float</typeLogData>
45     </logCurveInfo>
46     <logCurveInfo uid="Temperature">
47       <mnemonic>Temperature</mnemonic>
48       <unit>degf</unit>
49       <minDateTimeIndex>26.10.2008 21:30:00</minDateTimeIndex>
50       <maxDateTimeIndex>30.11.2008 18:00:00</maxDateTimeIndex>

```

Рис. 20 - Файл в формате WITSML

## 2.7 Преимущества использования «агрегатора» при приёме данных с мест бурения.

Создание национального интегрированного стандарта передачи данных для нефтяной промышленности России является очень перспективным направлением исследовательской за счет создания отечественного информационного продукта, в частности, в рамках программы импортозамещения. Стандартизация передачи данных при разработки и добычи углеводородов приняла характер стандартов де-факто практически во всех крупнейших нефтедобывающих и сервисных

компаниях мира. Инициатором создания стандартов является консорциум “Energistics” (<http://www.energistics.org>), образованный и финансируемый ведущими нефтедобывающими компаниями мира.

Разработаны три международных стандарта передачи данных:

- 1) WITSML – стандарт описания данных о бурении;
- 2) PRODMML – стандарт описания данных о добычи углеводородов;
- 3) RESQML – стандарт описания данных о состоянии подземного резервуара.

Отечественные технологии добычи углеводородов отличаются от зарубежных и очень во многом. А значит, использовать зарубежные пакеты моделирования добычи не можем в полную функциональную возможность. Математический аппарат и алгоритмы нам не известны, что приводит к полной зависимости от зарубежных производителей.

Стандартизация данных позволит получить следующие экономические эффекты:

- Прозрачность информационной инфраструктуры – данные могут быть получены и обработаны в любой точке информационного пространства предприятия;
- Ускорение формирования структуры «интеллектуального» месторождения – полностью автоматизированная добыча углеводородов с получением информации о всех происходящих технологических процессах в реальном времени.

В связи с этим создание «агрегатора» поможет решить проблему стандартизации данных, не меняя политики получения и обработки данных компании, а также, не требуя полной или частичной замены оборудования на станциях ГТИ. Использование «агрегатора» приближает этап обработки данных в отечественных компаниях к режиму реального времени.



### **Раздел 3. Анализ доступных для использования в блоке хранения СУБД. Обоснование выбора. Преимущества и недостатки. Реализация выбранной технологии.**

#### **3.1 Описание существующих СУБД.**

База данных представляет собой набор информации, которая организована таким образом, что ее можно легко получить, управлять и обновлять.

Данные упорядочиваются по строкам, столбцам и таблицам и индексируются, чтобы упростить поиск релевантной информации. Данные обновляются, расширяются и удаляются по мере добавления новой информации. Базы данных обрабатывают рабочие нагрузки, чтобы создавать и обновлять себя, запрашивать данные, которые они содержат, и запускать, необходимые для работы приложения.

Как правило, менеджер баз данных предоставляет пользователям возможность контролировать доступ на чтение / запись, задавать генерацию отчетов и анализировать их использование. В некоторых базах данных предусмотрена совместимость ACID (атомарность, согласованность, изоляция и долговечность), гарантирующая согласованность данных и завершение транзакций.

Базы данных развивались с момента их создания в 1960-х годах, начиная с иерархических и сетевых баз данных, в 1980-х годах уже были объектно-ориентированными базами данных, а сегодня наиболее популярны базы данных SQL и NoSQL, а также облачными базами данных.

В одном представлении базы данных могут быть классифицированы в соответствии с типом контента: библиографическим, полным текстом, цифровыми и изображениями. При создании базы данных информация иногда классифицируется в соответствии с организационным подходом. Существует множество различных баз данных, начиная от наиболее распространенного

подхода - реляционной базы данных, до распределенной базы данных, облачной базы данных или базы данных NoSQL.

Реляционная база данных, изобретенная Е.Ф. Коддом в IBM в 1970 году, представляет собой табличную базу данных, в которой данные определены таким образом, что их можно реорганизовать и получить доступ несколькими способами.

Реляционные базы данных состоят из набора таблиц с данными, которые вписываются в predetermined категорию. Каждая таблица имеет по меньшей мере одну категорию данных в столбце, и каждая строка имеет определенный экземпляр данных для категорий, определенных в столбцах.

Язык структурированных запросов (SQL) является стандартным языком общения между интерфейсом пользователя и прикладной программы для реляционной базы данных. Реляционные базы данных легко расширяются, и новая категория данных может быть добавлена после создания исходной базы данных без необходимости изменения всех существующих приложений.

Распределенная база данных - это база данных, в которой части базы данных хранятся в нескольких физических местах и в которых обработка распределяется или реплицируется между различными точками сети.

Распределенные базы данных могут быть однородными или гетерогенными. Все физические местоположения в однородной распределенной системе баз данных имеют одинаковое базовое оборудование и работают в тех же операционных системах и приложениях баз данных. Аппаратные средства, операционные системы или приложения баз данных в гетерогенной распределенной базе данных могут быть разными в каждом из местоположений.

Облачная база данных - это база данных, оптимизированная или созданная для виртуализированной среды. Облачные базы данных предоставляют такие преимущества, как возможность оплачивать емкость и пропускную способность для каждого пользователя по отдельности, а также обеспечивают масштабируемость по требованию наряду с высокой доступностью.

Облачная база данных также предоставляет предприятиям возможность поддержки бизнес-приложений в развертывании программного обеспечения как услуги.

Базы данных NoSQL полезны для больших наборов, распределенных данных.

Базы данных NoSQL эффективны для больших проблем с производительностью данных, которые реляционные базы данных не могут решить. Они наиболее эффективны, когда организации должны анализировать большие блоки неструктурированных данных или данных, хранящихся на нескольких виртуальных серверах в облаке.

Элементы, созданные с помощью объектно-ориентированных языков программирования, часто хранятся в реляционных базах данных, но объектно-ориентированные базы данных хорошо подходят для этих элементов.

Объектно-ориентированная база данных организована вокруг объектов, а не действий, данных, логики. Например, мультимедийная запись в реляционной базе данных может быть определяемым объектом данных, в отличие от буквенно-цифрового значения.

Графо-ориентированная база данных или база данных графов - это тип базы данных NoSQL, которая использует теорию графов для хранения, сопоставления и запросов. Базы данных графов - это в основном коллекции узлов и ребер, где каждый узел представляет сущность, а каждое ребро представляет собой соединение между узлами.

Базы данных графов становятся все более популярными для анализа взаимосвязей. Например, компании могут использовать базу данных графов для сбора данных о клиентах из социальных сетей.

Система управления базами данных (СУБД) - это тип программного обеспечения, которое позволяет вам определять, манипулировать, извлекать и управлять данными, хранящимися в базе данных.

Система управления реляционными базами данных (RDBMS) - это тип программного обеспечения для управления базами данных, которое было

разработано в 1970-х годах на основе реляционной модели и по-прежнему является самым популярным способом управления базой данных.

Microsoft SQL Server, Oracle Database, IBM DB2 и MySQL - лучшие продукты РСУБД, доступные для корпоративных пользователей. Технологии СУБД начались в 1960-х годах для поддержки иерархических баз данных и включают в себя систему управления информацией IBM и интегрированную систему управления базами данных СА.

### **3.2 Выбор технологии MS SQL Server. Преимущества использования.**

Microsoft SQL Server - это система управления реляционными базами данных или СУБД, которая поддерживает широкий спектр обработки транзакций, бизнес-аналитики и аналитики в корпоративных ИТ-средах. Это одна из трех передовых технологий баз данных, а также Oracle Database и IBM DB2.

Как и другое программное обеспечение РСУБД, Microsoft SQL Server построен на базе SQL - стандартизованного языка программирования, который администраторы баз данных (DBAs) и другие ИТ-специалисты используют для управления базами данных и запроса содержащихся в них данных. SQL Server привязан к Transact-SQL (T-SQL) - реализации SQL от Microsoft, которая добавляет набор собственных расширений программирования на стандартный язык.

Исходный код SQL Server был разработан в 1980-х годах бывшей Sybase Inc., которая сейчас принадлежит SAP. Специалисты Sybase изначально создали программное обеспечение для работы на Unix-системах и миникомпьютерных платформах. Затем Microsoft и Ashton-Tate Corp., будучи ведущими поставщиками баз данных для ПК, объединились для создания первой версии того, что стало Microsoft SQL Server, предназначенного для операционной системы OS / 2 и выпущенного в 1989 году.

В период с 1995 по 2016 год Microsoft выпустила 10 версий SQL Server. Ранние версии были рассчитаны прежде всего на приложения подразделений и рабочих групп, но Microsoft расширила возможности SQL Server в последующих, превратив его в реляционную СУБД корпоративного класса, которая могла бы

конкурировать с Oracle Database, DB2 и другими конкурирующими платформами для высокопроизводительных баз данных. На протяжении многих лет Microsoft также включала в SQL Server различные инструменты управления данными и анализа данных, а также функциональные возможности для поддержки новых технологий, в том числе веб-технологий, облачных вычислений и мобильных устройств.

Microsoft SQL Server 2016, который стал общедоступным в июне 2016 года, был разработан в рамках «технологической стратегии мобильного первого, облачного», принятой Microsoft двумя годами ранее. Среди прочего, SQL Server 2016 добавил новые функции для настройки производительности, оперативной аналитики в реальном времени и визуализации данных и отчетности на мобильных устройствах, а также поддержки гибридных облаков, которая позволяет администраторам баз данных работать с комбинацией локальных систем и общедоступных облачных сервисов. Для снижения затрат на ИТ. Например, технология SQL Server Stretch Database перемещает редко доступные данные из локальных устройств хранения в облако Microsoft Azure, при необходимости сохраняя данные для запроса.

SQL Server 2016 также увеличил поддержку больших аналитических данных и других продвинутых приложений аналитики с помощью служб SQL Server R, что позволяет СУБД запускать приложения для аналитики, написанные на языке программирования с открытым исходным кодом R, и технологию PolyBase, которая позволяет пользователям SQL Server получать доступ к данным, которые хранятся в кластерах Hadoop или в блоке памяти Azure для анализа. Кроме того, SQL Server 2016 был первой версией СУБД, предназначенной исключительно для 64-разрядных серверов на базе микропроцессоров x64.

Как и другие технологии РСУБД, SQL Server в первую очередь строится вокруг структуры таблицы на основе строк, которая связывает связанные элементы данных в разных таблицах друг с другом, избегая необходимости избыточного хранения данных в нескольких местах в базе данных. Реляционная

модель также обеспечивает ссылочную целостность и другие ограничения целостности для поддержания точности данных. Эти проверки являются частью более широкого соблюдения принципов атомарности, непротиворечивости, изоляции и долговечности, которые в целом известны как свойства ACID и предназначены для обеспечения надежной обработки транзакций базы данных.

Основным компонентом Microsoft SQL Server является SQL Server Database Engine, который контролирует хранение, обработку и безопасность данных. Он включает реляционный механизм, который обрабатывает команды и запросы, а также механизм хранения, который управляет файлами базы данных, таблицами, страницами, индексами, буферами данных и транзакциями. Хранимые процедуры, триггеры, представления и другие объекты базы данных также создаются и выполняются компонентом Database Engine.

Таким образом технология MS SQL Server обладает следующими преимуществами: упрощенная интеграция приложения с другими приложениями, также работающими с MS SQL Server (в случае роста и развития является значимым преимуществом), отсутствие ограничений по объему данных (зависит от выбранного пакета, например, в бесплатной версии, а также версии для разработчиков существует ограничение до 10 Гб на объемы данных, которые может хранить БД), в случае с Oracle необходимо платить за поддержку (он без этого не продается), а в MS SQL Server поддержка предоставляется бесплатно.

### **3.3 Реализация БД в блоке хранения.**

Для хранения данных, поступающих со станций ГТИ в исходных форматах LAS и WITS для последующей возможности их выборки, а также для возобновления процесса конвертации данных в формат WITSML в случаях различных ошибок с последующей передачей информации используется специальный, предназначенный для этого блок, включающий в себя базу данных MS SQL Server.

Блок, содержащий базу данных представляет собой набор специальных данных с атрибутами, по которым можно выполнить процедуру поиска или выбора информации. Данный блок предоставляет возможность использования «агрегатора» без полного или частичного изменения устоявшегося регламента компании, то есть помогает в полной мере оценить возможность обработки данных в режиме реального времени, что является основной целью создания «агрегатора».

Реализация базы данных (в том числе и клиентская часть) осуществляется на той же вычислительной технике, на которой будет установлено основное программное обеспечение «агрегатора», это дает целостность информационной системы, а также организует достаточную для работы скорость чтения и записи информации в базу данных.

При проектировании и создании базы данных будем руководствоваться набором следующих правил:

- Пользователь имеет определенную, закрепленную за ним роль;
- Станция ГТИ находится на определенном месторождении;
- Информация может поступать из нескольких станций ГТИ;
- Известно время поступления данных;
- Месторождения объединены в логические группы;
- Формат данных, которые хранятся в базе данных ограничивается форматами LAS и WITS.

MS SQL Server имеет набор встроенных возможностей и набор predefined ролей. Пользователи РСУБД могут создавать свои роли и управлять предоставлением прав доступа к ним. Имеется возможность создания иерархии привилегий. Все роли и привилегии хранятся в таблицах, владельцем которых является *sysadmin*. Основная таблица, содержащая информацию о ролях и их возможностях, описана ниже.

Таблица 2 Таблицы, содержащие информацию о ролях и привилегиях

| <b>Предопределенная роль уровня сервера</b> | <b>Описание</b>  |
|---|--|
| sysadmin                                    | Члены предопределенной роли сервера sysadmin могут выполнять любые действия на сервере.  |
| serveradmin                                 | Члены предопределенной роли сервера serveradmin могут изменять параметры конфигурации на уровне сервера, а также выключать сервер.   |
| securityadmin                               | <p>Элементы предопределенной роли сервера securityadmin управляют именами входа и их свойствами. Они могут предоставлять, запрещать и отменять разрешения на уровне сервера (инструкции GRANT, DENY и REVOKE). Они также могут предоставлять, запрещать и отменять разрешения на уровне базы данных (инструкции GRANT, DENY и REVOKE) при наличии доступа к базе данных. Кроме того, они могут сбрасывать пароли для имен входа SQL Server.</p> <p><b>**Примечание по безопасности.**</b> Возможность предоставления доступа к компоненту Компонент Database Engine и настройки разрешений пользователей позволяют администратору безопасности назначать большинство разрешений сервера. Роль securityadmin должна считаться эквивалентной роли sysadmin .</p> |
| processadmin                                | Члены предопределенной роли сервера processadmin могут завершать процессы, работающие на экземпляре SQL Server.  |
| setupadmin                                  | Члены предопределенной роли сервера setupadmin могут добавлять или удалять связанные серверы с помощью инструкций Transact-SQL. (Членство в роли sysadmin  |



|           |   |
|-----------|---|
|           | необходимо при использовании службы Среда Management Studio.)   |
| bulkadmin | Члены предопределенной роли сервера bulkadmin могут выполнять инструкцию BULK INSERT.   |
| diskadmin | Предопределенная роль сервера diskadmin используется для управления файлами на диске.   |
| dbcreator | Члены предопределенной роли сервера dbcreator могут создавать, изменять, удалять и восстанавливать любые базы данных.   |
| public    | <p>Каждое имя входа SQL Server принадлежит к роли сервера public. Если для участника на уровне сервера не были предоставлены или запрещены конкретные разрешения на защищаемый объект, он наследует разрешения роли public на этот объект. Разрешения роли public следует назначать только тому объекту, который будет доступен всем пользователям. Нельзя изменить членство в роли public.</p> <p>Примечание. Роль public реализована не так, как другие роли. Однако разрешения для роли public можно назначать, отменять или отзывать.</p> |

### 3.4 Блок схема и алгоритм работы базы данных.

Для возможности быстрого поиска необходимых исходных файлов в формате las 2.0, 3.0 и wits 1.0 необходимо спроектировать базу данных.

В каждом файле las и wits есть обязательные поля, которые всегда заполняются, и необязательные, в которых информация заполняется по необходимости. На основе обязательных полей и проектируется наша база данных.

Полученная схема базы данных представлена на рисунке 21.

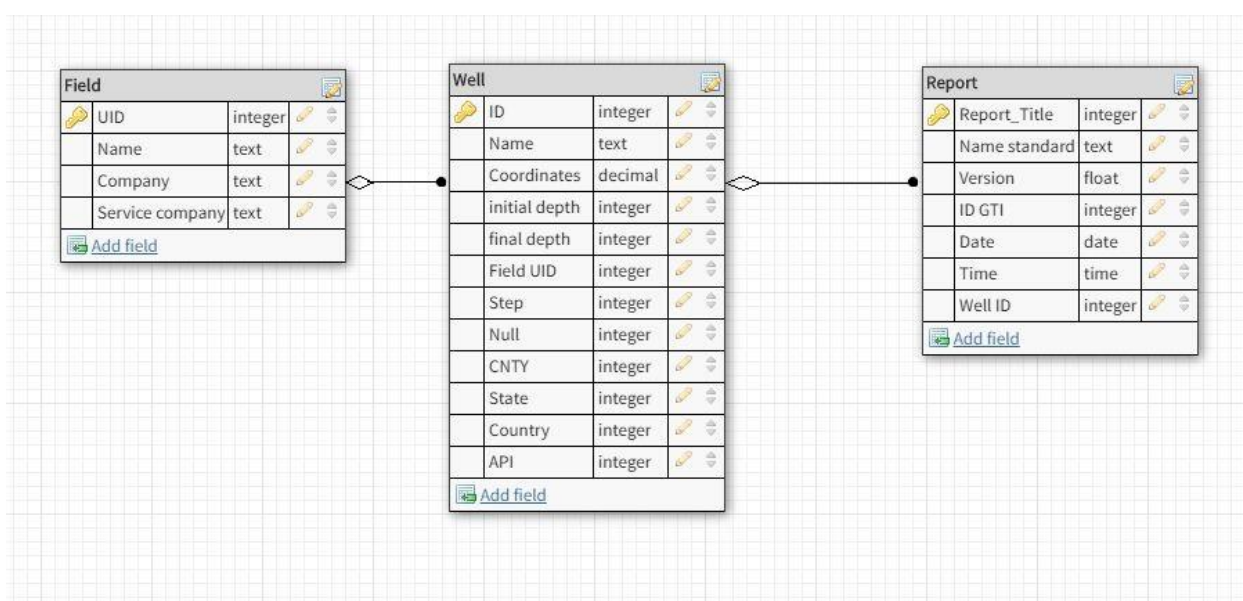


Рис. 21. Структура базы данных

Полученная база данных имеет три таблицы Well (Скважина), Field (Месторождение) и Report (отчет). Связь между таблицами один ко многим, что означает, что в одном месторождении находятся несколько скважин, а у каждой скважины есть множество отчетов. Каждое месторождение имеет свой уникальный ID, название, скважины, наименование компании-заказчика и название геофизической компании. У каждой скважины также есть уникальный ID, название, координаты скважины, начальная глубина замеров и конечная глубина замеров, шаг данных в колонке глубин, отсутствие данных на определённой глубине, район, область, страна, где находится скважина, API. Внешним ключом в этой таблице является UID месторождения. В таблице Report

находится информации о полученном отчете – название отчета, название стандарта, версия стандарта, ID станции ГТИ, с которой получен отчет, дата и время получения. Здесь внешним ключом является ID скважины.

### 3.5 Разработка запросов к БД

В рамках данной ВКР для создания и управления базой данных была выбрана СУБД Microsoft SQL Server. Основным языком запросов является Transcat-SQL. Утилита содержит редактор скриптов и графическую программу, которая работает с объектами и настройками сервера. Для работы с базой данных используются SQL запросы в базу данных. SQL-запросы состоят из операторов, Transcat-SQL включает в себя следующие операторы:

ALL – сравнение для набора данных;

AND – для того, чтобы условие было выполнено, два выражения должны быть истинными;

ANY - для того, чтобы условие было выполнено, хотя бы одно выражение должно быть истинным. Выполняется для набора данных;

BETWEEN – проверка наличия данных в выбранном диапазоне;

EXIST – проверка существования данных;

IN – проверка вхождения значения в указанный список;

LIKE – проверка значения на идентичность шаблону;

OR – условие будет выполнено, если хотя бы одно из двух выражений будет верно;

SOME – сравнения наборов данных.

Работа с текстовыми данными отличается от работы с символьными данными. Для работы с текстом существуют специальные функции:

DATALENGTH – для определения количества байт, которые занимает определенная величина;

PATINDEX – осуществляет поиск подстроки;

SUBSTRING – возврат подстроки;

TEXTPTR – возврат указателя на текст;

TEXTVALID – определение правильности указателя;  
READTEXT – считывание данных с текста;  
UPDATETEXT – изменение данных в текстовом поле;  
WRITETEXT – перезапись текста.

Для сложных команд, которые состоят из двух и более выражений, применяется группировка с помощью конструкции BEGIN...END. В начале блока записывается BEGIN, далее идет одна или несколько команд, в конце блока ставится END. Блоки данной конструкции могут быть вложенными.

Для задания определенных условий используется конструкция IF...ELSE. Синтаксис конструкции имеет вид:

