|  |  |  |
| --- | --- | --- |
| Территориальный фонд обязательного медицинского страхования Московской области | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| автоматизированная информационная система «Эксперт» | | |
| Частное техническое задание на доработку систем СМО для взаимодействия с АИС ТФОМС | | |
|  | | |
| ТФОМС-0248100000118000125-ЧТЗ-04 | | |
| На 18 листах | | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Содержание

1 Общие сведения 3

2 Назначение и цели доработки систем СМО 4

2.1 Назначение 4

2.2 Цели 4

3 Характеристики объекта автоматизации 5

4 Требования к доработке систем СМО 6

4.1 Требования к порядку взаимодействия 6

4.2 Требования к реализации коммуникационного протокола и системы команд для взаимодействия с АИС ТФОМС МО 6

4.2.1 Основные сущности 6

4.2.2 Обработка ошибок 9

4.2.3 Порядок установки соединения 11

4.2.4 Порядок передачи коммуникационных пакетов через TCP-сокет 11

4.2.5 Порядок получения данных системой СМО из АИС ТФОМС МО 12

4.2.6 Перечень команд 13

4.3 Требования к формату данных для информационного обмена 15

Перечень сокращений 17

Термины и определения 18

# Общие сведения

Настоящий документ определяет требования к доработке систем страховых медицинских организаций (далее — СМО) для обеспечения их взаимодействия с Автоматизированной информационной системой территориального фонда медицинского страхования Московской области (далее — АИС ТФОМС МО).

Документ разработан в рамках выполнения работ по 2 этапу Государственного контракта № 0248100000118000125 от 10.12.2018.

# Назначение и цели доработки систем СМО

## Назначение

Доработка систем СМО может быть выполнена в связи с внедрением в ФОМС МО Автоматизированной информационной системы контроля качества, объемов, сроков и условий предоставления медицинской помощи посредством отбора счетов для проведения МЭЭ и ЭКМП на основе алгоритмов искусственного интеллекта «ЭКСПЕРТ» (далее — ).

## Цели

Целями выполнения работ по доработке систем СМО являются:

* повышение качества, стандартизация и автоматизация процессов МЭЭ и ЭКМП;
* вывод процесса контроля качества, объемов, сроков и условий предоставления медицинской помощи на качественно новый уровень;
* снижение влияния человеческого фактора при проведении МЭЭ и ЭКМП.

Для доработки систем СМО необходимо:

* провести анализ состава и структур данных, необходимых для взаимодействия с АИС ТФОМС МО;
* реализовать коммуникационный протокол и набор команд для взаимодействия с АИС ТФОМС МО;
* реализовать прием из АИС ТФОМС МО данных, обработанных в .

# Характеристики объекта автоматизации

В ТФОМС МО внедрена , назначением которой является автоматизация деятельности ТФОМС МО в части проведения медико-экономической экспертизы (далее — МЭЭ) и экспертизы качества медицинской помощи (далее — ЭКМП).

МЭЭ и ЭКМП проводятся на основании приказа Федерального фонда обязательного медицинского страхования от 1 декабря 2010 г. № 230 «Об утверждении Порядка организации и проведения контроля объемов, сроков, качества и условий предоставления медицинской помощи по обязательному медицинскому страхованию».

выполняет для каждого элемента данных (счета) две независимых бинарных классификации, присваивая каждому счету количество очков (score) – два независимых действительных числа из диапазона [0;1]. Большее количество очков (score) соответствует большей вероятности того, что случай оказания медицинской помощи из реестра счета подлежит проведению МЭЭ и/или ЭКМП соответственно.

Результаты расчетов направляет в АИС ТФОМС МО, откуда они могут быть выгружены системами СМО для использования в своих процессах. Для получения данных из АИС ТФОМС МО необходима доработка систем СМО в соответствии с требованиями настоящего документа.

Далее в документе (раздел 4) приведены требования по обеспечению интеграции АИС ТФОМС МО и системами СМО в части обмена данными для поддержки процессов проведения МЭЭ и ЭКМП.

# Требования к доработке систем СМО

## Требования к порядку взаимодействия

Для реализации взаимодействия необходимо получить доступ к АИС ТФОМС МО (VPN-подключение или иной способ, согласованный с Заказчиком). АИС ТФОМС находится в закрытом контуре локальной сети ТФОМС МО.

## Требования к реализации коммуникационного протокола и системы команд для взаимодействия с АИС ТФОМС МО

Используемый протокол – TCP.

