

УДК 004.043

А.Б. ЛЕЩЕНКО, А.В. ГУК, Ю.А. ЛЕЩЕНКО

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина

ИНТЕГРАЦИОННАЯ СИСТЕМА НА ОСНОВЕ ЭЛЕКТРОННОГО ПРОТОКОЛА ОБМЕНА МЕДИЦИНСКИМИ ДАННЫМИ HEALTH LEVEL 7

Предлагается подход интеграции медицинских данных в единое хранилище. Используется модифицированный метод консолидации данных для обеспечения постоянного обмена данными между системами. Разработано приложение, способное объединить медицинские, страховые, информационные системы и обеспечить обмен сообщениями между ними посредством SOA (протокола обмена структурированными сообщениями), предоставить возможность использования и управления медицинской информацией ее владельцами. В качестве основного стандарта обмена медицинскими данными используется Health level 7. Рассмотрен сервис-ориентированный подход к интеграции медицинских, страховых и других информационных систем, способный принимать и обрабатывать сообщения, описанные в стандарте HL 7.

Ключевые слова: лечебно-профилактическое учреждение, медицинская информационная система, интеграция, Health level 7, web-сервисы.

Введение

Реформа в здравоохранении Украины предполагает ликвидацию разветвлённых сетей районных больниц и предполагает деление территории страны на отдельные госпитальные округа [1]. Вследствие чего возникла необходимость свободного доступа врачей к истории болезни пациента из любых государственных или частных поликлиник, больниц и диагностических центров, куда планирует обратиться гражданин. Для обеспечения эффективной работы лечебных заведений требуется создание технологических решений, на основе которых можно построить единое интегрированное пространство (ЕИП) [2] и механизмы обмена медицинскими данными в достаточно широких рамках – от ведомственного уровня до национального, что делает тему предлагаемой публикации достаточно актуальной.

Анализ проводимых исследований и постановка задачи

При поиске решений повышения эффективности комплексной обработки данных, находящихся в различных компьютерных системах, требуется ответить на вопрос - каким способом будет осуществляться интеграция существующих и все увеличивающихся информационных ресурсов в единое информационное пространство?

Дополнительно задача усложняется при внесении изменений в уже существующие внедренные компьютерные системы обработки данных, т.е. если используемые системы были разработаны сотруд-

никами предприятия под конкретные, сравнительно небольшие задачи, а документации и исходных кодов работающей системы не осталось. Аналогичная ситуация может возникнуть если система была куплена у третьей стороны («коробочное решение»), т.е. в виде готового инструмента без возможности внесения серьезных изменений и затраты на доработку системы под нужды конкретного заказчика превышают соответствующую статью бюджета IT-подразделения.

Организационно решение задачи осложняет ситуация, когда разные компьютерные системы обработки данных находятся в сравнительно независимых структурных подразделениях или организациях. Например, такая ситуация возникает при слиянии нескольких компаний, каждая из которых оставляет за собой некоторый суверенитет. Аналогично выглядит ситуация в государственном секторе, когда несколько ведомств или министерств имеют свои наработки в IT-сфере.

Следует отметить, что задача создания интегрированного информационного пространства не может быть решена статически. Компьютерные системы обработки данных появляются (внедряются, дорабатываются) и исчезают (выводятся из эксплуатации), а интегрирующая система должна позволить добавлять (изменять) подключение любой компьютерной системы обработки данных к единому пространству, а также убирать ее замещением другой системы или простым отключением источника данных.

Целью интеграции данных является получение единой и цельной картины корпоративных бизнес-

данных [3, 4].

Проведенный анализ показал, что существуют три основных подхода к интеграции данных [4]: консолидация; федерализация; распространение.

При использовании подхода консолидации данные собираются из нескольких первичных систем и интегрируются в одно постоянное хранилище. Хранилище, в последствии, может быть использовано для проведения анализа, подготовки отчетности, а также являться источником данных для других приложений (рис. 1).

