АНАЛИЗ ВОПРОСОВ РАЗРАБОТКИ МЕДИЦИНСКОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

А.А. Чувицкий, А.В. Бояровская, Н.А. Борсука

КФ МГТУ им. Н.Э. Баумана, г. Калуга

Охрана здоровья граждан – одна из ключевых тем развития государства. О необходимости выстроить систему здравоохранения на новой технологической базе заявил в Послании Федеральному Собранию 21.04.2021 Президент РФ Владимир Путин. Здравоохранение представляет собой достаточно сложную и динамическую систему, которая требует постоянной оценки ситуации и быстрого принятия решений. Непрерывная оценка эпидемиологической ситуации и мониторинг здоровья конкретного гражданина невозможен без качественно организованного обмена информацией и единой системы данных, внедренных в повседневную работу медицинских учреждений. На данный момент существует большое количество медицинских информационных систем (МИС), однако, многие из них содержат избыточный функционал и специфичный формат хранения данных. Адаптировать существующие МИС под нужды конкретных регионов довольно проблематично как в трудо- и времязатратах, так и финансовом плане. Была поставлена задача разработать медицинскую информационную систему Астрал.Мед, направленную на решение актуальных задач в сфере оказания первичной медико-санитарной помощи. Одной из задач разработки является возможность передачи и приема различной медицинской информации в уже существующие МИС с целью, в конечном итоге, отправки этих данных в Единую государственную информационную систему в сфере здравоохранения. В данном случае должно реализовываться единое хранилище всех данных пациентов независимо от медицинского учреждения, в котором они обслуживаются, и от информационной системы в данном учреждении. Реализация системы Астрал.Мед состоит из: разработки модуля интерфейсной части; разработки базы данных и способа взаимодействия с ней; модуля обработки запросов от специалистов медицинского учреждения и модуля интеграции со сторонними МИС (что является одной из самых основных и сложных задач из-за возможной несовместимости форматов хранения данных в различных информационных системах).

Ключевые слова: МИС, брокер сообщений, модель данных, сериализация и десериализация данных, хранимая куча, источник данных, мигратор сущностей

ВВЕДЕНИЕ

Важнейшей сферой общественной жизни сегодня является здравоохранение. В эпоху Covid-19 назрела необходимость трансформации национальных систем охраны здоровья. Пандемия выявила системные недостатки в организации медицины во всех странах мира. В России одной из таких проблем является ускорение процесса цифровизации здравоохранения.

Внедрение современных информационных технологий является важной задачей: обществу необходимы действия по повышению готовности к возможным угрозам в области общественного здоровья, а использование цифровых решений должно повысить безопасность пациентов и качество системы здравоохранения; также использование информационной системы обеспечит большую точность диагностики проблем, связанных со здоровьем человека, и поможет врачу вести прием комплексно, оказывая медицинскую помощь согласно стандартам и порядку оказания медицинской помощи и клинических рекомендаций [1].

Цифровизация отрасли здравоохранения должна основываться на принципах непрерывности лечебно-диагностического процесса, профилактической направленности в деятельности учреждений здравоохранения.

В практике организации деятельности медицинских учреждений можно встретить большое

количество решений, призванных облегчить работу врачей и среднего медицинского персонала. Крупные системы, которые используются в большинстве регионов, действительно пытаются автоматизировать отдельные области деятельности врачей, но в то же время многие из них не учитывают ряд особенностей, основанных на специфике организации охраны здоровья. Важными факторами для использования систем медицинским сообществом являются: удобный и понятный интерфейс для работы в информационной базе, интеграция с другими системами, что на данный момент затруднительно, поскольку стандарт обмена данными является уникальным для каждой из них.

На данный момент в России зарегистрированы сведения о порядке 800 различных программных продуктов для клиник, в том числе и медицинских информационных систем (МИС) [2]. В настоящее время часто можно увидеть лечебное учреждение, где МИС активно используется в повседневной работе врачей и других медицинских работников. Ведение приемов, анализ случаев заболеваний пациентов, а также отчетная деятельность автоматизирована в большом количестве медицинских организаций, как частных, так и государственных.