```
IF Boolean_expression  
{sql_statement | statement_block}  
[ELSE  
{sql_statement | statement_block}]
```

Если логическая команда, следующая за IF выполняется, то следом выполняется первая команда sql\_statement или блок команд statement\_block. Если же значение неверно, выполняется команда или блок команд, следующие после ELSE. В некоторых случаях, если значение неверно, то ничего не должно выполняться. Тогда блок ELSE не пишут.

Более простыми конструкциями являются SELECT, INSERT, UPDATE, DELETE, MERGE. Они означают выборку, вставку, обновление, удаление и слияние данных соответственно.

Для создания таблицы месторождений с уникальным идентификатором UID используем SQL-запрос CREATE:

```
CREATE TABLE Field (  
    UID int NOT NULL CONSTRAINT PK_Field PRIMARY KEY,  
    Name nvarchar(30),  
    Company nvarchar(30),  
    Service company nvarchar(30),  
)
```

Для того, чтобы поле обязательно было заполнено, необходимо использовать опцию NOT NULL. Для обновления записи в существующей таблице используем следующие команды:

-- обновление поля ID

```
ALTER TABLE Field ALTER COLUMN UID int NOT NULL
```

Для заполнения таблицы данными используем команду INSERT:

```
INSERT Field (UID,Name,Company,Service company) VALUES  
(1000,N'Двуреченское',N'ТомскНефть',N'Градиент'),  
(1001,N'Игло-Полуденское',N'Газпром',N'ГеоПрайм'),  
(1002,N'Западно-Лугинское',N'Газпром',N'ГеоПрайм'),  
(1003,N'Планерное',N'ТомскНефть',N'Градиент')
```

Аналогичным образом создаем и еще две таблицы – скважины и отчеты.

```
CREATE TABLE Well (  
    ID int NOT NULL CONSTRAINT PK_Well PRIMARY KEY,  
    Name nvarchar(30) NOT NULL,  
    Coordinates decimal NOT NULL  
    Intal depth int NOT NULL,  
    Final depth int NOT NULL,  
    Field UID int NOT NULL,  
    Step int NOT NULL,  
    Null int,  
    CNTY int,  
    State int,  
    Country int,  
    API int  
)  
CREATE TABLE Report (  
    ID int NOT NULL CONSTRAINT PK_Report PRIMARY KEY,  
    Report title int NOT NULL,  
    Name standart nvarchar(30) NOT NULL,
```

```
Version decimal NOT NULL,  
ID GTI int,  
Date datetime NOT NULL,  
Well ID int NOT NULL  
)
```

Для получения нужных полей в нужной последовательности выполняем следующий запрос, в результате которого получим поля UID и название месторождения из таблицы месторождений:

```
SELECT UID, Name  
FROM [TestDB].[dbo].[Field];
```

Для того, чтобы выбрать записи отчетов за 12 июня 2016 года выполняем следующий скрипт:

```
SELECT dt, product_id, title, amount, price  
FROM TestDB.dbo.Report INNER JOIN TestDB.dbo.Well  
ON well.id=report.well id  
dt='2016-06-12';
```

### **3.6 Веб интерфейс «агрегатора».**

Для комфортного взаимодействия оператора с «агрегатором» был разработан веб интерфейс, который имеет достаточно простое строение и позволяет эффективно работать с данными даже неопытному пользователю. В основе интерфейса лежит авторизация конкретного сотрудника по его уникальному ID и паролю. Также разрабатываемый интерфейс должен содержать следующие возможности:

1) отображать список файлов, которые находятся на данный момент в буфере данных (ждут очередь на конвертацию);

2) отображать список файлов, которые в данный момент проходят этап конвертации (рисунок 22);

3) показывать ссылку на список, который бы показал содержимое базы данных входящего буфера (всех принятых файлов);

4) показывать ссылку на список всех отправленных сконвертированных файлов;

5) так же на страничке можно указать ID станции гти откуда приходят файлы;

6) посмотреть адрес WITSML сервера, на который они уходят.

Веб интерфейс должен содержать также страницы: со списком принятых файлов (рисунок 23), со списком отправленных (рисунок 24), для удобства чтения с каждым файлом отображаются даты принятия и отправки, а для удобства выборка имеется функция поиска по названию или выборка по конкретному параметру (параметры считываются из исходного принятого файла).

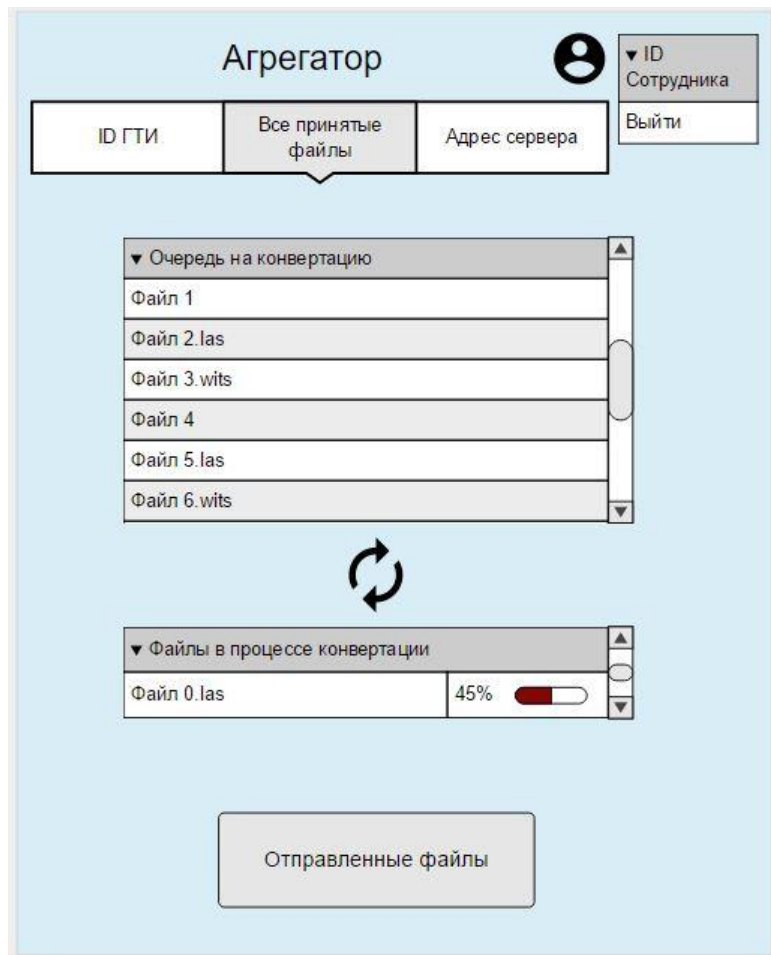


Рис. 22 - Главная страница "Агрегатора"



**Агрегатор**

ID ГТИ

Все принятые файлы

Адрес сервера

▼ ID Сотрудника  
 Выйти

Выбрать параметр ▼

Search...

| ▼ Принятые файлы | ▼ дата принятия | ▼ время |
|------------------|-----------------|---------|
| Файл 1           | 12.04.2017      | 12:04   |
| Файл 2.las       | 12.04.2017      | 13:45   |
| Файл 3.wits      | 12.04.2017      | 15:00   |
| Файл 4           | 12.04.2017      | 15:30   |
| Файл 5.las       | 12.04.2017      | 15:34   |
| Файл 6.wits      | 12.04.2017      | 15:38   |
| Файл 7           | 12.04.2017      | 16:00   |
| Файл 8.las       | 13.04.2017      | 12:04   |
| Файл 9.wits      | 13.04.2017      | 12:04   |
| Файл 1           | 12.14.2017      | 12:04   |
| Файл 2.las       | 14.04.2017      | 12:04   |
| Файл 3.wits      | 15.04.2017      | 12:04   |
| Файл 4           | 15.04.2017      | 12:04   |
| Файл 5.las       | 16.04.2017      | 12:04   |
| Файл 6.wits      | 16.04.2017      | 12:04   |

Рис. 23 - Страница принятых файлов

Агрегатор

Отправленные файлы

Выбрать параметр

Search...

| ▼ Отправленные файлы | ▼ дата принятия | ▼ время |
|----------------------|-----------------|---------|
| Файл 0.witsml        | 12.04.2017      | 13:00   |
| Файл 1.witsml        | 12.04.2017      | 13:04   |
| Файл 2.witsml        | 12.04.2017      | 13:08   |
| Файл 3.witsml        | 12.04.2017      | 13:12   |
| Файл 4.witsml        | 12.04.2017      | 13:40   |
| Файл 5.witsml        | 12.04.2017      | 16:00   |
| Файл 6.witsml        | 13.04.2017      | 10:00   |
| Файл 7.witsml        | 13.04.2017      | 13:00   |
| Файл 8.witsml        | 13.04.2017      | 18:00   |
| Файл 9.witsml        | 14.04.2017      | 13:00   |
| Файл 10.witsml       | 15.04.2017      | 13:00   |
| Файл 11.witsml       | 15.04.2017      | 13:00   |
| Файл 12.witsml       | 16.04.2017      | 13:00   |
| Файл 13.witsml       | 17.04.2017      | 13:00   |

▼ ID  
Сотрудника  
Выйти

Рис. 24 - Страница отправленных файлов

## Раздел 4. Организация очереди на конвертирование. Алгоритмы подготовки файлов на конвертирование.

### 4.1 Организация очереди на конвертирование (метод FIFO)

На «агрегатор» данные могут поступать из нескольких станций ГТИ, с разных месторождений, поэтому важно иметь возможность управлять очередностью передачи файлов на конвертирование.

Организация очереди организована в блоке конвертации программным способом, который напоминает по типу реализации стеки, то есть они не дают доступа к произвольным элементам, но в отличие от стека в очередь элементы кладутся и забираются с разных концов. Такой метод называется «First-In-First-Out» или «первый вошел, первый вышел». Таким образом обрабатываться элементы из очереди будут в том же порядке, что и попадают в эту очередь (рисунок 25). Такой метод очень похож на конвейер.

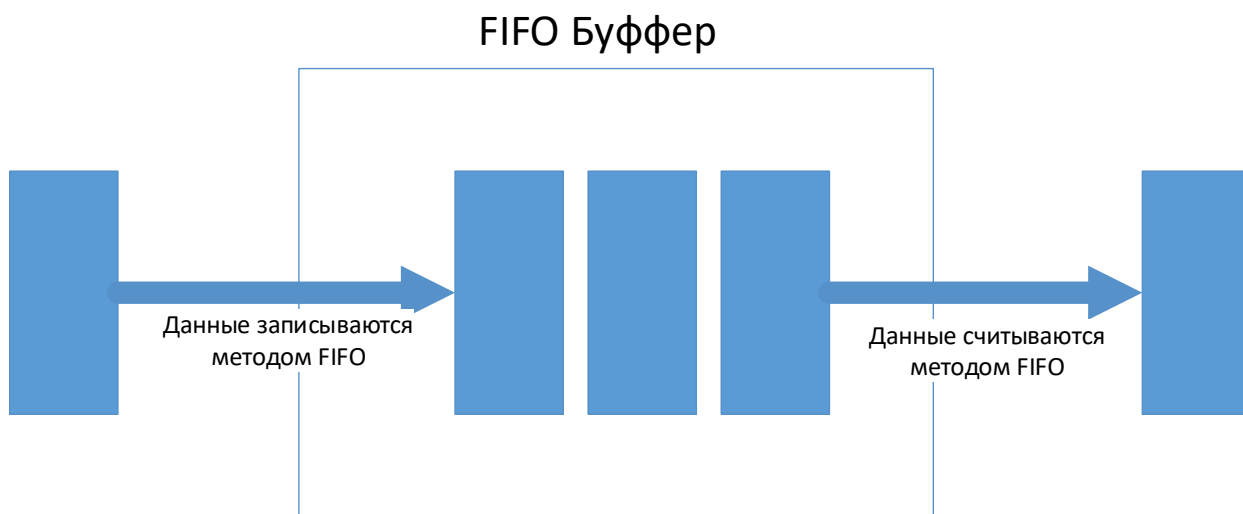


Рис. 25 - Организация очереди

Очереди зачастую используют в программных средствах для реализации буфера, в который можно положить элемент для последующей обработки, сохраняя порядок поступления. Например, если база данных поддерживает только одно соединение, можно использовать очередь потоков, которые будут ждать своей очереди на доступ к БД.

Таким образом каждый поступающим файл со станции ГТИ будет храниться в буфере до тех пор, пока поступивший ранее файл не пройдет процесс конвертации.

Класс Queue реализован с помощью связанного списка. Он будет предоставлять методы Enqueue для добавления элемента, Dequeue для удаления, Peek и Count. Он не будет реализовывать интерфейс ICollection<T>, поскольку это коллекции специального назначения.