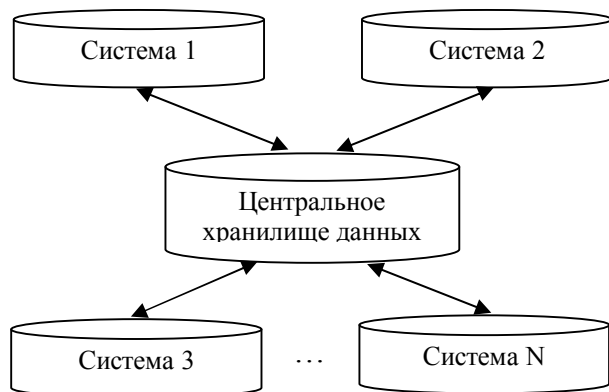


Рис. 1. Концептуальная схема подхода консолидации данных

Подход характеризуется задержками в актуализации данных хранилища. Данные обновляются с определенной периодичностью, которая может составлять как нескольких секунд, так и несколько часов, дней или месяцев, в зависимости от потребностей информационной среды.

Преимуществами метода консолидации данных являются:

1) возможность работы в режиме без постоянного подключения источников к централизованному хранилищу данных (так называемый «offline» или автономный режим);

2) возможность проведения любых видов аналитического анализа над интегрированными (консолидированными) данными;

3) возможность обработки значительных (больших) объемов данных с большим количеством всевозможных преобразований и изменений в структурах и форматах хранения данных.

Недостатками метода являются:

1) значительные вычислительные мощности и прочие накладные расходы, затрачиваемые на преобразование больших объемов данных;

2) данные консолидируются, но не передаются между различными системами;

3) высокие требования к дисковому пространству для накопления информации в центральном хранилище;

4) данные в централизованном хранилище за-

частую не отражают последние изменения в данных источников в связи с задержкой в обновлении информации.

При использовании метода федерализации данных [5] образуется единое виртуальное информационное пространство, данные в котором могут содержаться в различных источниках, а информация о расположении данных недоступна запрашивающей стороне. По получению запроса на получение данных специальная система создает «подзапросы» к источникам данных, которые содержат свои части общих данных и формирует единый ответ, интегрируя ответы от источников. В центре такой интегрирующей информационной системы размещается координирующее звено, в котором хранится информация о том, какие данные следует запрашивать и в какой системе-источнике их необходимо запрашивать (рис. 2).

Преимуществами федерализации данных являются:

1) сравнительно низкие вычислительные расходы в центральном узле системы (центральный узел является только координатором, распространяющим запросы системам-источникам, и собирающим воедино полученные ответы);

2) низкие требования к хранилищу данных, поскольку, данные из систем-источников в общем случае не скапливаются в некоем «центре» информационного пространства, исчезает требование к большому объему дискового пространства и требование к управлению такими данными;

3) актуальность информации (данные, получаемые от систем - источников, всегда самые «свежие», при условии постоянной доступности всех систем, участвующих в интеграции);

4) возможность разработать динамически расширяемое информационное пространство, к которому будут временно подключаться различные источники данных.

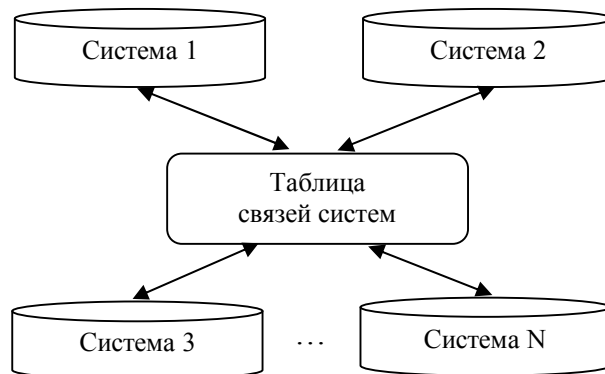


Рис. 2. Концептуальная схема подхода федерализации данных

Недостатками метода являются.