Но на этом развитие МИС не заканчивается. Помимо доработок существующего функционала, осуществляется и внедрение нового. Так, с помощью, внедряемой в Белгородской области ТМ:МИС

(СофтТраст) [3], помимо тривиальной деятельности врачей, недавно была реализована система электронных рецептов. 19 апреля 2019 года впервые в России коммерческой аптекой произведен отпуск «электронного рецепта», выписанного в «ТМ:МИС» по форме №107-1/у, а 25 апреля отпущен льготный рецепт в электронной форме.

Что касается дальнейшего развития информационных технологий в сфере медицины, то по словам министра здравоохранения России Михаила Мурашко, одной из ближайших задач системы здравоохранения является реализация национального проекта и государственной программы по модернизации первичного звена, а также пула проектов по цифровой трансформации, крайне важных для населения, и в том числе по коммуникации между медицинской организацией и пациентом [4].

Сейчас, в России продолжается внедрение МИС в бюджетных учреждениях здравоохранения, имеется тенденция к централизованному использованию единой МИС для целого региона. В большинстве случаев существующие системы разрабатываются сразу под все контексты медицинского учреждения, то есть системы позволяют проводить все действия для организации процесса оказания медицинской помощи населению. Рассмотрим некоторые существующие МИС.

КМИС.Регион - это централизованная облачная региональная медицинская информационная система, включающая такие модули, как региональная НСИ, центральный реестр пациентов (ЦРП), интегрированная электронная медицинская карта (ИЭМК), подсистемы "Паспорт МО", "Регистр медработников", "Электронный рецепт", "Нозологические регистры" и так далее [5]. Пример интерфейса ИЭМК приведен на рисунке 1.

Рис. 1. Интерфейс ИЭМК пациента в КМИС.Регион

ТрастМед - программный комплекс для построения единой государственной системы здравоохранения региона полностью автоматизирует медицинское учреждение: поликлинику, стационар, стоматологическую клинику, больничную аптеку [6]. Внешний вид интерфейса показан на рисунке 2.

Рис. 2. Интерфейс ИЭМК в ТрастМед

РМИС - программный комплекс, который обеспечивает автоматизацию основных процессов медицинского учреждения, а также обмен информацией между региональными и федеральными информационными ресурсами в лечебно-диагностическом процессе. РМИС обеспечивает повышение качества и доступности медицинской помощи населению за счет автоматизации медицинской и административной деятельности; ведение медицинской документации в электронном виде (электронная медицинская карта - ЭМК); обеспечение персонифицированного учета оказания медицинских услуг [7].

Внешний вид интерфейса показан на рисунке 3.

Рис. 3. Интерфейс РМИС

Проведя сравнительный анализ наиболее распространённых МИС, было выявлено, что у большинства из них отсутствуют удобный и понятный пользовательский интерфейс, система помощи принятия решений, стандартизированное хранение медицинских данных, также многие системы хранят в себе избыточную информацию, что делает их более дорогостоящими и требующими значительно больших объемов виртуальной памяти. Следует отметить, что формат хранения данных представлен в обычных таблицах, таблицы созданы под каждое поле (для имени одна таблица, для фамилии другая таблица и т.д.).

При внедрении существующих МИС данные платформы частично получают отрицательные оценки при организации автоматизированного процесса оказания медицинской помощи. Поэтому для создания более адресного продукта имеется смысл в разработке собственной медицинской

информационной системы, которая способна, в том числе, организовать унифицированную базу хранения информации о пациенте, чтобы из любой МИС того или иного региона имелась возможность получить доступ к медицинским данным человека, обратившегося в медицинскую организацию.