```
1 public class Queue
2 {
3     LinkedList _items = new LinkedList();
4
5     public void Enqueue(T value)
6     {
7         throw new NotImplementedException();
8     }
9
10    public T Dequeue()
11    {
12        throw new NotImplementedException();
13    }
14
15    public T Peek()
16    {
17        throw new NotImplementedException();
18    }
19
20    public int Count
21    {
22        get;
23    }
24 }
```

Метод Enqueue

- **Поведение:** Добавляет элемент в очередь.
- **Сложность:**  $O(1)$ .

Новые элементы очереди можно добавлять, как в начало списка, так и в конец. Важно только, чтобы элементы доставались с противоположного края. В данной реализации будем добавлять новые элементы в начало внутреннего списка.

```

1  Public void Enqueue(T value)
2  {
3      _items.AddFirst(value);
4  }
```

### Метод Dequeue

- **Поведение:** Удаляет первый помещенный элемент из очереди и возвращает его. Если очередь пустая, кидает `InvalidOperationException`.
- **Сложность:**  $O(1)$ .

Поскольку элементы попадают в начало списка, убирать мы их будем с конца.

Если список пуст, кидается исключение.

```

1  public T Dequeue()
2  {
3      if (_items.Count == 0)
4      {
5          throw new InvalidOperationException("The queue is empty");
6      }
7
8      T last = _items.Tail.Value;
9
10     _items.RemoveLast();
11
12     return last;
13 }
```

### Метод Peek

- **Поведение:** Возвращает элемент, который вернет следующий вызов метода Dequeue. Очередь остается без изменений. Если очередь пустая, кидает InvalidOperationException.
- **Сложность:** O(1).

```
1 public T Peek()
2 {
3     if (_items.Count == 0)
4     {
5         throw new InvalidOperationException("The queue is empty");
6     }
7
8     return _items.Tail.Value;
9 }
```

#### Метод Count

- **Поведение:** Возвращает количество элементов в очереди или 0, если очередь пустая.
- **Сложность:** O(1).