1. Высокая нагрузка на системы-источники и каналы связи. Нагрузка также возрастает, если ставится задача проведения анализа сводных данных, получение больших объемов информации от систем-источников, что может серьезно нагрузить как сервера систем-источников, так и линии связи.

2. Необходимость разработки интеллектуальных модулей в каждой информационной системе-источнике. Модуль должен понимать общепринятый протокол обмена данными систем в информационном пространстве и позволять преобразовывать получаемые запросы во внутренние запросы информационной системы-источника. При изменении структур данных в системе нужно менять и сами модули и структуры в центральном звене.

3. Необходимость постоянного подключения всех систем, так как данные не консолидируются для длительного хранения, получение их по запросу возможно только при наличии ответа от системы-источника.

Метод распространения данных, при котором осуществляется перенос (копирование) информации из одной системы в другую по определенным событиям (рис. 3).

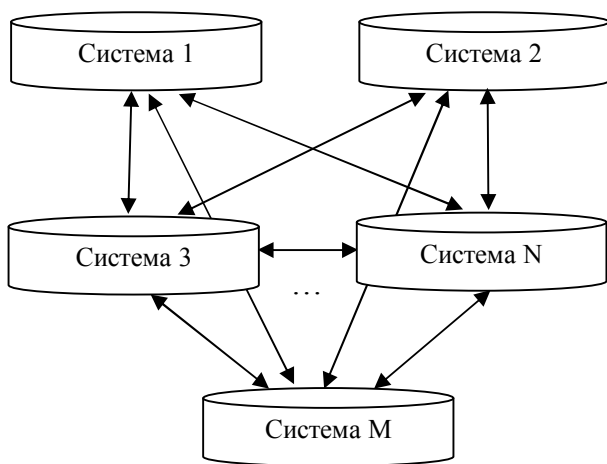


Рис. 3. Концептуальная схема подхода распространения данных

В качестве одного из решений распространения данных используется корпоративная шина передачи данных, к которой через адаптеры присоединяются различные компьютерные системы обработки данных (рис 4).

Отличительный признак метода - данные доставляются от системы-отправителя в систему-получатель гарантированно, независимо от методов, способов и форматов передачи данных. Примерами технологий, поддерживающих распространение данных, являются - интеграция корпоративных при-

ложений EAI (Enterprise application integration) и тиражирование корпоративных данных EDR (Enterprise data replication).

Можно выделить основные преимущества метода распространения данных.

1. Сравнительно низкие расходы на создание и поддержание (сопровождение) центрального элемента интегрирующей системы.

2. Данные перемещаются в режиме реального времени или близком к нему.

Недостатки метода.

1. Наибольшие накладные расходы на поддержание связности информационных систем в случае внесения изменений в сами системы.

2. Невозможность получить все данные консолидировано без применения дополнительной системы, которая будет собирать данные.

3. Вырождение метода во многом в неоправданно усложненную федерализацию в случае, если допускаются запросы от одной информационной системы к другой информационной системе.

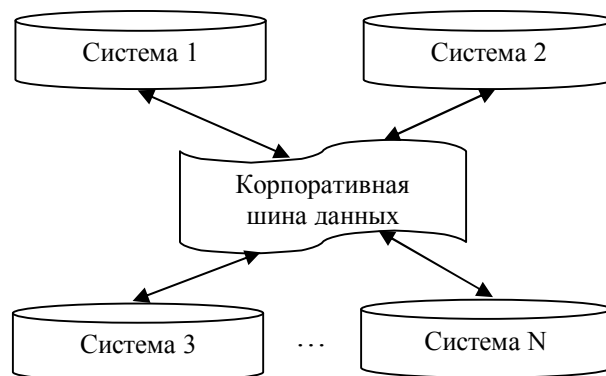


Рис. 4. Концептуальная схема корпоративной шины передачи данных

В данной работе, учитывая преимущества и недостатки рассмотренных распространенных методов решения задачи интеграции данных компьютерных систем, применяется метод консолидации данных для обеспечения постоянного обмена данными между системами.