Информационное обеспечение в сфере здравоохранения осуществляется посредством создания, развития и эксплуатации: федеральных государственных информационных систем в сфере здравоохранения; систем фонда обязательного медицинского страхования (ИС ФОМС); государственных информационных систем в сфере здравоохранения субъектов РФ (ГИС субъектов РФ), медицинских информационных систем медицинских организаций (МИС МО), информационных систем фармацевтических организаций (ИС ФО). В настоящее время свыше 210,5 млн рублей направлено на развитие Единой государственной информационной системы в сфере здравоохранения (ЕГИСЗ). Это поможет ускорить перевод медицинских и государственных услуг в электронный вид и предоставлять их по принципу цифровых суперсервисов [8].

Взаимодействие медицинских информационных систем, в том числе разрабатываемой Астрал.Мед, с Единой государственной информационной системой в сфере здравоохранения (ЕГИСЗ) представлено на рисунке 4. Разрабатываемая и анализируемая в статье система Астрал.Мед должна взаимодействовать с любой МИС и ЕГИСЗ путем формализации передаваемых и принимаемых медицинских данных через один из важных модулей - брокер сообщений, который связан с собственным хранилищем данных. В ближайшем будущем целесообразно объединять и хранить формализованную информацию о пациенте в едином хранилище данных для быстрого доступа к этой информации из любой МИС МО.

Рис. 4. Взаимодействие МИС

Любая МИС является клиент-серверным приложением, поэтому если врач в клиентской части медицинской системы изменяет административную

информацию пациента, эта информация отправляется на серверную часть приложения, на которой происходит валидация и первичная обработка полученных с клиентской части данных, после чего необходимо передать эти данные в системы, с которыми происходит интеграция.

Вопрос разработки модуля брокера сообщений, который будет рассмотрен далее, является одним из важных при реализации задачи общего доступа в единую базу данных из любой медицинской системы. Так, сервер отправляет данные на модуль брокера сообщений, который занимается процессом доставки данных до сторонних систем.

По причине того, что сторонние МИС дискретны и в определенный момент могут быть недоступны, необходимо обеспечить доставку данных в будущем, когда системы возобновят работу. С этой целью необходимо на стороне модуля брокера сообщений реализовать хранилище данных, которое будет хранить запросы и при восстановлении сторонней системы, возобновлять очередь сообщений.

ОСНОВНАЯ ЧАСТЬ

Основными задачами разрабатываемой информационной системы Астрал.Мед является создание программного комплекса, который позволит автоматизировать работу медицинского учреждения, а также создание архитектуры, которая позволит передавать данные между медицинскими системами и организовать полное взаимодействие между ними. Основной системой для интеграции является РМИС. При этом необходимо производить синхронизацию данных пациентов и различных справочников. Необходимо реализовать внешнюю авторизацию для удобного перехода врача из РМИС в Астрал.Мед.

Реализация данной системы производится по принципу микро-сервисной архитектуры, которая позволит легко разрабатывать отдельные части приложения, которые не связаны друг с другом, что поможет производить простое обновление конкретных блоков без затрагивания остальных. Также данное решение позволит легко масштабировать высоконагруженные блоки и тем самым получить большую гибкость.

Основные структурные блоки серверной части представлены на рисунке 5. Данная структура показывает, как представлены данные в приложении и связи между моделями данных.

Рис. 5. Структурные блоки приложения

Для автоматизации развёртывания, масштабирования управления приложением в качестве кластера используется Kubernetes.

Каждый отдельный микросервис представляет собой один контейнер, который в себе содержит всю инфраструктуру сервиса, все связи, а кластер, соединяет все контейнеры в единую структуру, также предоставляет удобную систему легирования, анализа нагрузки каждого контейнера и при необходимости реализации горизонтального масштабирования, добавлении контейнеров на другом железе и добавлении его в общую структуру проекта.

В качестве контейнеризации используется Docker, который позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть перенесён на любую Linux-систему, а также предоставляет среду по управлению контейнерами.

В качестве интерфейса пользователя задействован обычный браузер.

Данная система разделяется на следующие микросервисы: микросервис для работы с электронной медицинской карточкой пациента; микросервис медицинской информации; GraphQL – микросервис, участвующий точкой входа в проект, который производит маршрутизацию запросов внутри проекта; фронтенд; Redis – микросервис кэширования данных; Identity – микросервис для работы с идентификацией пользователей, который включает в себя регистрацию, авторизацию и аутентификацию; Aidbox – микросервис для работы с базой данных.