```
1 public int Count
2 {
3     get
4     {
5         return _items.Count;
6     }
7 }
```

## Раздел 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение.

### 5.1. Введение

Для оценки коммерческой привлекательности любого проекта, вне зависимости от того, какой характер он несёт, научный или практический, необходимо определить не только превышение технических параметров над предыдущими разработками, но и дать ответы на такие вопросы – сколько будет стоить проект, каков бюджет данного проекта, будет ли полученный продукт востребован на рынке, сколько потребуется времени для выхода продукта на рынок и т.д. В рамках данной исследовательской работы исследуется проект блока конвертеров данных из стандартов LAS и WITS в стандарты WITSML 1.3.1 и 1.4.1, в режиме реального времени.

### 5.2. Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

#### 5.2.1. Потенциальные потребители продукта исследования

Основными потребителями услуг блока конвертеров данных из стандартов LAS и WITS в стандарты WITSML 1.3.1 и 1.4.1 являются отечественные и зарубежные компании, занимающиеся нефтедобычей, бурением скважин, разработкой месторождений, а также компании, владеющие буровыми установками. Данные о сегментировании рынка продаж услуг блока конвертеров данных предоставлены в таблице 3.

Таблица 3. Карта сегментирования рынка продаж услуг блока конвертеров данных

|                         |                        | Местоположение блока конвертеров |                                |
|-------------------------|------------------------|----------------------------------|--------------------------------|
|                         |                        | Зарубежный блок конвертеров      | Отечественный блок конвертеров |
| Местоположение компании | Зарубежные компании    |                                  |                                |
|                         | Отечественные компании |                                  |                                |

Исходя из данных о сегментировании рынка, можно сделать вывод, что сегмент свободен. Целевым сегментом продажи нашей продукции будут являться отечественные компании с размещением и разработкой отечественных агрегаторов.

### 5.2.2. Анализ конкурентов

Для анализа конкурирующих разработок была составлена оценочная карта для сравнения конкурентных разработок. Таким образом, была рассмотрена конкурентоспособность относительно уже существующей компании, предоставляющей услуги конвертации данных из формата LAS и WITS в форматы WITSML 1.3.1 и 1.4.1 – Petrodaq (k1) и Petrolink (k2).

Для того, чтобы проанализировать конкурентные технические решения, необходимо использовать формулу:

$$K = \sum B_i * B_i, \tag{1}$$

где  $K$  – является ли проект или научное исследование конкурентоспособным;



$B_i$  – вес  $i$ -го критерия (в долях единицы);

$B_i$  – балл  $i$ -го показателя.

$K_{ф1} = 0,06 * 4 = 0,24$ . Прочие вычисления выполнены по аналогии.

Таблица 4. Карта оценки конкурентоспособности проекта

| Показатели  | Вес критерия | Баллы |     |     | Конкурентоспособность |      |      |
|---|--------------|-------|-----|-----|-----------------------|------|------|
|   |              | Бф    | Бк1 | Бк2 | Кф                    | Кк1  | Кк2  |
| 1   | 2            | 3     | 4   | 5   | 6                     | 7    | 8    |
| <b>Технические критерии оценки ресурсоэффективности</b> |              |       |     |     |                       |      |      |
| 1. Удобство в эксплуатации                              | 0,06         | 4     | 4   | 3   | 0,24                  | 0,24 | 0,18 |
| 2. Надежность   | 0,1          | 4     | 4   | 3   | 0,4                   | 0,4  | 0,3  |
| 3. Увеличение производительности труда пользователя     | 0,12         | 4     | 4   | 2   | 0,48                  | 0,48 | 0,24 |
| 4. Помехоустойчивость                                   | 0,05         | 3     | 4   | 4   | 0,15                  | 0,2  | 0,2  |
| 5. Качество интерфейса                                  | 0,05         | 4     | 4   | 4   | 0,2                   | 0,2  | 0,2  |
| 6. Легкость использования                               | 0,03         | 4     | 3   | 4   | 0,12                  | 0,09 | 0,12 |
| 7. Количество ресурсов памяти                           | 0,07         | 3     | 4   | 3   | 0,21                  | 0,28 | 0,21 |
| 8. Функциональность продукта                            | 0,13         | 4     | 3   | 4   | 0,52                  | 0,39 | 0,52 |
| 9. Отсутствие потери данных                             | 0,1          | 3     | 4   | 4   | 0,3                   | 0,4  | 0,4  |
| <b>Экономические критерии оценки эффективности</b>      |              |       |     |     |                       |      |      |
| 10. Уровень конкурентоспособности продукта              | 0,08         | 4     | 5   | 4   | 0,32                  | 0,4  | 0,32 |
| 11. Стоимость   | 0,05         | 4     | 3   | 2   | 0,2                   | 0,15 | 0,1  |
| 12. Срок эксплуатации                                   | 0,05         | 4     | 4   | 4   | 0,2                   | 0,2  | 0,2  |

|                                     |          |           |           |           |             |             |             |
|-------------------------------------|----------|-----------|-----------|-----------|-------------|-------------|-------------|
| 13. Поддержка продукта              | 0,07     | 3         | 3         | 4         | 0,21        | 0,21        | 0,28        |
| 14. Способность проникнуть на рынок | 0,04     | 4         | 3         | 4         | 0,16        | 0,12        | 0,16        |
| Итого                               | <b>1</b> | <b>52</b> | <b>52</b> | <b>49</b> | <b>3,71</b> | <b>3,76</b> | <b>3,43</b> |

Найдем коэффициент конкурентоспособности предприятия

$$K_{kc} = K_{\phi} / K_{\phi ko} = (3,71/3,76 + 3,71/3,43)/2 = 1,03.$$

Так как найденный коэффициент больше 1, наш проект конкурентоспособен.

### 5.2.3. Технология QuaD

В данном разделе мы рассмотрим конкурентоспособность нашего проекта с помощью оценочной карты QuaD.

Таблица 5. Карта оценки конкурентных разработок

| Показатели  | Вес критерия | Баллы | Максимальный балл | Относительное значение | Средневзвешенное значение |
|---|--------------|-------|-------------------|------------------------|---------------------------|
| 1   | 2            | 3     | 4                 | 5                      | 6                         |
| <b>Показатели оценки качества разработки</b>        |              |       |                   |                        |                           |
| 1. Удобство в эксплуатации                          | 0,1          | 30    | 100               | 0,3                    | 0,03                      |
| 2. Надежность                                       | 0,1          | 70    | 100               | 0,7                    | 0,07                      |
| 3. Увеличение производительности труда пользователя | 0,1          | 80    | 100               | 0,8                    | 0,08                      |
| 4. Помехоустойчивость                               | 0,05         | 70    | 100               | 0,7                    | 0,035                     |
| 5. Качество интерфейса                              | 0,05         | 85    | 100               | 0,85                   | 0,0425                    |
| 6. Легкость использования                           | 0,1          | 90    | 100               | 0,9                    | 0,09                      |
| 7. Количество ресурсов памяти                       | 0,05         | 90    | 100               | 0,9                    | 0,045                     |

| Показатели   | Вес критерия | Баллы | Максимальный балл | Относительное значение | Средневзвешенное значение |
|--|--------------|-------|-------------------|------------------------|---------------------------|
| 1  | 2            | 3     | 4                 | 5                      | 6                         |
| 8. Функциональность продукта                                 | 0,1          | 70    | 100               | 0,7                    | 0,07                      |
| 9. Отсутствие потери данных                                  | 0,1          | 3     | 4                 | 4                      | 0,3                       |
| <b>Показатели оценки коммерческого потенциала разработки</b> |              |       |                   |                        |                           |
| 10. Уровень конкурентоспособности продукта                   | 0,08         | 4     | 5                 | 4                      | 0,32                      |
| 11. Стоимость  | 0,05         | 4     | 3                 | 2                      | 0,2                       |
| 12. Срок эксплуатации  | 0,05         | 4     | 4                 | 4                      | 0,2                       |
| 13. Поддержка продукта                                       | 0,07         | 3     | 3                 | 4                      | 0,21                      |
| 14. Способность проникнуть на рынок                          | 0,04         | 4     | 3                 | 4                      | 0,16                      |
| Итого  | 1            | 52    | 52                | 49                     | 3,71                      |

$$P_{cp} = \sum V_i \cdot B_i, \quad (2)$$

$P_{cp1} = 0,1 * 0,3 = 0,03$ . Прочие вычисления выполнены по аналогии.

где  $P_{cp}$  – средневзвешенное значение оценки качества и перспектив научной разработки;

$V_i$  – вес критерия (в долях единицы);

$B_i$  – средневзвешенное значение  $i$ -го показателя.

Значение  $P_{cp}$  составило 72,25, что показывает перспективность разработки выше среднего.

#### 5.2.4. SWOT-анализ

Посредством SWOT-анализа выявим внешние и внутренние факторы среды проекта, и выявим степень соответствия сильных и слабых сторон проекта внешним условиям окружающей среды.

Таблица 6. Матрица SWOT

|             |  | Сильные стороны   | Слабые стороны  |
|-------------|--|---|---|
|             |  | <p>С1. Высокое качество программного продукта.</p> <p>С2. Интуитивно понятный интерфейс.</p> <p>С3. Удобство в использовании.</p> <p>С4. Быстрая установка программного продукта.</p> <p>С5. Экономичность ресурсов памяти.</p> | <p>Сл1. Небольшой функционал.</p> <p>Сл2. Локальное приложение.</p> <p>Сл3. Уступает конкурентам в быстродействии.</p> <p>Сл4. Неполная оптимизация функций.</p> <p>Сл5. Ограниченное количество ресурсов</p> |
| Возможности | <p>В1. Появление дополнительного спроса на продукт.</p> <p>В2. Удовлетворение новой потребности потребителя.</p> <p>В3. Использование дополнительных ресурсов.</p> <p>В4. Появление дополнительных финансовых ресурсов.</p> <p>В5. Повышение стоимости конкурентных разработок</p> | <p>1. Постоянная оптимизация продукта для более удобного пользования.</p> <p>2. Добавление новых функций в программный продукт.</p>   | <p>1. Улучшение быстродействия программного продукта.</p> <p>2. Повышение конкурентоспособности.</p>  |

|        |  |  |   |
|--------|--|--|---|
| Угрозы | <p>У1. Отсутствие спроса.</p> <p>У2. Выход продукта – конкурента с полным функционалом.</p> <p>У3. Выход продукта – конкурента с более полной информацией об оборудовании.</p> <p>У4. Проблемы с материально-техническим обеспечением</p> <p>У5. Введение дополнительных государственных требований к сертификации продукции</p> | <p>1. За счет повышения качества продукта увеличить спрос на рынке.</p> <p>2. Уменьшить использование ресурсов оперативной и локальной памяти.</p> | <p>1. Увеличить функционал продукта.</p> <p>2. Улучшить оптимизацию и уменьшить затраты ресурсов.</p> |
|--------|--|--|---|

Таблица 7. Интерактивная матрица проекта

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

Направления реализации проекта: B1C1C2C3C4C5, B3C1C3, B4C1C2C4C5, B5C1C3.

Таблица 8. Интерактивная матрица проекта

| Слабые стороны |    |     |     |     |     |     |
|----------------|----|-----|-----|-----|-----|-----|
| Возможности    |    | Сл1 | Сл2 | Сл3 | Сл4 | Сл5 |
|                | В1 | -   | -   | -   | -   | -   |
|                | В2 | 0   | -   | 0   | +   | -   |
|                | В3 | -   | -   | -   | -   | -   |
|                | В4 | +   | -   | +   | -   | +   |
|                | В5 | -   | -   | -   | -   | -   |

Направление реализации проекта: В2Сл4, В4Сл1Сл3Сл5.

Таблица 9. Интерактивная матрица проекта

| Сильные стороны |    |    |    |    |    |    |
|-----------------|----|----|----|----|----|----|
| Угрозы          |    | С1 | С2 | С3 | С4 | С5 |
|                 | У1 | -  | +  | -  | -  | +  |
|                 | У2 | -  | +  | -  | +  | +  |
|                 | У3 | +  | 0  | -  | -  | -  |
|                 | У4 | -  | 0  | -  | -  | -  |
|                 | У5 | 0  | -  | -  | +  | +  |

Направления реализации проекта: У1С2С5, У2С2С4С5, У3С1, У5С4С5.

Таблица 10. Интерактивная матрица проекта

| Угрозы | Слабые стороны |     |     |     |     |
|--------|----------------|-----|-----|-----|-----|
|        |                | Сл1 | Сл2 | Сл3 | Сл4 |
| У1     | +              | -   | -   | -   | -   |
| У2     | -              | -   | -   | -   | -   |
| У3     | +              | -   | -   | -   | +   |
| У4     | +              | +   | +   | +   | +   |
| У5     | -              | +   | -   | +   | -   |

Направления реализации проекта: У1Сл1, У3Сл1Сл5, У4Сл1Сл2Сл3Сл4Сл5, У5Сл2Сл4.

### 5.3. Нахождение альтернативных способов проведения научных исследований

Для нахождения возможных альтернативных способов проведения научных исследований воспользуемся морфологическим методом. Для этого построим морфологическую матрицу.

Таблица 11. Морфологическая матрица для веб-интерфейса агрегатора

|                               | 1                          | 2                  | 3                     | 4                       | 5              |
|-------------------------------|----------------------------|--------------------|-----------------------|-------------------------|----------------|
| А. Тип клиентского интерфейса | Веб-портал                 | Веб - страница     | Веб-приложение        | Стационарное приложение | Нет интерфейса |
| Б. Тип хранения данных        | Сервер хранения            | Система хранения   | Локальные базы данных |                         |                |
| В. Доступ к данным            | В режиме реального времени | По запросу клиента | Нет доступа           |                         |                |

|                                 | 1                                  | 2   | 3                     | 4                   | 5 |
|---------------------------------|------------------------------------|---|-----------------------|---------------------|---|
| Г. Процесс составления договора | Полностью автоматизированный       | Частично автоматизированный                 | Полностью ручной      | Отсутствие договора |   |
| Д. Составление тарифного плана  | Индивидуальное для каждого клиента | Общее для всех клиентов                     |                       |                     |   |
| Е. Расчет стоимости услуги      | Индивидуальный для каждого клиента | Строго определенная сумма для всех клиентов |                       |                     |   |
| Ж. Способ оплаты услуги         | Банковским переводом               | Электронные системы оплаты                  | Личная оплата в кассу |                     |   |

С помощью морфологического анализа были выделены следующие преимущественные варианты:

- 1) А1Б2В1Г2Д1Е1Ж1 – разработка веб-портала для агрегатора, с использованием глобальной системы хранения и предоставлением доступа клиентам к данным скважин в режиме реального времени, а также с обеспечением, частично автоматизированного документооборота, и индивидуальными расчетами тарифных планов и стоимости услуг для каждого клиента, оплачиваемых посредством банковского перевода.
- 2) А2Б1В2Г1Д2Е2Ж2 – разработка веб-страницы для агрегатора, с использованием серверов хранения данных и предоставления доступа к данным по запросу клиента, а также с обеспечением полностью автоматизированного документооборота, общие тарифные планы и стоимости услуг для всех клиентов, и оплата посредством электронных платежных систем.
- 3) А3Б2В1Г1Д1Е1Ж1 – разработка веб-приложения для агрегатора, с использованием глобальной системы хранения и предоставлением доступа клиентам к данным скважин в режиме реального времени, а также с



обеспечением, полностью автоматизированным документооборотом, и индивидуальными расчетами тарифных планов и стоимости услуг для каждого клиента, оплачиваемых посредством банковского перевода.

#### 5.4. Создание научно-исследовательских работ

##### 5.4.1. Этапы планирования работ в рамках научного исследования

Таблица 12 Список этапов, содержание работ и распределение ролей

| Этапы                     | № раб | Содержание работ  | Должность исполнителя  |
|---------------------------|-------|---|------------------------|
| Разработка ТЗ             | 1     | Выбор темы  | И1, И2<br>Руководитель |
|                           | 2     | Составление и утверждение ТЗ  | Руководитель           |
| Анализ предметной области | 3     | Создание календарного плана работ   | И1, И2                 |
|                           | 4     | Выбор и изучение теоретических материалов по теме   | И1, И2                 |
|                           | 5     | Анализ изученного материала   | И1, И2<br>Руководитель |
|                           | 6     | Проработка литературных источников и периодической литературы   | И1, И2                 |
| Основная                  | 7     | Анализ современного состояния стандартизации данных в нефтяной и газовой промышленности   | И2                     |
|                           | 8     | Разработка состава номенклатуры национального стандарта передачи данных   | И1, И2                 |
|                           | 9     | Проектирование основных функций и блок схемы национального стандарта передачи данных  | И1                     |
|                           | 10    | Проектирование и разработка средства мониторинга и предоставления продукта конечному пользователю сервера национального стандарта | И2                     |
|                           | 11    | Разработка системы формирования финансовой документации   | И1                     |
|                           | 12    | Тестирование и доработка системы  | И1, И2                 |
| Заключительная            | 13    | Написание пояснительной записки   | И1, И2                 |

|  |    |  |        |
|--|----|--|--------|
|  | 14 | Создание и оформление презентации дипломного проекта | И1, И2 |
|--|----|--|--------|

Для выполнения научного исследования формируется рабочая группа, в состав которой входят 2 студента-дипломника и один руководитель. Порядок этапов и работ, распределение исполнителей по данным видам работ приведен в таблице.

#### 5.4.2. Расчет трудоемкости исполнения работ

Чтобы определить ожидаемое значение продолжительности работ  $t_{ож}$  применяются две оценки:  $t_{min}$  и  $t_{max}$  (метод двух оценок).

$$t_{ож} = \frac{3 \cdot t_{min} + 2 \cdot t_{max}}{5}, \quad (3)$$

$t_{ож} = (3 \cdot 2 + 2 \cdot 3) / 5 = 1,2$ . Прочие вычисления выполнены по аналогии.

где  $t_{min}$  – наименьшая трудоемкость работ, чел/дн.;

$t_{max}$  – наибольшая трудоемкость работ, чел/дн.

После нахождения значения ожидаемой трудоемкости работ, необходимо определить длительность каждой работы в рабочих днях  $T_p$ , учитывая при этом возможность параллельности работ сразу несколькими исполнителями.

$$T_{pi} = \frac{t_{ожi}}{Ч_i}, \quad (4)$$

$T_{pi} = 1,2 / 3 = 0,4$ . Прочие вычисления выполнены по аналогии.

$T_{pi}$  – длительность одной работы, раб. дн.;

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

$Ч_i$  – количество исполнителей, которые выполняют одну и ту же работу параллельно, чел.

Для удобства построения графика, длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни.

$$T_{ki} = T_{pi} \cdot k_{\text{кал}}, \quad (5)$$

$T_{ki} = 0,4 \cdot 1,47 = 0,6$ . Округляем до 1. Прочие вычисления выполнены по аналогии.

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

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

$k$  – коэффициент календарности. кал

Коэффициент календарности определяется по следующей формуле:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}}, \quad (6)$$

$$k_{\text{кал}} = 365 / (365 - 105 - 13) = 1,47.$$

где  $T_{\text{кал}}$  – количество календарных дней в году;

$T_{\text{вых}}$  – количество выходных дней в году;

$T_{\text{пр}}$  – количество праздничных дней в году.

Для выполнения перечисленных в таблице 10 работ требуются специалисты: два студента, научный руководитель. Результаты расчетов представлены в таблице 11.

Таблица 13. Временные показатели проведения научного исследования

| № Работы  | Трудоёмкость работ |        |        |               |        |        |                |        |        | Исполнители |        |        | Длительность работ в рабочих днях<br>$T_{pi}$ |        |        | Длительность работ в календарных днях<br>$T_{ki}$ |        |        |
|---|--------------------|--------|--------|---------------|--------|--------|----------------|--------|--------|-------------|--------|--------|---|--------|--------|---|--------|--------|
|   | tmin, чел-дни      |        |        | tmax, чел-дни |        |        | тождi, чел-дни |        |        |             |        |        |   |        |        |   |        |        |
|   | Ис п.1             | Ис п.2 | Ис п.3 | Ис п.1        | Ис п.2 | Ис п.3 | Ис п.1         | Ис п.2 | Ис п.3 | Ис п.1      | Ис п.2 | Ис п.3 | Ис п.1  | Ис п.2 | Ис п.3 | Ис п.1  | Ис п.2 | Ис п.3 |
| Выбор темы  | 2                  | 2      | 2      | 3             | 3      | 3      | 1,2            | 1,2    | 1,2    | 3           | 3      | 3      | 0,4   | 0,4    | 0,4    | 1   | 1      | 1      |
| Составление и утверждение ТЗ                      | 4                  | 6      | 5      | 7             | 9      | 8      | 5,2            | 7,2    | 6,2    | 1           | 1      | 1      | 5,2   | 7,2    | 6,2    | 8   | 11     | 9      |
| Создание календарного плана работ                 | 1                  | 1      | 1      | 2             | 2      | 2      | 1,4            | 1,4    | 1,4    | 2           | 2      | 2      | 0,3   | 0,35   | 0,35   | 1   | 1      | 1      |
| Выбор и изучение теоретических материалов по теме | 10                 | 9      | 11     | 14            | 13     | 15     | 11,6           | 10,6   | 12,6   | 2           | 2      | 2      | 2,9   | 2,65   | 3,15   | 4   | 4      | 5      |
| Анализ изученного материала                       | 7                  | 8      | 10     | 10            | 11     | 13     | 8,2            | 9,2    | 11,2   | 3           | 3      | 3      | 1,6   | 1,84   | 2,24   | 2   | 3      | 3      |

|   |   |    |   |    |    |    |     |      |     |   |   |   |     |      |      |   |   |   |
|---|---|----|---|----|----|----|-----|------|-----|---|---|---|-----|------|------|---|---|---|
| Проработка литературных источников и периодической литературы | 3 | 3  | 3 | 6  | 6  | 6  | 4,2 | 4,2  | 4,2 | 2 | 2 | 2 | 1,0 | 1,05 | 1,05 | 2 | 2 | 2 |
| Анализ современного состояния стандартизации данных           | 7 | 10 | 8 | 14 | 18 | 13 | 9,8 | 13,2 | 10  | 1 | 1 | 1 | 2,4 | 3,3  | 2,5  | 4 | 5 | 4 |

|  |    |    |    |    |    |    |      |      |      |   |   |   |      |      |      |    |    |    |
|--|----|----|----|----|----|----|------|------|------|---|---|---|------|------|------|----|----|----|
| в нефтяной и газовой промышленности  |    |    |    |    |    |    |      |      |      |   |   |   |      |      |      |    |    |    |
| Разработка состава номенклатуры национального стандарта передачи данных              | 20 | 23 | 26 | 30 | 36 | 31 | 24   | 28,2 | 28   | 2 | 2 | 2 | 6    | 7,05 | 7    | 9  | 10 | 10 |
| Проектирование основных функций и блок схемы национального стандарта передачи данных | 31 | 31 | 31 | 50 | 50 | 50 | 38,6 | 38,6 | 38,6 | 1 | 1 | 1 | 38,6 | 38,6 | 38,6 | 57 | 57 | 57 |

|  |     |     |     |     |     |     |       |       |      |   |   |   |       |       |      |     |     |     |
|--|-----|-----|-----|-----|-----|-----|-------|-------|------|---|---|---|-------|-------|------|-----|-----|-----|
| Проектирование и разработка средства мониторинга и предоставления продукта конечному пользователю агрегатора | 31  | 31  | 31  | 50  | 50  | 50  | 38,6  | 38,6  | 38,6 | 1 | 1 | 1 | 38,6  | 38,6  | 38,6 | 57  | 57  | 57  |
| Разработка системы формирования финансовой документации  | 31  | 31  | 31  | 50  | 50  | 50  | 38,6  | 38,6  | 38,6 | 1 | 1 | 1 | 38,6  | 38,6  | 38,6 | 57  | 57  | 57  |
| Тестирование и доработка системы   | 31  | 31  | 31  | 50  | 50  | 50  | 38,6  | 38,6  | 38,6 | 2 | 2 | 2 | 38,6  | 38,6  | 38,6 | 57  | 57  | 57  |
| Написание пояснительной записки  | 8   | 11  | 16  | 14  | 20  | 24  | 10,4  | 14,6  | 19,2 | 2 | 2 | 2 | 2,6   | 3,65  | 4,8  | 4   | 5   | 7   |
| Создание и оформление презентации научного исследования  | 7   | 10  | 11  | 10  | 14  | 18  | 8,2   | 11,6  | 13,8 | 2 | 2 | 2 | 2,05  | 2,9   | 3,45 | 3   | 4   | 5   |
| Итого  | 197 | 211 | 221 | 317 | 339 | 340 | 245,2 | 262,2 | 269  | - | - | - | 180,3 | 186,3 | 187  | 268 | 276 | 277 |

$$K_{\text{кал}} = 365 / (365 - 118) = 1,48$$

По данным расчетам, программа будет разработана:

- в первом исполнении 268 дней
- во втором исполнении 276 дней
- в третьем исполнении 277 дней

Следовательно, можно сделать вывод, что в первом исполнении работы будет выполнена быстрее.

### 5.4.3. Бюджет научно-технического исследования

При распределении бюджета НТИ необходимо обеспечить полное и достоверное отражение всех видов расходов, связанных с его выполнением. В процессе формирования бюджета НТИ используется следующая группировка затрат по статьям:

- материальные затраты НТИ;
- затраты на специальное оборудование;
- основная заработная плата исполнителей темы;
  - дополнительная заработная плата исполнителей темы;
  - отчисления во внебюджетные фонды (страховые отчисления);
- накладные расходы.

#### 5.4.3.1. Расчет материальных затрат НТИ

Расчет материальных затрат осуществляется по формуле 7:

$$Z_m = (1 + k_T) \cdot \sum_{i=1}^m \Pi_i \cdot N_{расч i}$$

(7)

$Z_m = 1 * 400522 * 1 = 400522$ . Прочие вычисления выполнены по аналогии.

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

$N_{расхi}$  – количество материальных ресурсов  $i$ -го вида, планируемых к использованию при выполнении научного исследования (шт., кг, м, м<sup>2</sup> и т.д.);

$Ц_i$  – цена приобретения единицы  $i$ -го вида потребляемых материальных ресурсов (руб./шт., руб./кг, руб./м, руб./м<sup>2</sup> и т.д.);

$k_T$  – коэффициент, учитывающий транспортно-заготовительные расходы.

Для разработки данного продукта необходимы следующие материальные ресурсы:

- Система хранения данных
- Два сервера

Расчет материальных затрат представлен в таблице 14.

Таблица 14. Материальные затраты

| Наименование                              | Ед<br>из<br>ме<br>ре<br>щ<br>и<br>ц<br>н<br>и<br>а<br>я | Количество |           |           | Цена за ед.,<br>руб. |           |           | Затраты на материалы,<br>(З <sub>м</sub> ), руб. |        |        |
|---|---|------------|-----------|-----------|----------------------|-----------|-----------|--|--------|--------|
|   |   | Ис<br>п.1  | Ис<br>п.2 | Ис<br>п.3 | Ис<br>п.1            | Ис<br>п.2 | Ис<br>п.3 | Исп.1  | Исп.2  | Исп.3  |
| Система хранения данных DEPO Storage 3436 | Шт.   | 1          | 1         | 1         | 400522               | 400522    | 400522    | 400522   | 400522 | 400522 |
| Сервер Depo Storm 4355T2                  | Шт.   | 2          | 1         | 0         | 459591               | 459591    | 0         | 919182   | 459591 | 0      |
| Сервер Depo Storm 2350V2                  | Шт.   | 0          | 1         | 3         | 0                    | 109810    | 109810    | 0  | 109810 | 329430 |
| Итого                                     |   |            |           |           |                      |           |           | 1319704  | 969923 | 729952 |



#### 5.4.3.2. Расчет основной заработной платы исполнителей темы

Для расчета основной заработной платы используется формула 8:

$$C_{\text{осн/зп}} = \sum_{i=1}^n t_i \cdot C_{\text{зп}_i} \quad (8)$$

$C_{\text{осн/зп}} = 7 \cdot 1530,2 = 10711,4$ . Прочие вычисления выполнены по аналогии.

где  $n$  – количество работ;

$t_i$  – затраты труда на выполнение  $i$ -го вида работ, в днях;

$C_{\text{зп}_i}$  – средняя заработная плата работника, который выполняет  $i$ -ый вид работы, руб/день.

Средняя заработная плата в день рассчитывается по формуле 9:

$$C_{\text{зп}_i} = \frac{D \cdot K \cdot M_p}{F_0} \quad (9)$$

$C_{\text{зп}_i} = (10711,4 \cdot 1,3 \cdot 11,08) / 219 = 1530,2$ . Прочие вычисления выполнены по аналогии.

где  $D$  – должностной оклад работника в месяц;

$K$  – районный коэффициент ( $K=1,3$ );

$M_p$  – количество месяцев работы без отпуска в течение года;

$F_0$  – действительный годовой фонд рабочего времени работника, в днях.

При отпуске 28 дня  $M_p=11,08$ .

Результаты расчета действительного годового фонда проведены в таблице 15.

Таблица 15. Баланс рабочего времени

| Показатели рабочего времени                  | Руководитель | Студент1 | Студент2 |
|--|--------------|----------|----------|
| Календарное число дней                       | 365          | 365      | 365      |
| Количество нерабочих дней                    | 118          | 118      | 118      |
| - выходные дни                               |              |          |          |
| - праздничные дни                            |              |          |          |
| Потери рабочего времени                      | 28           | 28       | 28       |
| - отпуск                                     |              |          |          |
| - невыходы по болезни                        |              |          |          |
| Действительный годовой фонд рабочего времени | 219          | 219      | 219      |

Расчет затрат на основную заработную плату приведен в таблице 15. При этом затраты на оплату труда первого студента-дипломника определяются как оклад студента 1 ( $D = 6976,22$ ), второго – как оклад студента 2 с окладом в размере 6976,22 рублей. А оклад руководителя проекта составляет 23264,86. Коэффициент  $K$  - районный коэффициент равен 1,3, а  $M_p$  равно 11,08.

Таблица 16. Затраты на основную заработную плату

| Исполнители  | Среднедневная заработная плата $C_{зп}$ (руб.) |         |         | Трудоемкость ( $t_i$ ), чел-дни |         |         | Затраты на основную зарплату (руб.) |         |         |
|--------------|--|---------|---------|---------------------------------|---------|---------|-------------------------------------|---------|---------|
|              | Ис п. 1  | Ис п. 2 | Ис п. 3 | Ис п. 1                         | Ис п. 2 | Ис п. 3 | Ис п. 1                             | Ис п. 2 | Ис п. 3 |
| Руководитель | 1530,2   |         |         | 7                               | 9       | 8       | 10711,4                             | 13771,8 | 12241,6 |
| Б            |  |         |         |                                 |         |         |                                     |         |         |

|           |       |    |    |    |         |         |         |
|-----------|-------|----|----|----|---------|---------|---------|
| Студент 1 | 458,8 | 43 | 44 | 45 | 19728,4 | 20187,2 | 20646   |
| Студент 2 | 458,8 | 43 | 44 | 45 | 19728,4 | 20187,2 | 20646   |
| Итого     |       |    |    |    | 39456,8 | 54146,2 | 53533,6 |

### 5.4.3.3. Дополнительная заработная плата

Дополнительная заработная плата включает заработную плату за не отработанное рабочее время, но гарантированную действующим законодательством.

Расчет дополнительной заработной платы ведется по формуле 10:

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

$Z_{\text{доп}}=0,12 \cdot 10711,4=1285,4$ . Прочие вычисления выполнены по аналогии.

где  $k_{\text{доп}}$  – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12 – 0,15).  $k_{\text{доп}}$  равен 0,12. Результаты по расчетам дополнительной заработной платы сведены в таблицу 17.

Таблица 17. Затраты на дополнительную заработную плату

| Исполнители  | Основная зарплата( руб.) |         |         | Коэффициент дополнительной заработной платы ( $k_{\text{доп}}$ ) | Дополнительная зарплата( руб.) |        |        |
|--------------|--------------------------|---------|---------|--|--------------------------------|--------|--------|
|              | Ис п.1                   | Ис п.2  | Ис п.3  |  | Ис п.1                         | Ис п.2 | Ис п.3 |
| Руководитель | 10711,4                  | 13771,8 | 12241,6 | 0,12   | 1285,4                         | 1652,6 | 1469   |
| Студент 1    | 19728,4                  | 20187,2 | 20646   | -  | 2367,4                         | 2422,5 | 2477,5 |
| Студент 2    | 19728,4                  | 20187,2 | 20646   | -  | 2367,4                         | 2422,5 | 2477,5 |
| Итого        |                          |         |         |  | 4734,8                         | 6497,6 | 6424   |

### 5.4.3.4. Отчисления во внебюджетные фонды (страховые отчисления)

Величина отчислений во внебюджетные фонды определяется исходя из формулы 11:

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

$Z_{\text{внеб}} = 27,1 \cdot (10711,4 + 1285,4) = 15227,06$ . Прочие вычисления выполнены по аналогии.

где  $k_{\text{внеб}}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.). На 2014 г. В соответствии с Федеральным законом от 24.07.2009 №212-ФЗ установлен размер страховых взносов равный 30%. На основании пункта 1 ст.58 закона №212-ФЗ для учреждений, осуществляющих образовательную и научную деятельность в 2014 году водится пониженная ставка – 27,1%.

Отчисления во внебюджетные фонды представлены в таблице 18.

Таблица 18. Отчисления во внебюджетные фонды

| Исполнитель                                  | Основная заработная плата, руб. |            |            | Дополнительная заработная плата, руб. |            |            |
|--|---------------------------------|------------|------------|---------------------------------------|------------|------------|
|  | Ис .1<br>п                      | Ис .2<br>п | Ис .3<br>п | Ис .1<br>п                            | Ис .2<br>п | Ис .3<br>п |
| Руководитель проекта                         | 10711,4                         | 13771,8    | 12241,6    | 1285,4                                | 1652,6     | 1469       |
| Студент 1                                    | -                               | -          | -          | -                                     | -          | -          |
| Студент 2                                    | -                               | -          | -          | -                                     | -          | -          |
| Коэффициент отчислений во внебюджетные фонды | 27,1%                           |            |            |                                       |            |            |
| Итого  |                                 |            |            |                                       |            |            |
| Исполнение 1                                 | 15227,06                        |            |            |                                       |            |            |
| Исполнение 2                                 | 16434,47                        |            |            |                                       |            |            |
| Исполнение 3                                 | 16248,51                        |            |            |                                       |            |            |

#### 5.4.3.5. Формирование бюджета затрат научно-исследовательского проекта

Рассчитанная величина затрат научно-исследовательской работы (темы) является основой для формирования бюджета затрат проекта, который при формировании договора с заказчиком защищается научной организацией в качестве нижнего предела затрат на разработку научно-технической продукции. Определение бюджета затрат на научно-исследовательский проект по каждому варианту исполнения приведен в таблице 19.

Таблица 19. Расчет бюджета затрат НИИ

| Наименование статьи   | Сумма, руб. |          |          |
|---|-------------|----------|----------|
|   | Исп.1       | Исп.2    | Исп.3    |
| 1. Материальные затраты НИИ                                     | 1509804     | 1149923  | 915552   |
| 2. Амортизационные отчисления                                   | 10315,8     | 7662,6   | 6189,3   |
| 3. Затраты по основной заработной плате исполнителей темы       | 39456,8     | 54146,2  | 53533,6  |
| 4. Затраты по дополнительной заработной плате исполнителей темы | 4734,8      | 6497,6   | 6424     |
| 5. Отчисления во внебюджетные фонды                             | 15227,06    | 16434,47 | 16248,51 |
| 6. Бюджет затрат НИИ  | 1579368     | 1233286  | 996755   |

**Вывод:** Основываясь на данных, полученных в пунктах 3.4.1 – 3.4.6, был рассчитан бюджет затрат научно-исследовательской работы для трех исполнителей. Наиболее низким по себестоимости оказался проект третьего исполнителя, затраты на его полную реализацию составляют **996755 рублей**.

## Вывод раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение».

С каждым годом требования к аппаратно-программному комплексу в отрасли нефтедобычи все больше повышаются. Возникает необходимость разрабатывать более сложные месторождения, где требуется большая точность и осторожность. Ошибки, аварии при бурении могут повлечь за собой потери, исчисляемые в огромных суммах. Все это способствует поиску решений, увеличивающих контроль процесса бурения. Идеальная схема – данные передаются с множества датчиков сразу же прямо в центр моделирования месторождений, где создается объемная модель пластов, прогнозируются возможные аварийные ситуации, проблемы, происходит мгновенная корректировка процесса.

На практике все выглядит иначе. Данные не передаются в режиме real-time, месторождение моделируется в «плоском» варианте. Для того, чтобы заменить всю систему получения, обработки и передачи данных потребуются огромные суммы, не каждая компания имеет такую возможность. Переход на новую схему должен иметь место, однако этот переход должен быть постепенным.

Именно для этой цели создается «агрегатор» – аппаратно-программный решение, позволяющее принимать данные с ГТИ в режиме реального времени, конвертировать их в новый формат WITSML 1.3.1 или 1.4.1, сохраняя при этом в базе данных старые отчеты на случай неполадок при конвертации или передачи данных на WITSML сервер. При этом новая схема передачи данных будет работать параллельно со старой, что позволит, не нарушая регламент, внедрить новые технологии, протестировать их на предмет ошибок, не вызывая сбоев в отлаженной, хотя уже и устаревшей, системе.

В России пока нет аналогов данному устройству. Лишь за рубежом существует единственное подобное решение. Компания PetroDAQ разработала

продукт PetroDAS, который поддерживает множество типов датчиков, различных протоколов. «Агрегатор» PetroDAS реализует режим реального времени, буферизацию данных, а также конвертирует отчеты в формат WITSML.

Данное устройство за рубежом стоит 1232000 рублей. Агрегатор хоть и более функционален, по сравнению с нашей разработкой, однако его стоимость превышает стоимость создания отечественного агрегатора. Плюс ко всему, с учетом эмбарго, получить импортное устройство достаточно трудно. Таким образом, данное устройство будет высоко востребовано на отечественном рынке буровой отрасли, затраты на его полную реализацию будут покрыты.

## **Раздел 6. Социальная ответственность**

### **6.1. Введение**

При работе над проектной и научно-исследовательской деятельностью значительную роль играет обеспечение безопасности охраны труда и окружающей среды. Понятие «социальная ответственность» включает в себя: охрану здоровья работников, обеспечение безопасных условий работы, защита от профессиональных заболеваний и производственного травматизма.

Выше перечисленные аспекты регламентируются в соответствии с международным стандартом ICCSR26000:2011 «Социальная ответственность организации». Целью данного стандарта является принятие проектных решений, исключающих несчастные случаи на производстве и снижение негативных воздействий на окружающую среду. На основе стандарта написана данная часть работы.

Раздел, посвященный социальной ответственности организации, является обязательной частью дипломной работы и включает в себя следующие составляющие: техногенная безопасность, региональная безопасность, организационные мероприятия обеспечения безопасности, особенности законодательного регулирования проектных решений и безопасность в чрезвычайных ситуациях.

Научно-исследовательская работа представляет собой разработку проекта национального стандарта передачи данных по сети и предполагает непосредственную работу с персональным компьютером. Поэтому определение оптимальных условий труда является неотъемлемым критерием безопасности.



## 6.2. Производственная безопасность

### 6.2.1. Недостаточная освещенность рабочей зоны

Освещение в недостаточной степени может привести к напряжению зрения, ослаблению внимания и наступлению преждевременной утомленности. Слепление, резь в глазах и раздражение могут быть вызваны чрезмерно ярким освещением. Неверное направление света на месте труда может создать резкие тени или блики, а также дезориентировать работающего. Перечисленные причины могут привести к профзаболеваниям.

В аудитории № 204 кибернетического центра, где осуществлялось выполнение ВКР, применяется искусственное освещение, в качестве источников которого используются люминесцентные лампы типа ЛД-20. Коэффициент пульсации ( $K_{\text{п}}$  – колебания светового потока, падающего на единицу поверхности во времени) в аудитории составляет около 40%. Согласно требованиям, СанПиН 2.2.2/2.4.1340-03, в помещениях, оборудованных компьютерами коэффициент пульсации должен быть не более 5%.

Поэтому в рамках данной работы проводится расчет искусственного освещения для светодиодных светильников взамен установленным люминесцентным.

Площадь помещения составляет  $56\text{ м}^2$  (при длине 8 м и ширине 7 м). Светильников в аудитории размещены в четыре ряда, по шесть в каждом ряду.

Каждый светильник позволяет установить четыре светодиодных лампы типа ССОН СД В-О-01-110-30-001-IP20-УХЛ4 (мощность 30 Вт, световой поток 2000 лм). Общее количество ламп в помещении составит 75.

Светильники размещены таким образом, чтобы равномерно освещать помещение и рабочие места (рисунок 26).

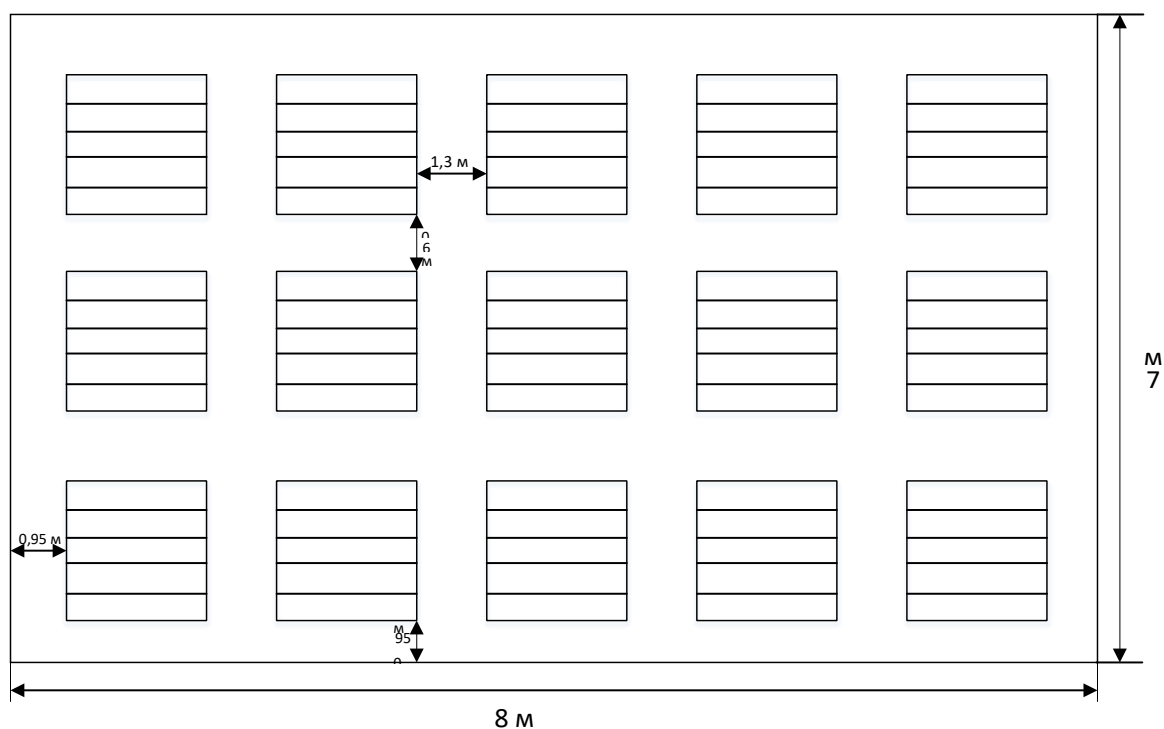


Рис. 26 - Схема размещения светодиодных светильников

Согласно справочной информации, коэффициент естественного освещения (КЕО) должен быть не ниже 3% при выполнении работы с высокой зрительной точностью при наименьшем размере объекта различения 0,3 – 0,5 мм, и не менее 2,4% при выполнении зрительной работы средней точности (наименьший размер объекта различения 0,5...1,0 мм).

Установлено, что для помещений, в которых установлены компьютеры, требуется освещенность не менее 300 лк при выполнении зрительной работы высокой точности и не менее 200 лк, при выполнении работы средней точности.

Для проведения расчетов выбран метод коэффициента использования светового потока.

Требуемая освещенность рассчитывается по следующей формуле:

$$E = \frac{F \cdot N \cdot \eta}{S \cdot z \cdot k} \quad (1)$$

Где:

$E_n$  – нормированная минимальная освещенность, лк;

$F = 2000$  лм – световой поток для ССОН СД В-О-01-110-30-001-IP20-УХЛ4;

$S$  – площадь помещения, м<sup>2</sup>;

$z$  – коэффициент неравномерности освещения (для светодиодных светильников  $z = 1$ );

$k_3$  – коэффициент запаса, зависящий от вида технологического процесса и типа применяемых источников света ( $k = 1,5$  – для помещений с малым выделением пыли);  $N$  – количество ламп в помещении;  $\eta$  – коэффициент использования светового потока.

Коэффициент использования светового потока выбираем в зависимости от типа светильников, размеров помещения, определяемых индексом помещения, коэффициентов отражения стен и потолка помещения. Коэффициент отражения побеленного потолка  $\rho_{п} = 70\%$ ; коэффициент отражения от стен, окрашенных в светлую окраску (белый цвет)  $\rho_{ст} = 50\%$ ;

Расчет индекса помещения произведем по формуле

$$i = \frac{l \cdot d}{h \cdot (l + d)}, \quad (2)$$

Где:

$h$  – высота подвеса светильников, м ( $h = 4$  м);

$l$  – длина помещения, м ( $l = 8$  м);

$d$  – ширина помещения, м ( $d = 7$  м).

Тогда индекс помещения равен  $i = 1$ . Согласно табличным данным, находим коэффициент использования светового потока при коэффициентах отражения потолка  $\alpha_{\text{п}}=0,7$  и стен  $\alpha_{\text{с}} =0,5$ ,  $\eta = 0,3$ , который показывает какая часть светового потока ламп попадает на рабочую поверхность.

Проведем расчет освещенности: теперь определяем нормированную освещенность:

$$E_{\text{н}} = (2000*75*0,3) / (56*1,5*1) = 535 \text{ лк.}$$

В результате нормированная минимальная освещенность составила 535 лк, что удовлетворяет санитарным нормам для помещения, где выполнялась ВКР.

### **6.2.2. Отклонение показателей микроклимата**

Компьютеры могут привести к повышению температуры и снижению относительной влажности в помещении. В таких помещениях должны соблюдаться определенные параметры микроклимата. В санитарных нормах - СанПиН 2.2.4.548 установлены величины параметров микроклимата, создающие комфортные условия. Отклонения параметров микроклимата от установленных норм способствуют в первую очередь нарушению физиологической функции человека сохранять тепловой баланс организма, что может повлиять на состояние здоровья и общую производительность труда. В обычных климатических условиях теплоотдача осуществляется в основном за счет излучения (примерно 45 % всей удаляемой организмом теплоты), конвекции (30 %) и испарения (25 %).

Аудитория №204 кибернетического центра оснащена компьютерами в количестве 10 штук, которые являются источниками тепла и могут вызвать существенное повышение температуры и уменьшение относительной влажности в помещении.

Нормы на параметры микроклимата в помещении зависят от времени года, характера трудового процесса и характера помещения.

Работа оператора ЭВМ относится к категории работ Ia, в которую входят работы с интенсивностью энергозатрат до 139Вт, производимые сидя и сопровождающиеся незначительным физическим напряжением. Оптимальные параметры микроклимата для этой категории работ представлены в таблице 20.

Таблица 20. Параметры микроклимата для помещений, где установлены компьютеры

| Период   | Параметр микроклимата               | Величина  |
|----------|-------------------------------------|-----------|
| Холодный | Температура воздуха в помещении, °С | 22...24   |
|          | Относительная влажность, %          | 44...60   |
|          | Скорость движения воздуха, м/с      | До 0,1    |
| Теплый   | Температура воздуха в помещении, °С | 23...25   |
|          | Относительная влажность, %          | 40...60   |
|          | Скорость движения воздуха, м/с      | 0,1...0,2 |

Обеспечить комфортные условия труда можно как за счет организационных методов (рациональная организация проведения работ в зависимости от времени года и суток, чередование труда и отдыха), так и за счет внедрения технических средств (вентиляция, кондиционирование воздуха, отопительная система). Поэтому в работе проведен расчет требуемого воздухообмена в аудитории.

Расчет требуемого воздухообмена ( $L$  м<sup>3</sup>/ч), определяется по формуле:

$$L = \frac{G \cdot 1000}{x_v - x_n}, \quad (3)$$

где:

$L$ , м<sup>3</sup>/ч – требуемый воздухообмен;

$G$ , г/ч – количество вредных веществ, выделяющихся в воздух;

$x_v$ , мг/м<sup>3</sup> – предельно допустимая концентрация вредности в воздухе рабочей зоны помещения;

$X_n$ , мг/м<sup>3</sup> – максимально возможная концентрация той же вредности в воздухе населенных мест.

По справочным данным, определяем количество углекислого газа, выделяемое одним человеком. Если в помещении работают 10 человек,  $g = 23$  л/ч, допустимую концентрацию CO<sub>2</sub>,  $X_v = 1$  л/м<sup>3</sup>. Содержание CO<sub>2</sub> в наружном воздухе для больших городов принимаем:  $X_n = 0,5$  л/м<sup>3</sup>, то потребный воздухообмен составит:

$$L = 23 \cdot 4 / (1 - 0.5) = 184 \text{ м}^3/\text{ч}.$$

Следовательно, потребный воздухообмен для 204 кибернетического центра общей площадью рабочего помещения 56 м<sup>2</sup> и объемом 224 м<sup>3</sup>, где на каждого работающего приходится в среднем 5,6 м<sup>2</sup> общей площади и 22,4 м<sup>3</sup> объема, составляет 184 м<sup>3</sup>/ч. Соответствует принятым нормам.

### 6.2.3. Уровень шума на рабочем месте

Работающие в условиях длительного шумового воздействия испытывают раздражительность, головные боли, головокружение, снижение памяти, повышенную утомляемость, понижение аппетита, боли в ушах и т. д. Все это снижает работоспособность человека и его производительность, качество и безопасность труда.

В аудитории №204 кибернетического центра источниками шума служат процессоры ЭВМ. Согласно установленным нормам, уровень шума на рабочем месте операторов ЭВМ не должен превышать 50 дБ. С целью снижения шумового воздействия, потолок аудитории выполнен из звукопоглощающих панелей. Уровень шума в аудитории №204 кибернетического центра составляет около 40 дБ, что не превышает допустимых норм.

#### 6.2.4. Уровень электромагнитных излучений

Максимальный уровень рентгеновского излучения на рабочем месте оператора компьютера обычно не превышает 10 мкбэр/ч, а интенсивность ультрафиолетового и инфракрасного излучений от экрана монитора лежит в пределах 10...100 мВт/м<sup>2</sup>.

Для снижения воздействия этих видов излучения рекомендуется применять мониторы с пониженным уровнем излучения (MPR-II, TCO-92, TCO-99), устанавливать защитные экраны, а также соблюдать регламентированные режимы труда и отдыха.

Таблица 21. Допустимые значения параметров неионизирующих электромагнитных излучений (в соответствии с СанПиН 2.2.2.542-96)

| Наименование параметра  | Допустимые |
|---|------------|
| Напряженность электрической составляющей электромагнитного поля на расстоянии 50см от поверхности видеомонитора | 10 В/м     |
| Напряженность магнитной составляющей электромагнитного поля на расстоянии 50см от поверхности видеомонитора     | 0,3А/м     |
| Напряженность электростатического поля не должна превышать:<br>для взрослых пользователей                       | 20 кВ/м    |

Выполнение выпускной квалификационной работы проводилось на современном компьютере, монитор которого удовлетворяет нормативным требованиям по напряженности электромагнитного поля и другим показателям.

#### 6.2.5. Электробезопасность

Электропитание аудитории № 204 кибернетического центра ТПУ осуществляется от силового распределительного щита однофазного переменного тока с действующим значением напряжения 220В. Таким образом,

в соответствии с правилами устройства электроустановок все электроприборы, используемые в аудитории, относятся к низковольтным с напряжением питания до 1000 В. В помещении большая часть электрической проводки является скрытой. Поражение электрическим током возможно только контакте человека с неизолированным проводником.

Следует отметить, что кабель имеет двойную изоляцию, что существенно снижает риск поражения. Однако не следует исключать также опасность поражения и от токоведущих частей компьютера в случае их пробоя и нарушении изоляции.

Для обеспечения электробезопасности в аудитории должны быть проведены следующие мероприятия:

1. для защиты от токов короткого замыкания необходимо предусмотреть наличие быстродействующих устройств защиты; электрическая сеть должна иметь защиту от токов короткого замыкания, обеспечивающую по возможности наименьшее время отключения и требования селективности; в качестве аппаратов защиты должны применяться автоматические выключатели или предохранители;

2. для защиты от напряжения прикосновения все токоведущие части должны быть изолированы; запрещается использовать кабели и провода с поврежденной или потерявшей защитные свойства изоляцией; неизолированные токоведущие части должны быть оборудованы защитными ограждениями или расположены в недоступном для прикосновения месте; запрещается пользоваться поврежденными розетками, распределительными коробками, рубильниками и другими электроустановочными приборами; устройство и эксплуатация временных электросетей не допускается;

3. для защиты от поражения электрическим током путем возникновения потенциала на проводящих корпусах электроприборов необходимо наличие защитного заземления; зануление, согласно ПУЭ сопротивление заземляющего устройства в любое время года должно быть не более 4 Ом, при этом сечение



заземляющей жилы должно быть не менее 4 мм<sup>2</sup> для медных проводников, не менее 6 мм<sup>2</sup> – для алюминиевых и не менее 20 мм<sup>2</sup> – для стальных.

Для предотвращения электротравматизма большое значение имеет правильная организация обслуживания аудитории, проведение ремонтных, монтажных и профилактических работ.

Ремонт, разборку и сборку, наладку электротехнологического оборудования может выполнять только подготовленный персонал, имеющий необходимую для данных работ группу допуска по электробезопасности.

### **6.3. Экологическая безопасность**

В настоящее время проблема экологической безопасности является приоритетной. Это стало поводом для принятия жестких законов, ограничивающих обычную утилизацию компьютерной техники. В большей мере это обуславливается тем, что в производстве такой техники используется множество различных материалов, которые способны нанести непоправимый вред окружающей среде и, соответственно, здоровью человека. Утилизация компьютерного оборудования является достаточно сложной. Непосредственная переработка большей части компонентов включает в себя их сортировку, последующую гомогенизацию и отправку для повторного использования, т.е. с предварительным помолом или переплавкой.

Люминесцентные лампы представляют собой «чрезвычайно опасные» виды отходов. Все люминесцентные лампы содержат от 3 до 5 мг ртути. Так как в производстве энергосберегающих ламп используется ртуть (в небольших дозах, в виде амальгамы), их утилизация не должна представлять простой выброс в контейнер с мусором. Лампы должны утилизироваться коммунальными службами, занимающимися вывозом специальных отходов. Транспортировка ламп на полигоны складирования должна выполняться организациями, которые специализируются на утилизации опасных отходов.

#### **6.4. Правовые и организационные вопросы обеспечения безопасности**

Охрана труда и техника безопасности это – система сохранения жизни и здоровья работников в процессе трудовой деятельности, включающая в себя правовые, социально-экономические, организационно-технические, санитарно-гигиенические, лечебно-профилактические, реабилитационные и иные мероприятия (статья № 1 Федерального закона «Об основах охраны труда в Российской Федерации», 17.07.1999 г. №181-ФЗ), образующие механизм реализации конституционного права граждан на труд (ст. 37 Конституции РФ) в условиях, отвечающих требованиям безопасности и гигиены. (Это право закреплено также в ст. 7 международного пакта об экономических, социальных и культурных правах).

Статья 37 Конституции Российской Федерации гарантирует свободу труда, а также право на труд, в условиях, отвечающих требованиям безопасности и гигиены.

Федеральным органом исполнительной власти, осуществляющим функции по контролю и надзору в сфере труда, является Федеральная служба по труду и занятости Министерства здравоохранения и социального развития Правительства Российской Федерации.

Основными задачами трудового законодательства являются создание необходимых правовых условий для достижения оптимального согласования интересов сторон трудовых отношений, интересов государства, а также правовое регулирование трудовых отношений и иных непосредственно связанных с ними отношений.

Помимо обеспечения безопасных условий труда гражданина, законодательство налагает ответственность на каждого за состояние окружающей природной среды. Так Конституция Российской Федерации статьей 58 обязывает каждого «сохранять природу и окружающую среду, бережно относиться к природным богатствам».

## 6.5. Безопасность в чрезвычайных ситуациях

Пожарная безопасность — состояние защищённости личности, имущества, общества и государства от пожаров. Пожарная безопасность обеспечивается системой пожарной защиты и системой предотвращения пожара.

*Опасный фактор пожара (ОФП):*

- пламя и искры;
- тепловой поток;
- повышенная температура окружающей среды;
- повышенная концентрация токсичных продуктов горения и термического разложения;
- пониженная концентрация кислорода;
- снижение видимости в дыму.

Действия при пожаре в здании:

- При наличии телефона, «112» или «01» сообщить о пожаре и своем местоположении;
- Не входить в места с высокой концентрацией дыма и видимостью менее, чем 10 метров.
- Покинуть помещение, используя запасные и основные пути эвакуации;
- Попутно отключить электроэнергию;
- Передвигаться к выходу на четвереньках, при этом закрывая рот и нос подручными средствами защиты;
- Плотно закрыть дверь при выходе;

Если дым и пламя в соседних помещениях не позволяет выйти наружу:

- Стараться не поддаваться панике;
- Проверить возможности спуститься по пожарной лестнице или выйти на крышу;

- При отсутствии возможности эвакуироваться для защиты от дыма и тепла необходимо как можно надёжней загерметизировать своё помещение:
- Плотно закрыть двери, окна и форточки, заткнуть щели изнутри, используя при этом любую, желательно мокрую, ткань;
- При наличии воды, постоянно смачивать двери и пол.

#### 6.6. Организационные мероприятия обеспечения безопасности

В соответствии с государственными стандартами и правовыми нормами обеспечения безопасности предусмотрена рациональная организация труда в течение смены, которая предусматривает:

- длительность рабочей смены не более 8 часов;
- установление двух регламентируемых перерывов (не менее 20 минут после 1-2 часов работы, не менее 30 минут после 2 часов работы);
- обеденный перерыв не менее 40 минут.

Обязательно предусмотрен предварительный медосмотр при приеме на работу и периодические медосмотры.

Каждый сотрудник обязан пройти инструктаж по технике безопасности перед приемом на работу и в дальнейшем, должен быть пройден инструктаж по электробезопасности и охране труда. Предприятие обеспечивает рабочий персонал всеми необходимыми средствами индивидуальной защиты.

Оплата труда, социальные пособия, дополнительные выплаты устанавливаются в соответствии со степенью вредности и опасности выполняемых обязанностей.

Согласно требованиям СанПиН 2.2.2/2.4.1340–03, расстояние между рабочими столами с видеомониторами, равно 2 м, а расстояние между боковыми поверхностями видеомониторов примерно 1,2 м. Площадь на одно рабочее место пользователей персонального компьютера с монитором на базе плоских дискретных экранов (жидкокристаллические, плазменные) - 4,5 м<sup>2</sup>.

## **6.7. Выводы раздела «Социальная ответственность»**

В данном разделе ВКР были рассмотрены вредные и опасные факторы на рабочем месте оператора ЭВМ в аудитории № 204 кибернетического центра Томского политехнического университета. Рассмотрены требования по технике безопасности, электробезопасности, пожаробезопасности, проведены расчеты требуемой освещенности и воздухообмена рассматриваемого помещения.

Также были выявлены возможные чрезвычайные ситуации и приведен план действий для наиболее вероятной ЧС в данном помещении – возникновение пожара.

В рамках настоящего раздела также изучены правовые основы безопасности труда.

## Заключение

Результатом исследовательской работы является изучение современного стандарта передачи WITSML версии 1.3.1 и 1.4.1. Изучен мировой опыт создания и разработки стандартов передачи данных и их использования в отечественной нефтедобычи, технология применения с учётом требований отечественных компаний, создана блок - схема АПК. Была разработана концепция использования устройства, которая, не нарушая устоявшихся регламентов компании и не требуя замены оборудования, осуществляет агрегирование и конвертирование данных со станций ГТИ, передачу их на сервер WITSM версии 1.3.1 и 1.4.1 для использования моделью месторождения в режиме реального времени.

В рамках дипломной работы было создано ряд программных модулей, позволяющее принимать, хранить и конвертировать данные о бурении в режиме реального времени из стандартов LAS и WITS в современный стандарт передачи данных WITSM версии 1.3.1 и 1.4.1, спроектирована база данных для хранения файлов в исходных форматах, разработан блок конвертации, реализующий преобразование данных, а также спроектирован веб интерфейс АПК «агрегатор».

## Список использованных источников:

1. Сэм Улиарис, «Что такое экономические модели?», Финансы и развитие, июнь 2011.
2. Н.М. Абдикеев, «Интернет технологии в экономике знаний: Учебник»- М.:ИНФРА-М, 2012.
3. Ю.М. Осипова, Е.Е. Смирнова, «Основы предпринимательского дела: учебник» 2-е издание, Издательство БЭК, 1996.
4. Пикеринг, Д., Грёвик, Л., Франссенс, Д., Дикс, Н., Донигер, А., Шей, Д.: “WITSML достигает совершеннолетия в глобальной индустрии бурения и освоения”. США, 2009.
5. [Электронный ресурс] - Оптимизация добычи и интеллектуальные скважины, <http://www.weatherford.ru/ru/service/production/53>.
6. [Электронный ресурс] - Оптоволоконные датчики в бурении, <http://stud24.ru/geology/optovolokonnye-datchiki-vburenii/476130-1811034.html>
7. [Электронный ресурс] - портал Energistics, <http://www.energistics.org/>
8. [Электронный ресурс] - портал XML, <http://www.codenet.ru/webmast/xml/part1.php>
9. [Электронный ресурс] - статья XML схемы, [https://ru.wikipedia.org/wiki/XML\\_Schema](https://ru.wikipedia.org/wiki/XML_Schema)
10. [Электронный ресурс] - Промышленные стандарты обмена данными по бурению и заканчиванию скважин, <http://www.energistics.org/Assets/witsmlflyerrussian.pdf>
11. Майо Д. Самоучитель Microsoft Visual Studio 2010, С.: «БХВ-Петербург», 2010. — С. 464.
12. Дж. Рихтер, Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд.
13. Александр Климов, советы программистам, «БХВ-Петербург», 2008. - С. 544.

14. Beeming G. Team Foundation Server 2013 Customization, Packt Publishing, 2014. — С. 102.
15. Мейер Дж.Д., Тейлор Джейсон и др. Командная разработка с использованием Visual Studio Team Foundation Server: Справочник, 2007. - С. 575
16. Виктор Гольцман , SQL и реляционная теория.
17. [Электронный ресурс] - Microsoft SQL Server, [https://ru.wikipedia.org/wiki/Microsoft\\_SQL\\_Server/](https://ru.wikipedia.org/wiki/Microsoft_SQL_Server/)
18. [Электронный ресурс] – Rogtec, <http://rogtecmagazine.com/>
19. [Электронный ресурс] Положение о ВКР ТПУ, <http://tpu.ru/f/2039/vkr.pdf#2>
20. PETRODAS USER GUIDE, PetroDAQ Inc. Publishing, 2013. — С. 41.



## Приложения

### Приложение 1. Программный код конвертера

#### Конвертер Las to Witsml

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Text.RegularExpressions;
using System.Linq;
using System.Globalization;

namespace Las2witsmlLIB
{
    /// <summary>
    ///     Класс для описания сущности las-файла
    /// </summary>
    public class Las
    {
        public Las(StreamReader inputStream)
        {
            InputStream = inputStream;
            nextLine = null;

            LogCurveInfos = new List<LogCurveInfo>();
            CurveValues = null;
            MeasuredDepthUnit = null;
            StartMeasuredDepthIndex = 0;
            StopMeasuredDepthIndex = 0;
            StartDateTimeIndex = new DateTime(1970, 1, 1, 0, 0, 0);
            StopDateTimeIndex = new DateTime(1970, 1, 1, 0, 0, 0);
            StepIncrement = 0;
            NullValue = 0;
            ServiceCompany = null;
            ElevationKellyBushing = null;
            LogMeasuredFrom = null;
            PermanentDatum = null;
            AbovePermanentDatum = 0;
            ElevationPermanentDatum = 0;
            RunNumber = 0;
        }

        public List<LogCurveInfo> LogCurveInfos; // Лист параметров в секции А - берется по
        всем выделенным параметрам кривых в секции С
        public string CurveValues { get; set; }
        public string MeasuredDepthUnit { get; set; }
        public double StartMeasuredDepthIndex { get; set; }
        public double StopMeasuredDepthIndex { get; set; }
        public DateTime StartDateTimeIndex { get; set; }
        public DateTime StopDateTimeIndex { get; set; }
        public double StepIncrement { get; set; }
        public double NullValue { get; set; }
        public string ServiceCompany { get; set; }
        public string ElevationKellyBushing { get; set; }
        public string LogMeasuredFrom { get; set; }
        public string PermanentDatum { get; set; }
        public double AbovePermanentDatum { get; set; }
        public double ElevationPermanentDatum { get; set; }
        public double RunNumber { get; set; }
    }
}
```

```

public string ElevationUnit { get; set; }

private StreamReader InputStream { get; set; }
private string nextLine { get; set; }
private string Line { get; set; }

// Главный метод данного класса, где будет считываться весь файл по секциям
//
public void Process()
{
    while (!InputStream.EndOfStream)
    {
        Line = ReadNextLine();
        if (String.IsNullOrEmpty(Line) || Line[0]=='#')
        {
            Line = ReadNextLine();
        }
        var section = '0';
        if (Line != null && Line[0] == '~')
        {
            section = char.ToLower(Line[1]);
        }

        switch (section)
        {
            case 'w':
                nextLine = ReadNextLine();
                SectionWell();
                break;
            case 'c':
                nextLine = ReadNextLine();
                SectionCurve();
                break;
            case 'p':
                nextLine = ReadNextLine();
                SectionParameter();
                break;
        }

        if (section == 'a')
        {
            break;
        }
    }
}

public void EachDataLine(Witsml witsml)
{
    while(!InputStream.EndOfStream)
    {
        try
        {
            Line = ReadNextLine();
            if (String.IsNullOrEmpty(Line) || Line[0] == '#')
            {
                Line = ReadNextLine();
            }

            string dataString = "";
            Line = Line.Replace("\t", " ");
            var dataLine = Line.Split(' ').Where(x=> x!= "").ToArray();
            if (String.IsNullOrEmpty(this.MeasuredDepthUnit))
            {