С конца 90-х годов XX века, де-факто, стандартом интеграции данных признан подход сервис-ориентированной архитектуры (Service-Oriented Architecture (SOA)).

SOA – технология разработки распределенных систем, функциональность которых обеспечивается с помощью сервисов (служб). Службы взаимодействуют между собой посредством сообщений и реализуют бизнес-функции и правила, описанные контрактом (интерфейсом). W3C определяет SOA как набор компонент, которые могут быть вызваны, а их описание может быть опубликовано и найдено.

Службы предоставляют доступ к данным, бизнес-процессам и другим информационным системам. Вызов служб может быть как синхронным, так и асинхронным. SOA системы обычно разрабатываются на основе web-служб и могут быть использованы в гетерогенных системах, состоящих из большого числа приложений, работающих на различных платформах и созданных с помощью разных языков программирования. SOA система может быть построена и на применении COM/DCOM, EJB, CORBA. Преимуществом web-служб является то, что они основаны на открытых стандартах технологий (XML (eXtensible Markup Language), SOAP (Simple Object Access Protocol), UDDI (Universal Description Discovery & Integration), WS-* (web service)), которые реализованы на различных платформах. Интерфейс компонентов SOA-программы предоставляет инкапсуляцию деталей реализации конкретного компонента (ОС, платформы, языка программирования, вендора, и т. п.) от остальных компонентов. Таким образом, SOA предоставляет гибкий и элегантный способ комбинирования и многократного использования компонентов для построения сложных распределённых программных комплексов [6].

SOA хорошо зарекомендовала себя для построения крупных корпоративных программных приложений. Целый ряд разработчиков и интеграторов предлагают инструменты и решения на основе SOA (например, платформы IBM WebSphere, Oracle/BEA, Microsoft Windows Communication Foundation).

Подход позволяет организовать обмен данными между компьютерными системами на основе распространенных протоколов и форматов: HTTP (Hypertext Transfer Protocol), HTTPS (Hypertext Transfer Protocol Secure), XML и др. В SOA формируется специальная среда обмена данными между системами на основе запросов одной системы к другой. SOA может быть особенно полезно, если речь идет о каких-либо унифицированных сервисах (например, каждая авиакомпания может создать интерфейс для взаимодействия с ее компьютерной системой на бронирование билетов). Тогда приложение, которое консолидирует данные о наличии авиабилетов, обратится через SOA к нескольким системам (которые ответят) и получит данные от каждой.

В самом общем виде SOA предполагает наличие трех основных участников: поставщика сервиса, потребителя сервиса и реестра сервисов (рис. 5). Взаимодействие участников выглядит достаточно просто: поставщик сервиса регистрирует свои сервисы в реестре, а потребитель обращается к реестру с запросом.

Для использования сервиса необходимо следовать соглашению об интерфейсе для обращения к сервису - интерфейс должен не зависеть от платформы. SOA реализует масштабируемость сервисов - возможность добавления сервисов, а также их модернизацию. Поставщик сервиса и его потребитель оказываются несвязанными - они общаются с помощью сообщений. Поскольку интерфейс должен не зависеть от платформы, то и технология, используемая для определения сообщений, также должна не зависеть от платформы. Поэтому, как правило, сообщения являются XML-документами, которые соответствуют XML-схеме [7].