Большим вопросом является реализация работы с медицинскими данными. Обычные реляционные базы данных не позволяют реализовать быструю работу с информацией в связи со своим принципом работы, также сама организация данных в различных системах разная, это связано со сложностью медицинского контекста. Поэтому было принято решение строить всю архитектуру моделей данных и

их хранения по международному стандарту FHIR (Fast Healthcare Interoperability Resources).

Так как стандарт очень сложен в своей реализации и содержит множество контекстов для интеграции с другими системами, различными терминологиями и реализацией контекстов, и поддержкой внутреннего сервера, было принято решение использовать готовый FHIR сервер и базу данных, которая хранит данные в формате стандарта систему Aidbox. Aidbox предоставляет автоматизированную инфраструктуру, включая базу данных с FHIR-схемой, FHIR REST API и целую экосистему расширений, также поддерживает систему контейнеризации через Docker, что позволяет без каких-либо проблем встраивать данную систему в проект.

На основе анализа научных материалов по решению вопросов интеграции между различными системами сделан вывод о том, что необходимо учитывать структуру и размерность форматов хранения данных в различных ИС во избежание искажения и потери информации или неверного истолкования при передаче и записи этой информации от одной системы к другой [9, 10, 11].

Преобразование информации из различных баз данных связано с различием логических структур данных, а также с многообразием представления данных, с различиями в логических структурах данных, в справочниках, классификаторах и в системах кодирования информации, в использовании различных языков для представления информации.

Преобразование данных может проводиться разными способами. Например, путем создания специальных скриптов и файлов с учетом ранее введенных данных без их дублирования и корректного приведения несовместимых типов данных (рисунок 6), но на данном этапе могут возникнуть ошибки, связанные с изменением форматов данных, дополнением старых справочников новыми данными, также данный метод является не целесообразным, так как для каждой системы необходимо писать собственный скрипт. Поэтому оптимальным решением является разработка универсального преобразователя данных.

Рис. 6. Реализация преобразования данных

Согласно 41 статье Конституции РФ, каждый имеет право на охрану здоровья и медицинскую помощь. Получить её гражданин может в государственных, муниципальных и частных учреждениях здравоохранения. Конституционное право граждан на медицинское обслуживание создает у учреждений потребность в формализации данных. Организации, являясь независимыми друг от друга субъектами, вправе использовать различные медицинские информационные системы. Однако, это не должно сказываться на пациентах, а полученные в результате оказания медицинских услуг данные, могли бы передаваться в различные учреждения без искажений.

Для решения данной проблемы необходимо реализовать модуль, позволяющий производить инфраструктурное взаимодействие со сторонними медицинскими системами (брокер сообщений), который будет преобразовывать данные и хранить запросы систем до возобновления связи с ними. Рассмотрим реализацию данного модуля подробнее.

Основной частью модуля брокера сообщений является хранилище данных. В него входят компоненты, представленные на рисунке 7.

Рис. 7. Реализация хранилища данных

1) Модель данных.

Под моделью данных понимается конкретная структура данных, содержащая полезную нагрузку, которая будет передана в стороннюю систему и которая будет сохраняться в хранилище. Для примера модель данных для обновления информации о пациенте представлена на рисунке 8.

```

/// <summary> Информация пациента на обновление.
5 references
public class PatientUpdateFm
{
    /// <summary> Идентификатор пациента.
    2 references
    public int Id { get; set; }
    /// <summary> Фамилия.
    1 reference
    public string Surname { get; set; }
    /// <summary> Имя.
    1 reference
    public string Name { get; set; }
    /// <summary> Отчество.
    1 reference
    public string Patronymic { get; set; }
    /// <summary> Дата рождения.
    1 reference
    public DateTime BirthDate { get; set; }
    /// <summary> Пол.
    1 reference
    public int Gender { get; set; }
    /// <summary> Телефон.
    1 reference
    public string Phone { get; set; }
    /// <summary> Законный представитель.
    1 reference
    public LegalRepresentativeUpdateFm LegalRepresentative { get; set; }
}

```

Рис. 8. Модель данных для обновления пациента

2) Сериализация и десериализация данных.