```

```

        dataLine[0] = witsml.MakeDateFromDouble(dataLine[0], this);
    }

    foreach(var str in dataLine)
    {
        if (str == dataLine.Last())
        {
            dataString += str;
        }
        else
        {
            dataString += str + ',';
        }
    }

    witsml.xmlWriter.WriteElementString("data", dataString);
}
catch(Exception ee)
{}
}
}

// Переход к следующей строке файла
//
private string ReadNextLine()
{
    Line = "";
    if (!string.IsNullOrEmpty(nextLine))
    {
        Line = nextLine;
        nextLine = null;
    }
    else
    {
        Line = InputStream.ReadLine();
    }
    return Line;
}

// Переход к предыдущей строке файла
//
private void PreviousLine(string line)
{
    nextLine = line;
}

// Проверка - является ли измеряемый промежуток отрезком глубины
//
public string IsDepthUnit(string unit)
{
    if ((unit.ToLower() == "ft") ||
        (unit.ToLower() == "f") ||
        (unit.ToLower() == "fm") ||
        (unit.ToLower() == "cm") ||
        (unit.ToLower() == "in") ||
        (unit.ToLower() == "m"))
    {
        return unit;
    }
    else return null;
}

// Если измеряемый промежуток не глубина, то парсим дату

```

```

//
private static DateTime ParseDateTime2(string data, string info, string unit = "s")
{
    DateTime dateTime;
    if (DateTime.TryParse(data, DateTimeFormatInfo.InvariantInfo,
DateTimeStyles.AssumeUniversal, out dateTime))
    {
        return dateTime;
    }

    var offset = Parsing.ParseDouble(data);
    if (double.TryParse(data, NumberStyles.Any, NumberFormatInfo.InvariantInfo, out
offset))
    {
        var secondsSince1970 = unit?.Equals("date",
StringComparison.OrdinalIgnoreCase) == true;
        if (!secondsSince1970)
        {
            var date = Parsing.ParseDateTime(info);
            //return date.AddSeconds(offset);
            return date;
        }
        return new DateTime(1970, 1, 1, 0, 0, 0, 0,
DateTimeKind.Utc).AddSeconds(offset);
    }
    throw new ApplicationException($"Значение '{data}' не возможно представить как
дату/время.");
}

// Обработка секции W
//
private void SectionWell()
{
    nextLine = null;
    while (Line[0] != '~')
    {
        try {
            var regex = new Regex(@"^([\^~][^\.]*)\.([\^s]*)(.*?):(.*)$",
RegexOptions.IgnoreCase);
            var m = regex.Match(Line).Groups.Cast<Group>().Skip(1).Take(4).Select(x
=> x.Value.Trim()).ToList();
            var mnemonic = m[0];
            var unit = m[1];
            var data = m[2];
            var info = m[3];

            switch (mnemonic)
            {
                case "STRT":
                    if (IsDepthUnit(unit) != null)
                    {
                        MeasuredDepthUnit = unit;
                        StartMeasuredDepthIndex = Parsing.ParseDouble(data);
                    }
                    else
                    {
                        StartDateTimeIndex = ParseDateTime2(data, info, unit); //
тут ошибка пока-что = нужно парсить дату или секунды - хз пока
                    }
                    break;
                case "STOP":
                    if (IsDepthUnit(unit) != null)
                    {