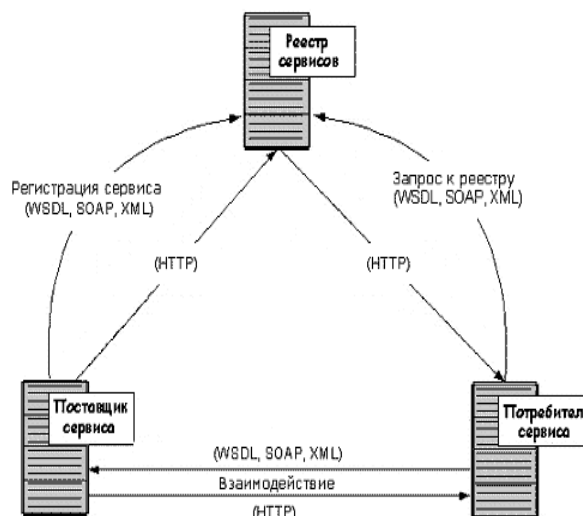


Рис. 5. Общая схема SOA

Учитывая мировой положительный опыт применимости SOA для построения распределенных систем, а также наличие готовых библиотек для реализации веб-сервисов на практически любой платформе, открытость SOA, протоколов и использование языка XML для описания данных, предлагается использовать SOA для обеспечения транспортного уровня обмена данными между системами.

Основным моментом, определяющим успешность решения интеграции, становится стандартизация. Для разработчиков очень важно не оказаться лицом к лицу с проблемами поддержания несовместимых структур коммуникации и выполнения транзакций. Во избежание такой ситуации необходимо создать базу, на основе которой можно минимизировать несовместимость и максимизировать однотипность обмена информацией между системами.

В качестве основного стандарта обмена медицинскими данными, при разработке системы, был выбран HL7. Стандарт HL7 (рис. 6.) используется как метаструктура, обеспечивающая единство спецификаций и методологии при разработке сетевых систем. Поддержка стандарта HL7 позволяет обеспечить обмен клинической, финансовой и

```

- <ADT_A03>
  - <MSH>
    <MSH.1>|</MSH.1>
    <MSH.2>^~\&</MSH.2>
    <MSH.3>102030</MSH.3>
    <MSH.5>GLOBALHEALTH</MSH.5>
    <MSH.6>PlaceServer</MSH.6>
    <MSH.7>20111117023612</MSH.7>
  - <MSH.9>
    <UNKN.1>ADT</UNKN.1>
    <UNKN.2>A03</UNKN.2>
  </MSH.9>
  <MSH.10>740</MSH.10>
  <MSH.11>P</MSH.11>
  <MSH.13>3</MSH.13>
</MSH>
- <EVN>
  <EVN.1>A03</EVN.1>
  <EVN.2>20111117023612</EVN.2>
  <EVN.4>03</EVN.4>
  <EVN.5>123</EVN.5>
</EVN>
- <PID>
  <PID.1>20010</PID.1>
  <PID.2>123</PID.2>
  <PID.3>46</PID.3>
  - <PID.5>
    <UNKN.1>Ковалев</UNKN.1>
    <UNKN.2>Петр</UNKN.2>
    <UNKN.3>Иванович</UNKN.3>
  </PID.5>
  <PID.8>F</PID.8>
  <PID.13>Домашній, 215-25-23</PID.13>
  <PID.14>Харьков</PID.14>
  <PID.15>ул.Ленина</PID.15>
  <PID.16>д.25</PID.16>
  <PID.17>кв.158</PID.17>
  <PID.25>12540</PID.25>
  <PID.26>Україна</PID.26>
</PID>
- <PV1>
  <PV1.2>госпитализирован</PV1.2>
  - <PV1.3>
    <UNKN.1>Терапевтичний №1</UNKN.1>
    <UNKN.2>Палата №5</UNKN.2>
    <UNKN.3>Койка № 7</UNKN.3>
  </PV1.3>
  <PV1.4>1</PV1.4>
  <PV1.7>Федоров В.И.</PV1.7>

```

Рис. 6 Структура XML файла в стандарте HL 7 сообщения (выписка пациента (код события A03))

административной информацией между информационными системами ЛПУ, страховых компаний и интеграционной платформы [8].