Метод класса сериализатора возвращает количество байт, в которое будет сериализован экземпляр, другой метод принимает экземпляр модели данных и в массив байт (buffer) будет записать байты с указанным смещением (offset). В методе сериализации вызывается проверка переданных параметров на валидность (buffer не равен null, или offset не превышает диапазон массива), после чего по указателю записываются данные.

Метод десериализатора из массива байт и смещения вернёт экземпляр модели данных.

Любая модель данных состоит из примитивных типов (целочисленные значения, строки, массивы и т.п.). Для создания сериализатора составного типа, необходимо создать сериализаторы всех примитивов. Для сериализации массивов используется сериализатор элемента массива и перед последовательной сериализацией каждого элемента, вначале записывается его длина с помощью сериализатора примитива Int32.

Логика сборки сериализаторов разрабатывается на основе рефлексии типов и программирования на промежуточном языке виртуальной машины CLR.

3) Хранимая куча.

Все данные хранимой кучи будут сохраняться в два физических файла: сегменты (segments) и данные (data). Всё состояние кучи будет определяться её сегментами, а каждый сегмент, в свою очередь определяет определённый участок из data. Сегмент – структура, хранящая указатель и размер.

Также необходимо учитывать освобождение участка памяти по указанному сегменту, производить дефрагментацию данных.

Для быстрого выделения памяти и поиска сегментов, сегменты хранятся в памяти в структуре данных – красно-чёрное дерево, которое производит самобалансировку при добавлении и удалении элементов. Основным преимуществом такой структуры данных является компактность хранения информации и логарифмический доступ к ней.

4) Источник данных.

Источник данных предоставляет удобный интерфейс для работы с наборами информации. Он использует хранимую кучу и поверх неё выстраивает структуру хранения моделей данных и метайнформацию об источнике данных.

Для доступа к данным используется структура, которая позволяет добавлять экземпляры модели данных, искать экземпляр по идентификатору, удалять его и возвращать все экземпляры, хранящиеся на диске данного типа. Набор данных хранит начальные индексы, представленные также красно-чёрным деревом. Это дерево в качестве ключа хранит идентификатор модели данных, а по значению сегмент (дескриптор, где хранится сущность в хранимой куче).

Структура источника данных представлена на рисунке 9. HeapMetaPointer – сегмент, указывающий начало, где хранится мета информация кучи. Данный сегмент можно получить как сегмент инициализации, который предоставляет хранимая куча, так как этот сегмент может находиться, где угодно в памяти, не обязательно в самом начале. По этому указателю источник данных загружает метайнформацию, которая содержит указатели на наборы информации определенных моделей данных. По данному указателю загружается информация по конкретному набору, она содержит указатель на дерево первичных ключей, при выгрузке данного дерева инициализируется набор информации для конкретной модели данных. По ключу идентификатора сущности возвращается указатель на место, где хранится непосредственно сущность.

Рис.9. Структура источника данных

5) Мигратор сущностей.

Мигратор сущностей хранит метайнформацию о моделях в разные промежутки времени изменения сущности. Он используется в тех случаях, когда база

данных хранит определённые сущности с определённой структурой (и определённым конвертером), но при изменении кодовой базы, модель данных сущности изменяется, например, добавляется дополнительная информация. При этом при считывании данных из базы, хранящаяся сущность должна конвертироваться к новому формату данных, если она не равна текущей версии; и при записи сущностей, они должны записываться в последнем актуальном формате.

Мигратор содержит версию той сущности, которую он обрабатывает, сериализатор сущности, для преобразования в байты и обратно в модель данных и метод для конвертации данных под разные версии (базовая версия мигратора просто проверяет переданную версию и при совпадении десериализует данные).