```

```

        MeasuredDepthUnit = unit;
        StopMeasuredDepthIndex = Parsing.ParseDouble(data);
    }
    else
    {
        StopDateTimeIndex = ParseDateTime2(data, info, unit); // тут
        // ошибка пока-что = нужно парсить дату или секунды - хз пока
    }
    break;
case "STEP":
    StepIncrement = Parsing.ParseDouble(data);
    break;
case "NULL":
    NullValue = Parsing.ParseDouble(data);
    break;
case "SRVC":
    ServiceCompany = data;
    break;
}
}
catch(Exception ee) { }

Line = ReadNextLine();
if (String.IsNullOrEmpty(Line))
{
    Line = ReadNextLine();
}
}
PreviousLine(Line);
}

// Обработка секции C
//
private void SectionCurve()
{
    nextLine = null;
    while (Line[0] != '~')
    {
        try {
            var regex = new Regex(@"^([\^~][^\.\.]+)\.([\^: \t]*)\s*([\^:]*):(.*)$",
                RegexOptions.IgnoreCase);
            var m = regex.Match(Line).Groups.Cast<Group>().Skip(1).Take(4).Select(x
=> x.Value.Trim()).ToList();
            var mnemonic = m[0];
            var unit = m[1];
            var _ = m[2];
            var info = m[3];

            LogCurveInfos.Add(new LogCurveInfo(mnemonic, unit.ToLower(), info));
        }
        catch(Exception ee) { }

        Line = ReadNextLine();
        if (String.IsNullOrEmpty(Line) || Line[0] == '#')
        {
            Line = ReadNextLine();
        }
    }
    PreviousLine(Line);
}

// Обработка секции P
//

```

```

private void SectionParameter()
{
    // У меня такое чувство, что все эти параметры разработчики взяли откуда то
    непонятно, и их не надо сюда писать
    // но для примера пусть будут
    while (Line[0] != '~')
    {
        try {
            var regex = new Regex(@"^([\^~][^\.]*)\.([\^s]*)(.*):(.*)$",
RegexOptions.IgnoreCase);
            var m = regex.Match(Line).Groups.Cast<Group>().Skip(1).Take(4).Select(x
=> x.Value.Trim()).ToList();
            var mnemonic = m[0];
            var unit = m[1];
            var data = m[2];

            switch (mnemonic)
            {
                case "RUN":
                    RunNumber = Convert.ToDouble(data);
                    break;
                case "PDAT":
                    PermanentDatum = data;
                    break;
                case "EPD":
                    ElevationPermanentDatum = Convert.ToDouble(data);
                    ElevationUnit = unit;
                    break;
                case "EGL":
                    ElevationPermanentDatum = Convert.ToDouble(data);
                    ElevationUnit = unit;
                    break;
                case "LMF":
                    LogMeasuredFrom = data;
                    break;
                case "APD":
                    AbovePermanentDatum = Convert.ToDouble(data);
                    ElevationUnit = unit;
                    break;
                case "EKB":
                    ElevationKellyBushing = data;
                    ElevationUnit = unit;
                    break;
            }
        }
        catch(Exception ee) { }

        Line = ReadNextLine();
        if (String.IsNullOrEmpty(Line) || Line[0] == '#')
        {
            Line = ReadNextLine();
        }
    }

    PreviousLine(Line);
}

}

}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

```

```

using System.Threading.Tasks;

namespace Las2witsmLIB
{
    /// <summary>
    /// Класс для описания сущности лога кривой
    /// </summary>
    public class LogCurveInfo
    {
        public string Mnemonic { get; set; }
        public string Unit { get; set; }
        public string Description { get; set; }

        public LogCurveInfo(){}

        public LogCurveInfo(string mnemonic, string unit, string description)
        {
            this.Mnemonic = mnemonic;
            this.Unit = unit;
            this.Description = description;
        }
    }
}
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;

namespace Las2witsmLIB
{
    class Parsing
    {
        public static double ParseDouble(string data)
        {
            return double.Parse(data, NumberStyles.Any, NumberFormatInfo.InvariantInfo);
        }

        public static DateTime ParseDateTime(string data)
        {
            var dateTimePattern = new Regex(@"[\d:\/T\-\.\s]+");
            var dateTimeSubstring = dateTimePattern.Match(data);
            if (!dateTimeSubstring.Success)
            {
                throw new ApplicationException("Либо кривая дата, либо кривой регексп");
            }

            var dateTime = DateTime.Parse(dateTimeSubstring.Value,
            DateTimeFormatInfo.InvariantInfo, DateTimeStyles.AllowWhiteSpaces |
            DateTimeStyles.RoundtripKind);
            if (data.Length > dateTimeSubstring.Length)
            {
                var timeZoneSubstring = data.Substring(dateTimeSubstring.Length).Trim(' ',
                '('', ')');
                var timeZone = TimeZoneInfo.GetSystemTimeZones().FirstOrDefault(x =>
                x.DaylightName == timeZoneSubstring || x.StandardName == timeZoneSubstring);
                if (timeZone != null)
                {
                    dateTime = dateTime.Add(-timeZone.BaseUtcOffset);
                }
            }
        }
    }
}