Формат обмена данными – RFC 7159 THE JavaScript Object Notation (JSON) Data Interchange Format.

### Основные сущности

Для организации информационного обмена должны использоваться следующие сущности:

* Команда (command);
* Сообщение (message);
* Контент сообщения (content).

Описание назначения основных сущностей и порядка их использования приведено в подразделах 4.2.1.1 – 4.2.1.3.

#### Команда

Команда — это 128-битный идентификатор (UUID), который идентифицирует выполняемое действие. Команда характеризуется следующими состояниями:

* command — непосредственно прямая команда на выполнение действия;
* answer — ответ (или результат) выполнения команды с типом command;
* event — событие; команда, не предполагающая получения ответа (answer). Используется для рассылки широковещательных сообщений о событиях.

Примеры команд:

QUuidEx Unknown = "4aef29d6-5b1a-4323-8655-ef0d4f1bb79d"; // Идентификатор

// неизвестной команды

QUuidEx Error = "b18b98cc-b026-4bfe-8e33-e7afebfbe78b"; // Идентификатор сообщения

// об ошибке

#### Сообщение

Сообщение — контейнер, используемый для пересылки команды совместно с данными, необходимыми для выполнения этой команды. Для JSON-варианта сообщения наиболее значимыми полями являются:

* id — персональный идентификатор сообщения (UUID);
* command — идентификатор команды (UUID);
* flags — поле битовых флагов, используется для оптимизации при сериализации сообщения;
* type — тип пересылаемой команды: command, answer, event (содержится внутри поля flags);
* execStatus — статус выполнения/обработки команды (содержится внутри поля flags);
* priority — приоритет сообщения (содержится внутри поля flags);
* maxTimeLife — максимальное время жизни сообщения (для реализации синхронных вызовов);
* content — контент сообщения (полезная нагрузка, сериализованные данные);
* tag — содержит произвольные пользовательские данные размером не более 8 байт без сохранения их в поле content.

Пример записи сообщения в JSON-представлении:

{

"id":"3caca371-bab1-4758-98db-e1251e49d45f",

"command":"2ec05fe5-366a-41c9-a283-5a617767738d",

"flags":2164267082,

"content":{"users":["user1","user2","user3"]}

}

#### Контент сообщения

Контент — это данные (полезная нагрузка), передаваемые с конкретной командой. Контент может передаваться с любым типом сообщения (command, answer, event). В контексте языка С++ контент описывается с помощью структуры данных (struct). Для того, чтобы установить ассоциативную связь между командой и контентом сообщения (структурой данных), используется механизм наследования от базового класса Data.

Так, описание контента сообщения для команды CloseConnection будет выглядеть следующим образом:

struct CloseConnection : Data<&command::CloseConnection,

Message::Type::Command>

{

qint32 code = {0}; // Код причины. Нулевой код соответствует

// несовместимости версий протоколов.

QString description; // Описание причины закрытия соединения

};

В этом примере со структурой CloseConnection ассоциируется команда command::CloseConnection, а также структуре назначается атрибут Message::Type::Command. Это означает, что данная структура может быть использована в качестве контента только в сообщениях с типом command, но не может быть использована в сообщении с типами answer и event.

### Обработка ошибок

Для информировании об ошибочной ситуации должна использоваться команда Error:

QUuidEx Error = "b18b98cc-b026-4bfe-8e33-e7afebfbe78b"

Контент для команды Error выглядит следующим образом:

struct Error : Data<&command::Error,

Message::Type::Command>

{

QUuidEx commandId; // Идентификатор команды для которой произошла ошибка

qint32 code = {0}; // Код ошибки

QString description; // Описание ошибки

};

Другим вариантом является возврат информации об ошибке в сообщении с типом Message::Type::Answer. Для этой цели определены две структуры:

struct MessageError

{

qint32 code = {0}; // Код ошибки

QString description; // Описание ошибки

};

struct MessageFailed

{

qint32 code = {0}; // Код неудачи

QString description; // Описание неудачи

};

Структура MessageError предназначена для описания ошибки при выполнении команды. Структура MessageFailed используется в том случае, когда выполнение команды не завершилось ошибкой, но и успешного результата тоже нет.