Так как модель данных в процессе жизни ПО преобразуется, для этого существуют эволюционные миграторы. Данный мигратор используется для тех сущностей, которые уже были изменены. Метод содержит ссылку на предыдущий мигратор, который является «последним актуальным» и будет заменен на текущий, после регистрации в источнике данных. Также интерфейс содержит метод для преобразования старой модели данных в новую и переопределённый момент конвертации из массива байт, который при не соответствии версий, будет передавать метод конвертации предыдущим миграторам, получать их модель данных и в конечном счёте конвертировать старую модель в новую. Данная структура позволяет создавать сколь угодно много иерархий сущностей и очень быстро преобразовывать их, а также система миграторов позволяет не перезаписывать данные в базе данных, после обновления модели. Это приводит к уменьшению времени обмена информацией между различными МИС.

В качестве примера рассмотрим сохранение и поиск сущности пациента по ключу. Первоначально необходимо создать мигратор для сущности пациента (рисунок 10).

```
public class PatientUpdateMigrator : EntityMigrator<PatientUpdateFm>
{
    1 reference
    public override int Version => 0;
    10 references
    public override ISerializer<PatientUpdateFm> CurrentSerializer => DefaultSerializer<PatientUpdateFm>.Instance;
}
```

Рис. 10. Мигратор сущности пациента

Мигратор указывает версию 0 и сериализатор по умолчанию. Далее создаётся контекст данных, который наследуется от источника данных (рисунок 11).

```
public class DataContext : DataSource
{
    1 reference
    public DataContext(DirectoryInfo heapDirectory, long maxHeapSize) : base(heapDirectory, maxHeapSize)
    {
        PatientUpdateFlow = InitializeDataSet(dataSetId: 1, entityMigrator: new PatientUpdateMigrator());
    }
    2 references
    public DataSet<PatientUpdateFm> PatientUpdateFlow { get; }
}
```

Рис. 11. Контекст данных

Контекст данных содержит `DataSet<PatientUpdateFm>` - источник данных к сущностям пациента, который регистрируется в конструкторе контекста данных методом `InitializeDataSet`, который первым параметром принимает идентификатор источника данных (по которому в метаданных будет соотнесён нужный `DataSetMetaPointer` - из структуры источника данных) и вторым параметром - последний актуальный мигратор.

Инициализируются модель данных пациента и контекст данных (рисунок 12). Метод `PatientUpdateFlow.Add` пытается добавить модель данных пациента в базу данных и возвращает флаг при добавлении пациента. Следующей строчкой осуществляется поиск в базе данных пациента по идентификатору 0, если пациент был успешно найден, то переменная `wasFound` будет равна `true`, а модель данных `readInstance` будет иметь считанный экземпляр данных из базы данных.

```

PatientUpdateFm patient = new PatientUpdateFm
{
    Id = 0,
    Name = "ИмяПациента",
    Surname = "ФамилияПациента",
    Patronymic = "ОтчествоПациента",
    BirthDate = new DateTime(year: 1993, month: 2, day: 1),
    Gender = 1,
    Phone = "+79992299229",
    LegalRepresentative = new LegalRepresentativeUpdateFm
    {
        Id = 34,
        RelationshipTypeId = 2
    }
};

DataContext dataContext = new DataContext(DatabaseDirectoryInfo, MaxHeapSize);
bool wasAdded = dataContext.PatientUpdateFlow.Add(patient.Id, patient);

bool wasFound = dataContext.PatientUpdateFlow.TryGetById(0, out PatientUpdateFm readInstance);

```

Рис. 12. Использование контекста данных

Данная реализация позволяет значительно ускорить доступ к данным, весь исполняющий код находится непосредственно в одной программе с обработкой этих данных (не выделена СУБД в отдельное приложение), сам доступ к данным осуществлён с минимальными абстракциями, что также осуществляет скорость доступа.

ЗАКЛЮЧЕНИЕ

Архитектура системы Астрал.Мед разрабатывается под гибкую интеграцию с другими МИС. В качестве перспективы можно выделить реализацию экспертной системы для помощи принятия решений врачам, включая внедрение искусственных нейронных сетей.