```

```

        if (dateTime.Kind == DateTimeKind.Unspecified)
        {
            dateTime = DateTime.SpecifyKind(dateTime, DateTimeKind.Utc);
        }

        return dateTime.ToUniversalTime();
    }
}
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using Newtonsoft.Json;

namespace Las2witsmLLIB
{
    public class Uom
    {
        string path { get; set; }
        Dictionary<string, string> uomMap { get; set; }

        public Uom(string path)
        {
            if (!String.IsNullOrEmpty(path))
            {
                this.path = path;
            }
            else
            {
                this.path = DefaultUomFile();
            }
            this.uomMap = LoadUomMap(this.path);
        }

        public string Translate(string in_uom)
        {
            return uomMap.First(x => x.Key == in_uom.ToLower()).Value;
        }

        private string DefaultUomFile()
        {
            return Directory.GetCurrentDirectory() + "/uom.json";
        }

        private Dictionary<string, string> LoadUomMap(string path)
        {
            // JSON.parse( File.open( path, 'r' ) { |f| f.read })
            StreamReader file = new StreamReader(path);
            string stringJson = file.ReadToEnd();
            var list = JsonConvert.DeserializeObject<Dictionary<string,
string>>(stringJson);
            return list;
        }
    }
}
using Las2witsmLLIB;
using System;

```



```

using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Linq.Expressions;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Web;
using System.Xml;

namespace Las2witsmlLIB
{
    class UnrecognizedUnitException: Exception
    {}

    class ConverisionError : Exception
    {
        public Dictionary<string,string> badUnits { get; set; }
        public ConverisionError(Dictionary<string, string> badUnits)
        {
            this.badUnits = badUnits;
        }
    }

    /// <summary>
    /// Класс для описания сущности witsml-файла
    /// </summary>

    public class Witsml
    {
        private ConverisionError conversionError { get; set; }
        public StreamWriter OutputStream { get; set; }
        private Uom Uom { get; set; }
        private int Indent { get; set; }
        public XmlTextWriter xmlWriter { get; set; }

        public int WitsmlVersion { get; set; }
        public List<LogCurveInfo> LogCurveInfos;
        public Dictionary<string, string> badUnits;

        public Witsml(StreamWriter outputStream, int witsmlVersion, Uom uomFile)
        {
            this.OutputStream = outputStream;
            this.WitsmlVersion = witsmlVersion;
            //this.UomFile = uomFile;
            this.Indent = 0;
            this.Uom = uomFile;
            this.conversionError = new ConverisionError(badUnits);
            this.badUnits = new Dictionary<string, string>();
            this.xmlWriter = new XmlTextWriter(outputStream);
        }

        public string MakeDateFromDouble(string data, Las las)
        {
            DateTime dtDateTime = new DateTime(1970, 1, 1, 0, 0, 0);
            DateTime startDateTime = las.StartDateTimeIndex;
            var dtf = startDateTime.Subtract(dtDateTime).TotalSeconds;
            var offset = Parsing.ParseDouble(data);
            dtDateTime = dtDateTime.AddSeconds(dtf + offset);
            //if (dtf < 86400000 && offset != 0)
            //{

```

```

        // dtDateTime = dtDateTime.AddSeconds(dtf + offset);
        //}
        //else
        //{
        // dtDateTime = dtDateTime.AddSeconds(dtf);
        //}
        return dtDateTime.ToString();
    }

    // Главный процесс
    //
    public void FromLasFile(Las las, string uidWell, string uidWellbore, string uid,
string name)
    {
        //var digestLas = DigestLas(las);
        //List<LogCurveInfo> newLcis = (List<LogCurveInfo>)digestLas[0];
        //LogCurveInfo indexLci = (LogCurveInfo)digestLas[1];
        //var isIndexIndex = digestLas[2];
        //var getIndex = digestLas[3];

        List<LogCurveInfo> newLcis = new List<LogCurveInfo>();
        newLcis = las.LogCurveInfos;

        var ns = "";
        var vers = "";

        if (WitsmlVersion >= 1410)
        {
            ns = "http://www.witsml.org/schemas/1series";
            vers = "1.4.1.0";
        }
        else
        {
            ns = "http://www.witsml.org/schemas/131";
            vers = "1.3.1.1";
        }

        var indexCurve = newLcis[0].Mnemonic;
        xmlWriter.Formatting = Formatting.Indented;
        xmlWriter.Indentation = 4;

        xmlWriter.WriteStartDocument();
        xmlWriter.WriteStartElement("logs");
        xmlWriter.WriteAttributeString("xmlns", ns);
        xmlWriter.WriteAttributeString("version", vers);

        xmlWriter.WriteStartElement("log");
        xmlWriter.WriteAttributeString("uidWell", uidWell);
        xmlWriter.WriteAttributeString("uidWellbore", uidWellbore);
        xmlWriter.WriteAttributeString("uid", uid);

        xmlWriter.WriteElementString("nameWell", name);
        xmlWriter.WriteElementString("nameWellbore", name);
        xmlWriter.WriteElementString("name", name);
        xmlWriter.WriteElementString("serviceCompany", las.ServiceCompany ?? "");
        xmlWriter.WriteElementString("description", "Created by lab212");

        var measureDepthUnit = "";
        try {
            measureDepthUnit = NormalizeUnit(las.MeasuredDepthUnit);
        }
        catch (UnrecognizedUnitException e)
        {
    }

```

```

        conversionError.badUnits.Add("measured depth", e.Message);
        measureDepthUnit = "unitless";
    }

    if (!String.IsNullOrEmpty(measureDepthUnit))
    {
        xmlWriter.WriteElementString("indexType", "measured depth");
        //xmlWriter.WriteElementString("startIndex",
las.StartMeasuredDepthIndex.ToString().Replace(',', '.'));
        xmlWriter.WriteStartElement("startIndex");
        xmlWriter.WriteAttributeString("uom", measureDepthUnit);
        xmlWriter.WriteString(las.StartMeasuredDepthIndex.ToString().Replace(',',
'.'));
        xmlWriter.WriteEndElement();
        //xmlWriter.WriteElementString("endIndex",
las.StopMeasuredDepthIndex.ToString().Replace(',', '.'));
        xmlWriter.WriteStartElement("endIndex");
        xmlWriter.WriteAttributeString("uom", measureDepthUnit);
        xmlWriter.WriteString(las.StopMeasuredDepthIndex.ToString().Replace(',',
'.'));
        xmlWriter.WriteEndElement();
    }
    else
    {
        xmlWriter.WriteElementString("indexType", "date time");
        xmlWriter.WriteElementString("startDateTimeIndex",
las.StartDateTimeIndex.ToString().Replace(',', '.'));
        xmlWriter.WriteElementString("endDateTimeIndex",
las.StopDateTimeIndex.ToString().Replace(',', '.'));
    }

    if (WitsmlVersion >= 1410)
    {
        xmlWriter.WriteElementString("indexCurve", indexCurve);
    }
    else
    {
        //xmlWriter.WriteStartElement("indexCurve", indexLci.Mnemonic);
        //xmlWriter.WriteAttributeString("columnIndex", "1");
        //xmlWriter.WriteEndElement();
    }

    xmlWriter.WriteElementString("nullValue", las.NullValue.ToString());

    foreach (var lci in newLcis.Select((x, i) => new { Value = x, Index = i }))
    {
        try
        {
            if (String.IsNullOrEmpty(las.MeasuredDepthUnit))
            {
                var start = las.StartDateTimeIndex.ToString();
                var stop = las.StopDateTimeIndex.ToString();
                AddLogCurveInfo(lci.Value, lci.Index, start, stop,
measureDepthUnit); // error
            }
            else
            {
                var start = las.StartMeasuredDepthIndex.ToString();
                var stop = las.StopMeasuredDepthIndex.ToString();
                AddLogCurveInfo(lci.Value, lci.Index, start, stop,
measureDepthUnit); // error
            }
        }
    }

```

```

    }
    catch(UnrecognizedUnitException ee)
    {
        badUnits.Add(lci.Value.Mnemonic, ee.Message);
    }
}

if (badUnits.Count == 0)
{
    xmlWriter.WriteStartElement("logData");
    if (WitsmlVersion >= 1410)
    {
        var listMnemonic = "";
        var listUnit = "";
        foreach (var lci in newLcis)
        {
            listMnemonic += lci.Mnemonic + ",";
            listUnit += lci.Unit + ",";
        }

        xmlWriter.WriteElementString("mnemonicList", listMnemonic);
        xmlWriter.WriteElementString("unitList", listUnit);
    }

    // Вот здесь записываются все значения data
    las.EachDataLine(this);
}

xmlWriter.WriteEndElement();
xmlWriter.WriteEndElement();
xmlWriter.WriteEndElement();
xmlWriter.Close();
OutputStream.Close();
}

/*
private object[] DigestLas(Las las)
{
    LogCurveInfos = las.LogCurveInfos;
    List<LogCurveInfo> newLcis = new List<LogCurveInfo>();
    Func<int, bool> isIndexIndex;
    Func<DateTime[], DateTime> getIndex;
    LogCurveInfo indexLci;

    if (!String.IsNullOrEmpty(las.MeasuredDepthUnit))
    {
        newLcis = LogCurveInfos;
        indexLci = LogCurveInfos[0];
        isIndexIndex = (i) => { return (i == 0); };
        getIndex = (values) => { return values[0]; };
    }
    else // это временной отрезок
    {
        object[] timeIndexes = MakeTimeIndexes(LogCurveInfos);
        int dateIndex = (int)timeIndexes[0];
        int timeIndex = (int)timeIndexes[1];
        string dateFormat = (string)timeIndexes[2];

        indexLci = new LogCurveInfo();
        indexLci.Mnemonic = "DATETIME";

        // rest_lcis = lcis.reject {|lci| ['time',
'date'].member?(lci.mnemonic.downcase)}

```

```

        List<LogCurveInfo> restLcis = LogCurveInfos.Where(x => x.Mnemonic.ToLower()
!= "time" && x.Mnemonic.ToLower() != "date").ToList();
        restLcis.Insert(0, indexLci);
        newLcis = restLcis;

        isIndexIndex = (i) =>
        {
            return i == timeIndex || i == dateIndex;
        };

        getIndex = (values) =>
        {
            DateTime Date = values[dateIndex];
            DateTime Time;
            DateTime dtDateTime = new DateTime(1970, 1, 1, 0, 0, 0, 0);
            var date = "";
            var time = "";

            if (dateFormat == "1")
            {
                double offset =
DateTime.UtcNow.Subtract(las.StartDateTimeIndex).TotalSeconds;
                double dtf = DateTime.UtcNow.Subtract(Date).TotalSeconds;
                if (dtf < 86400000 && offset != 0)
                {
                    dtDateTime = dtDateTime.AddSeconds(dtf + offset);
                }
                else
                {
                    dtDateTime = dtDateTime.AddSeconds(dtf);
                }
            }
            else
            {
                if (dateFormat == "yymmdd")
                {
                    date = Regex.Replace(date, @"(\d\d)(\d\d)(\d\d)", @"$2/$3/$1");
                }

                Time = values[timeIndex];
                //time = time.Substring("(\\d\\d)(\\d\\d)(\\d\\d)", "\\1:\\2:\\3");
                time = Regex.Replace(date, @"(\d\d)(\d\d)(\d\d)", @"$1/$2/$3");
                dtDateTime = Parsing.ParseDateTime(time + " " + date);
            }
            return dtDateTime;
        };
    }

    return new object[] { newLcis, indexLci, isIndexIndex, getIndex };
}

private object[] MakeTimeIndexes(List<LogCurveInfo> lcis)
{
    int dateIndex = lcis.FindIndex(x => x.Mnemonic.ToLower() == "date");
    int timeIndex = lcis.FindIndex(x => x.Mnemonic.ToLower() == "time");
    string dateFormat = "1";

    if (timeIndex == 0)
    {
        timeIndex = (dateIndex + 1);
    }
}

```

```

        if (dateIndex <= 0)
        {
            dateIndex = timeIndex;
            dateFormat = "1";
        }
        else
        {
            dateFormat = lcis[dateIndex].Unit.ToLower();
        }

        return new object[] { dateIndex, timeIndex, dateFormat };
    }

    private void AddElement(string name, string text, Dictionary<string, string>
attributes)
    {
        OutputStream.Write("<" + name);
        OutputStream.Write(" ");

        if (attributes != null)
        {
            foreach (var attr in attributes)
            {
                OutputStream.Write(attr.Key + "=" + escapeText(attr.Value) + " ");
            }
        }

        OutputStream.Write(">\n");

        OutputStream.Write(escapeText(text));

        OutputStream.Write("</" + name + ">\n");
        OutputStream.Flush();
    }

    public void AddTextElement(string name, string text, Dictionary<string, string>
attributes)
    {
        //AddElement(name, attributes);
        //OutputStream.Write("<" + name + ">");
        OutputStream.Write(escapeText(text));
        //OutputStream.Write("</" + name + ">\n");
        OutputStream.Flush();
    }

    private string escapeText(string text)
    {
        if (!String.IsNullOrEmpty(text))
        {
            return HttpUtility.HtmlEncode(text.Trim());
        }
        return "";
    }
    */

    private void AddLogCurveInfo(LogCurveInfo lasLci, int columnIndex, string minIndex,
string maxIndex, string measureDepthUnit)
    {
        xmlWriter.WriteStartElement("logCurveInfo");
        xmlWriter.WriteAttributeString("uid", lasLci.Mnemonic);
        xmlWriter.WriteElementString("mnemonic", lasLci.Mnemonic);
        xmlWriter.WriteElementString("unit", lasLci.Unit);
    }

```

```

        if (!String.IsNullOrEmpty(measureDepthUnit))
        {
            //xmlWriter.WriteElementString("minIndex", minIndex.ToString().Replace(',',
            '.'));

            xmlWriter.WriteStartElement("minIndex");
            xmlWriter.WriteAttributeString("uom", measureDepthUnit);
            xmlWriter.WriteString(minIndex.ToString().Replace(',', '.'));
            xmlWriter.WriteEndElement();
            //xmlWriter.WriteElementString("maxIndex", maxIndex.ToString().Replace(',',
            '.'));

            xmlWriter.WriteStartElement("maxIndex");
            xmlWriter.WriteAttributeString("uom", measureDepthUnit);
            xmlWriter.WriteString(maxIndex.ToString().Replace(',', '.'));
            xmlWriter.WriteEndElement();
        }
        else
        {
            xmlWriter.WriteElementString("minDateTimeIndex", minIndex.ToString());
            xmlWriter.WriteElementString("maxDateTimeIndex", maxIndex.ToString());
        }

        if (WitsmlVersion < 1410)
        {
            xmlWriter.WriteElementString("columnIndex", columnIndex.ToString());
        }

        xmlWriter.WriteElementString("curveDescription", lasLci.Description);
        xmlWriter.WriteElementString("typeLogData", "float");

        xmlWriter.WriteEndElement();
    }

    private string NormalizeUnit(string lasUnit)
    {
        if (String.IsNullOrEmpty(lasUnit))
        {
            return lasUnit;
        }
        else
        {
            string retval = Uom.Translate(lasUnit);
            if (String.IsNullOrEmpty(retval))
            {
                throw new UnrecognizedUnitException();
            }
            return retval;
        }
    }
}
}
}

```