Формальное представление сообщения в виде HL7 выглядит так:

$$\text{MessageHL7} = \langle T, E, S \rangle,$$

где T – тип сообщения, $T = \{ADT, BAR, \dots\}$;

ADT (Admission, Discharge, Transfer) – госпитализация, выписка, перевод физического лица;

BAR (Billing Account Record) – добавление (изменение) счета физического лица;

E = {A01...A37} – тип события (A01 – Госпитализация пациента; A02 – Перевод пациента; ... A37 – Разъединение сведений о пациенте);

S – набор сегментов, $S = \{C, M\}$;

C – управляющие сегменты сообщений:

$$C = \{MSH, MSA, ERR, \dots\},$$

где MSH – сегмент заголовка сообщения определяет назначение, источник, приемник сообщения и определенную специфику его синтаксиса;

MSA – сегмент подтверждения сообщения содержит информацию, которая передается в случае, когда данное сообщение используется для подтверждения другого сообщения;

ERR – сегмент ошибки используется для того, чтобы добавить к подтверждению комментарий по поводу возникшей ошибки.

M – сегменты сообщений:

$$M = \{M1, [M2], \{M3\}\},$$

где M1 – обязательные сегменты;

[M2] – необязательные сегменты;

{M3} – повторяющиеся сегменты;

$$M = \{PID, PV1, [PV2], [\{OBX\}], \dots\},$$

где PID – идентификация пациента (сегмент используется во всех приложениях как основное средство обмена информацией, идентифицирующей пациента);

PV1 – визит пациента (сегмент используется приложениями при обмене специфичной информацией о визите пациента);

PV2 – дополнительные сведения о визите пациента (в сегменте передается информация, дополняющая сведения о пациенте, включенные в сегмент PV1);

OBX – результаты медицинских исследований.

Сегменты сообщений M состоят из набора полей и разделителей:

$$M = \langle P, Z \rangle,$$

где P – поля, $P = \{p1, p2, p3, p4, p5, p6, p7, p8\}$ (p1 – порядковый номер поля данных в сегменте; p2 – уникальное имя поля; p3 – идентификатор поля данных в тексте; p4 – максимальное число символов в сегменте; p5 – признак того, является ли поле данных обязательным; p6 – признак того, может ли поле повторяться в сегменте; p7 – стандарт HL7 определяет для этого поля таблицу значений; p8 – ограничение на содержание поля данных (тип данных)). Количество полей в каждом сегменте зависит от типа сегмента (ADT, BAR, ...);

Z – разделители сообщения, $Z = \{ |, ^, \sim, \&, \backslash \}$ (| – разделяет два смежных поля данных сегмента; ^ – разделяет смежные компоненты полей данных (если таковые имеются); & – разделяет смежные субкомпоненты полей данных, когда таковые встречаются; ~ – разделяет повторы полей данных, если таковые имеются; \ – управляющий символ (Escape), используемый в текстовых полях.

Количество сегментов в каждом сообщении зависит от типа сообщения (ADT, BAR, ...). В каждом управляющем сегменте C содержится набор сегментов сообщений M.

Анализ существующих общепринятых подходов к решению задач интеграции данных [2 – 4, 9] выделяет особенности, преимущества и недостатки применяемых архитектур интегрирующих систем, что позволяет сделать выводы о пригодности существующих подходов для решения задачи интеграции.

Рассмотрение существующих предлагаемых на рынке готовых решений [10 – 15] от мировых лидеров по разработке и поставкам программного обеспечения позволяет указать на те преимущества, которые следует развивать и использовать при построении системы, а также те недостатки, которых следует избегать.

На основании проведенного анализа была сформирована цель и постановка задачи работы.

Целью работы является повышение эффективности процессов комплексной обработки медицинских данных, находящихся в разных компьютерных системах, в различных структурах и форматах представления, на основе разработки методов интеграции медицинских данных в едином интегрированном информационном пространстве.

Для осуществления цели были решены следующие задачи:

- 1) разработать архитектуру и общие требования к интеграции данных разнородных компьютерных систем;
- 2) создать модели и методы представления и обработки данных компьютерных систем для обеспечения интеграции и организации централизованного хранилища;
- 3) построить алгоритмы синхронизации и сохранения данных в централизованном хранилище;
- 4) создать прототип программного обеспечения, реализующего предлагаемые методы и алгоритмы в рамках описанных моделей, с последующим проведением анализа эффективности предлагаемого подхода.