Внедрение системы Астрал.Мед позволит значительно ускорить доступ к основным медицинским и персональным данным пациента за счет того, что весь исполняющий код находится непосредственно в одной программе с обработкой этих данных (не выделена СУБД в отдельное приложение), сам доступ к данным осуществлён с минимальными абстракциями, что также увеличивает скорость доступа. В ближайшее время система может быть модернизирована и расширена за счет добавления

связей между сущностями, индексов, кеширования и т.п., что позволит решать более общие задачи.

Система позволит сократить время оказания медицинской помощи пациентам. При этом повысится качество оказания медицинских услуг в соответствии с порядками оказания медицинской помощи и на основе клинических рекомендаций.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Справочная информация: "Стандарты и порядки оказания медицинской помощи, клинические рекомендации" [Электронный ресурс] // Веб-узел КонсультантПлюс. – 2021 – Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_141711/.
2. Лапынин Н. В. Комплексная система автоматизации деятельности медицинского учреждения [Текст] / Лапынин Н. В. – М., 2019 – 5 с.
3. События [Электронный ресурс] // Веб-узел SofTrust. – 2021 – Режим доступа: <https://www.softtrust.ru/events/novosti-kompanii/>.
4. Мурашко рассказал на заседании СФ о мерах по повышению устойчивости системы здравоохранения к новым вызовам [Электронный ресурс] // Веб-узел СОВЕТ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОГО СОБРАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ. – 2021 – Режим доступа: <http://council.gov.ru/events/news/123233/>.
5. Компания К-МИС. Медицинские информационные системы. Комплексная автоматизация медицинских организаций [Электронный ресурс] // Веб-узел КМИС. – 2021 – Режим доступа: <https://www.kmis.ru/>.
6. ТрастМед:МИС [Электронный ресурс] // Веб-узел ТрастМед. – 2021 – Режим доступа: <https://www.softtrust.ru/products/trustmed-mis/>.
7. Единая цифровая платформа для здравоохранения региона [Электронный ресурс] // Веб-узел РТЛабс. – 2021 – Режим доступа: <https://www.rtlabs.ru/projects/regionalnaya-meditsinskaya-informatsionnaya-sistema-rmis/>.
8. Новости – Правительство России [Электронный ресурс] // Веб-узел Правительство России. – 2021 – Режим доступа: <http://government.ru/news/41980/>.
9. Иринархов М. Г. Устранение неопределенности в процессах интеграции неоднородных информационных систем журнал [Текст] / М. Г. Иринархов, К. И. Квятковский // // Вестник Астраханского государственного технического университета. — 2012. — № 1. — С. 118-123.
10. Иванюгин М.А. Интеграция информационных систем [Текст] / М.А. Иванюгин // // Международный журнал гуманитарных и естественных наук. — 2020. — С. 80-83.
11. Сафронова М.А. Проектирование подсистемы интеграции данных из системы "БОСС-кадровик" в систему "1 с: Предприятие" [Текст] / М.А. Сафронова // // Известия Тульского государственного университета. Технические науки. — 2018. — С. 48-51.

Чувилькин Алексей Александрович – студент кафедры «Информационные системы и сети», Московский государственный технический университет им. Н.Э. Баумана (филиал), КФ МГТУ, тел. +79992289833, e-mail: leha.chuvilkin.90@mail.ru.

Бояровская Алена Валентиновна – студент кафедры «Информационные системы и сети», Московский государственный технический университет им. Н.Э. Баумана (филиал), КФ МГТУ, тел. +79005772305, e-mail: egorova230598@mail.ru.

Борсук Наталья Александровна – к.т.н., доцент кафедры «Информационные системы и сети», Московский государственный технический университет им. Н.Э. Баумана (филиал), КФ МГТУ, тел. +79109102799, e-mail: borruk.65@yandex.ru.