У сообщения есть специальное поле — execStatus, оно описывает результат выполнения операции. Поле execStatus имеет смысл только для сообщений с типом Message::Type::Answer, в остальных случаях оно имеет неопределенное состояние. Поле execStatus может принимать следующие значения:

* Unknown = 0 — Неопределенное состояние.
* Success = 1 — Сообщение было обработано успешно и содержит корректные ответные данные.
* Failed = 2 — Сообщение не было обработано успешно, но результат не является ошибкой. В данном случае сообщение будет содержать данные в формате MessageFailed.
* Error = 3 — При обработке сообщения произошла ошибка, и в качестве ответа отправляется сообщения с описанием причины ошибки. В данном случае сообщение будет содержать данные в формате MessageError.

Пример записи сообщения с параметром execStatus = Error в JSON-представлении:

{

"id":"7a411ec0-ef09-4bde-a5a5-bc8c709c631d",

"command":"fc7f12a8-9643-4e81-9d27-53b99c19d319",

"flags":2164267090,

"content":{"code":0,"description":"Failed user authorization"}

}

### Порядок установки соединения

После установки TCP-соединения клиент должен отправить на сокет сервера 16 байт (обычный UUID), это сигнатура протокола. Строковое представление сигнатуры для JSON-формата: "fea6b958-dafb-4f5c-b620-fe0aafbd47e2".

Если сервер возвращает этот же UUID, то возможно информационное взаимодействие и далее можно работать, иначе TCP-соединение разрывается.

Сервер ожидает получения сигнатуры в течение 3 секунд, клиент ожидает ответа от сервера в течение 6 секунд. Если в течение указанного времени сигнатуры не будут получены, TCP-соединение будет разорвано.

После того, как сигнатуры протокола проверены, клиент и сервер отправляют друг другу первое сообщение — ProtocolCompatible. Это сообщение помимо проверки версий протокола запускает очередь обработки сообщений; если это сообщение не будет отправлено, механизм не сможет корректно работать, никакие другие сообщения, кроме CloseConnection не будут обрабатываться.

### Порядок передачи коммуникационных пакетов через TCP-сокет

После проверки совместимости сокеты обмениваются коммуникационными пакетами в следующем формате:

* поле 1: 4 байта содержат размер пакета (поле соответствует типу integer);
* поле 2: пакет данных длиной, указанной в поле 1.

Примечания:

1. размер пакета в поле 1 записывается в сетевом представлении (порядок байт big-endian);
2. если размер пакета меньше нуля, это означает, что пакет сжат с использованием zip-алгоритма. В этом случае размер пакета (поле 1) имеет значение -1. После чтения пакета из буфера сокета нужно провести его декомпрессию.
3. Механизм сжатия (п. 2) может быть отключен по согласованию взаимодействующих сторон.

### Порядок получения данных системой СМО из АИС ТФОМС МО

Для получения данных предназначены следующие команды:

1. SmoScores — запрос на получение оценок системой СМО из АИС ТФОМС МО. В качестве ответа возвращается список оценок;
2. RequestSmoScore — запрос из системы СМО в АИС ТФОМС МО на получение данных для конкретного отчета.
3. GetSmoScoreData — получение данных из АИС ТФОМС МО по оценке, запрошенной ранее при помощи команды RequestSmoScore.

Для начала информационного обмена система СМО должна отправить АИС ТФОМС МО команду SmoScores. В ответ на команду АИС ТФОМС МО возвращает список выполненных оценок.

Система СМО должна отправить команду RequestSmoScore на получение данных конкретного отчета из полученного ранее списка.

АИС ТФОМС МО подготавливает данные и сообщает о готовности путем отправки положительного ответа системе СМО. Если при подготовке данных возникли ошибки, то в ответ должно быть отправлено сообщением со статусом «MessageFailed». При получении подобного сообщения система СМО должна прекратить сеанс с АИС ТФОМС МО.

В случае положительного ответа от АИС ТФОМС МО на подготовку данных, система СМО должна отправить команду GetSmoScoreData.

Система СМО заполняет параметр reqCount команды GetSmoScoreData, указывающий на количество строк данных, которое АИС ТФОМС МО должна передать в сообщении.

АИС ТФОМС МО, получив команду GetSmoScoreData приступает к извлечению из буфера строки данных в количестве reqCount. Далее происходит создание ответного сообщения, в котором заполняются следующие параметры:

* isLast – устанавливается значение «true», если данных для отправки больше нет;
* reqCount – параметр, заполненный системой СМО (не изменяется);
* totalCount – общее количество строк данных, помещенных в сообщении;
* items – строки данных.