Решение поставленных задач

Как уже было сказано выше, одним из наиболее распространенных и часто применяемых, при создании единого интегрированного информационного пространства, является подход для консолидации данных, который достаточно прост при объединении данных из компьютерных систем. Подход предусматривает построение единого интегрированного информационного пространства с центральной точкой - отдельной компьютерной системой (или несколькими компьютерными системами), которая берет на себя функции управления процессами интеграции и обмена данными между системами, а также связывает компьютерные системы между собой благодаря веб-сервисам. При этом в

интеграционной платформе формируются и располагаются не только логические связи преобразования данных, но и сами данные.

Детализированная концептуальная схема единого интегрированного информационного пространства, представлена на рис. 7.

Центральным элементом схемы является интеграционная платформа. Как уже было отмечено выше, она состоит из центрального хранилища данных, а также из программных модулей, которые обеспечивают взаимодействие с компьютерными системами - источниками, обработку хранящихся в центральном хранилище данных и предоставление данных пользователям.

На схеме обозначено, что каждая компьютерная система подключается посредством протокола обмена структурированными сообщениями SOAP, используя транспортный протокол HTTP. Такое решение обеспечивает быстроту подключения новых компьютерных систем к единому интегрированному информационному пространству.

Предлагаемая архитектура, в отличие от консолидации данных, позволяет не только свести данные из различных источников в едином центре (что позволит производить комплексную обработку данных), но и обеспечивает обмен данными между системами под контролем центрального звена, очистки данных и применение механизмов их синхронизации.

Предлагаем следующую математическую модель единого интегрирующего информационного пространства, которая создает основу для понимания интегрирующих процессов.

Пусть имеется множество $M = \{m_1, m_2, \dots, m_y\}$ компьютерных систем обработки данных, являющихся источниками (хранилищами) профильных данных определенной области знаний.

Пусть также имеется множество $N = \{n_1, n_2, \dots, n_u\}$ компьютерных систем обработки данных, которым для функционирования необходимо получение информации из внешних источников, которыми могут являться компьютерные системы множества M .

При этом множества M и N могут пересекаться: так, одна и та же компьютерная система может одновременно являться и источником и потребителем данных.

Для упрощения назовем компьютерные системы множества M компьютерными системами-источниками или просто источниками, а компьютерные системы множества N – компьютерными системами-потребителями или просто потребителями интегрированного информационного пространства.

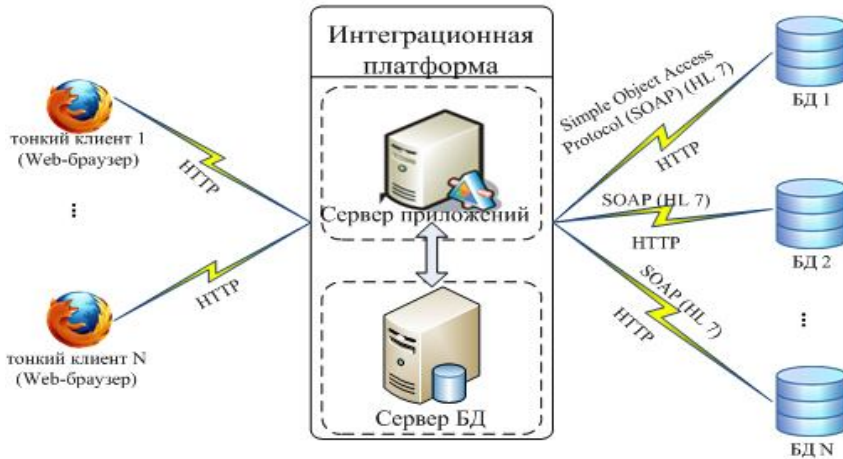


Рис. 7. Концептуальная схема единого информационного пространства

Очевидно, что в общем случае могут отсутствовать прямые связи между компьютерными системами из различных множеств и внутри одного из множеств. Введем множество узлов передачи данных $P = \{p_1, p_2, \dots, p_i\}$ между компьютерными системами.