ANALYSIS OF THE DEVELOPMENT OF A MEDICAL INFORMATION SYSTEM

A.A. Chuvilkin, A.V. Boyarovskaya, N.A. Borsuk
Moscow State Technical University named after N.E. Bauman, Kaluga

Abstract – Protection of the health of citizens is one of the key topics in the development of the state. The need to build a health care system on a new technological base was announced by the President of the Russian Federation Vladimir Putin in his Address to the Federal Assembly on April 21, 2021. Health care is a rather complex and dynamic system that requires constant assessment of the situation and quick decision-making. A continuous assessment of the epidemiological situation and monitoring of the health of a particular citizen is impossible without a well-organized exchange of information and a unified data system introduced into the daily work of medical institutions. At the moment, there are a large number of medical information systems (MIS), however, many of them contain redundant functionality and a specific data storage format. It is quite problematic to adapt existing IIAs to the needs of specific regions, both in terms of labor and time costs, and financially. The task was to develop a medical information system Astral.Med, aimed at solving urgent problems in the field of primary health care. One of the development tasks is the possibility of transferring and receiving various medical information to the already existing MIS with the aim, ultimately, of sending this data to the Unified State Information System in the field of health care. In this case, a single repository of all patient data should be implemented, regardless of the medical institution in which they are served and the information system in this institution. Implementation of the Ast-Ral.Med system consists of: development of the interface module; developing a database and a way to interact with it; a module for processing requests from specialists of a medical institution and a module for integration with third-party MIS (which is one of the most basic and complex tasks due to the possible incompatibility of data storage formats in various information systems).

Index terms: MIS, message broker, data model, data serialization and deserialization, stored heap, data source, entity migrator.

REFERENCES

1. Reference information: "Standards and procedures for the provision of medical care, clinical guidelines" [Electronic resource] Access mode: http://www.consultant.ru/document/cons_doc_LAW_141711/.
2. Lapynin, N. V. Complex system of automation of the activity of a medical institution. Moscow: 2019.
3. Events [Electronic resource] Access mode: <https://www.softtrust.ru/events/novosti-kompanii/>.
4. FEDERATION COUNCIL OF THE FEDERAL ASSEMBLY OF THE RUSSIAN FEDERATION [Electronic resource] Access mode: <http://council.gov.ru/events/news/123233/>.
5. K-MIS company. Medical information systems. Complex automation of medical organizations [Electronic resource] Access mode: <https://www.kmis.ru/>.
6. TrustMed: MIS [Electronic resource] Access mode: <https://www.softtrust.ru/products/trustmed-mis/>.
7. A single digital platform for healthcare in the region [Electronic resource] Access mode: <https://www.rtlabs.ru/projects/regionalnaya-meditsinskaya-informatsionnaya-sistema-rmis/>.
8. News - Russian Government [Electronic resource] Access mode: <http://government.ru/news/41980/>.
9. Irinarkhov M.G. and Kvyatkovsky K.I. "Elimination of uncertainty in the integration processes of heterogeneous information systems magazine", Bulletin of the Astrakhan State Technical University, No. 1, pp. 118-123, 2012
10. Ivanyugin M.A. "Integration of information systems", International Journal of Humanitarian and Natural Sciences, pp. 80-83, 2020
11. Safronova M.A. "Designing a subsystem for integrating data from the "BOSS-personnel officer" system into the "IC: Enterprise" system", Bulletin of the Tula State University. Technical science, pp. 48-51, 2018

Chuvilkin Aleksey Aleksandrovich – student of the department "information systems and networks", Moscow State Technical University named after N.E. Bauman (branch), +79992289833, e-mail: leha.chuvilkin.90@mail.ru.

Boyarovskaya Alyona Valentinovna – student of the department "information systems and networks", Moscow State Technical University named after N.E. Bauman (branch), +79005772305, e-mail: egorova230598@mail.ru.

Borsuk Natalia Alexandrovna – Ph.D., associate professor of the department "information systems and networks", Moscow State Technical University named after N.E. Bauman (branch), +79109102799, e-mail: borsuk.65@yandex.ru