При возникновении ошибки в процессе отправки данных АИС ТФОМС МО вернет системе СМО сообщение GetSmoScoreData со статусом «MessageFailed». Система СМО в этом случае должна прервать сеанс связи с АИС ТФОМС МО.

Процесс отправки сообщения GetSmoScoreData повторяется до тех пор, пока системой СМО не будет получено сообщение с параметром isLast, равным «true». Также процесс отправки прерывается в случае получения сообщения с ошибкой.

### Перечень команд

Сводный перечень команд для организации информационного обмена приведен в таблице 1.

Таблица 1 – Команды для взаимодействия и АИС ТФОМС МО

| № | Команда | Идентификатор команды | Описание |
| --- | --- | --- | --- |
| Базовые команды | | | |
|  | Unknown | 4aef29d6-5b1a-4323-8655-ef0d4f1bb79d | Неизвестная команда |
|  | Error | b18b98cc-b026-4bfe-8e33-e7afebfbe78b | Сообщение об ошибке |
|  | ProtocolCompatible | 173cbbeb-1d81-4e01-bf3c-5d06f9c878c3 | Запрос информации о совместимости (при подключении клиент и сервер отправляют друг другу информацию о совместимости): |
|  | CloseConnection | e71921fd-e5b3-4f9b-8be7-283e8bb2a531 | Требование закрыть соединение.  Сторона, желающая закрыть соединение, отправляет это сообщение с информацией о причине необходимости закрыть соединение. Принимающая сторона записывает эту информацию в свой лог (или использует иным образом), затем отправляет обратное пустое сообщение.  После того, как ответное сообщение получено, TCP-соединение может быть разорвано. Такое поведение реализовано для того, чтобы сторона, с которой разрывают соединение, имела информацию о причине разрыва. |
| Команды для обмена данными систем СМО с АИС ТФОМС МО | | | |
|  | SmoScores | d307296e-fc68-432e-8972-54060e439347 | Запрос на получение оценок системой СМО из АИС ТФОМС МО. В качестве ответа возвращается список оценок. |
|  | RequestSmoScore | a372111a-f4af-4861-9c61-eb3ab1b08451 | Запрос из системы СМО в АИС ТФОМС МО на получение данных для конкретного отчета.  Если есть готовность предоставить данные по запрашиваемой оценке, то не обходимо дать положительный ответ.  Данные передаются с использованием команды GetSmoScoreData. |
|  | GetSmoScoreData | 8b67a97a-576f-4dfa-93ba-22ea4a374752 | Команда отправляется из системы СМО в АИС ТФОМС МО для получения порции данных по оценке, запрошенной ранее при помощи команды RequestSmoScore.  Параметры:   * isLast – флаг-признак последнего блока; * reqCount – количество запрашиваемых строк данных; * totalCount – фактическое количество строк данных; * items – строки данных. |
|  | BreakDataTransfer | cbff8e30-0dfe-11e9-ab14-d663bd873d93 | Запрос на прерывание обмена данными. Данное сообщение может быть отправлено как системой СМО, так и АИС ТФОМС МО. |

## Требования к формату данных для информационного обмена

Состав данных для информационного обмена между АИС ТФОМС МО и системами СМО:

1. Список моделей:

* Идентификатор модели.
* Дата начала.
* Дата окончания.
* Идентификатор пользователя.

1. Перечень результатов обработки в (по выбранной модели):

* Перечень ключевых полей;
* Score по МЭЭ;
* Score по ЭКМП.

Перечень сокращений

|  |  |  |
| --- | --- | --- |
| АИС | – | Автоматизированная информационная система |
| БД | – | База данных |
| ГК | – | Государственный контракт |
| ГОСТ | – | Государственный стандарт |
| МО | – | Московская область |
| МЭЭ | – | Медико-экономическая экспертиза |
| ПО | – | Программное обеспечение |
| РД | – | Руководящий документ |
| ТФОМС | – | Территориальный фонд обязательного медицинского страхования |
| ЭКМП | – | Экспертиза качества медицинской помощи |

Термины и определения

|  |  |  |
| --- | --- | --- |
| Машинное обучение | – | класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач |
| Размеченные данные | – | совокупность счетов, для которых есть отметка эксперта о результатах выполнения МЭЭ или ЭКМП |
|  | – | Автоматизированная информационная система контроля качества, объемов, сроков и условий предоставления медицинской помощи посредством отбора счетов для проведения МЭЭ и ЭКМП на основе алгоритмов искусственного интеллекта «ЭКСПЕРТ» |