Для описания связей между компьютерными системами и другими элементами (узлами передачи данных) добавим в модель множество ориентированных дуг (связей) $L = \{l_1, l_2, \dots, l_x\}$, соединяющих, в одном из направлений, между собой, компьютерную систему из множества M и узел передачи данных множества P или узел множества P и компьютерную систему из множества N . Каждому элементу $l_j \in L, j = \overline{1, x}$ соответствует один и только один элемент $k_j \in K = \{k_1, k_2, \dots, k_x\}$, который описывает состав (набор) данных, передаваемых по данной дуге (связи). При этом в каждый элемент $p_j \in P$ входит одна и только одна связь $l_{j1} \in L$ (и соответствующие $k_{j1} \in K$) и выходит одна и только одна связь $l_{j2} \in L$ (и соответствующие $k_{j2} \in K$).

Все дополнительные ограничения задаются в виде множества $F = \{f_1, f_2, \dots, f_z\}$, описывающего условия на передачу данных между компьютерными системами, временные условия передачи данных и т.п. В каждом узле преобразования $p_j \in P, j = \overline{1, l}$ выполняется преобразование $h_j \in H = \{h_1, h_2, \dots, h_s\}$ входящих данных в исходящие. Иными словами

$$k_{j2} \in K = h_j(k_{j1} \in K), \forall k \in K.$$

Совокупность множеств $\langle M, N, P, L, K, F, H \rangle$ образует единое интегрированное информационное пространство, отображенное на приведенной схеме (рис. 8).

Зададим дополнительные ограничения и особые условия.

1. Элементы множества L (связи) могут быть временными (частный случай - периодическими). Таким образом, кроме возможности постоянной передачи данных существует также возможность того, что та или иная связь l_i существует только определенные промежутки времени (например, 1 раз в сутки на протяжении часа).

2. Данные, входящие в элемент p_j множества P по связи l_{j1} могут иметь структуру

отличную от структуры данных, выходящих из элемента p_j множества P по связи l_{j2} .

3. Один и тот же элемент множества M может быть соединен произвольным количеством связей с элементами множества P .

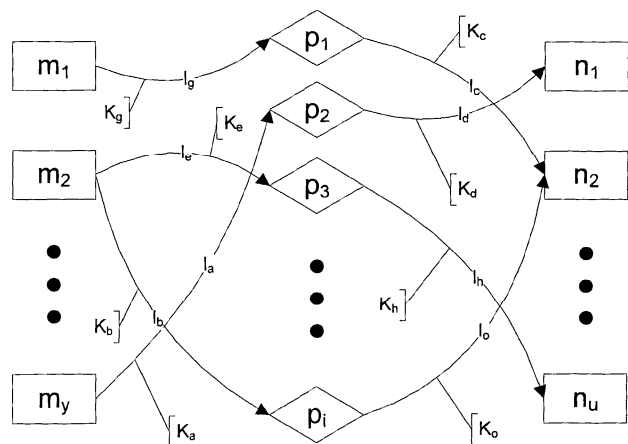


Рис. 8. Модель единого интегрированного пространства

4. Один и тот же элемент множества N может быть соединен произвольным количеством связей с элементами множества P .

Введем дополнительные ограничения на единое информационное пространство. Для этого количество элементов множества M возьмем в пределах не менее 3-х, множества N - от 1 информационного пространства до 3-х, а в качестве множества F - описанные выше ограничения и особые условия.

Отдельно выделим элемент t , который входит в оба множества M и N и представляет собой центральный узел - ключевое звено пространства.

Далее обозначим требования к преобразованию данных на узлах множества P : некорректные и неточные данные должны максимально отсеиваться и

не попадать в компьютерную систему-потребителя.

Таким образом, в предлагаемом едином интегрированном информационном пространстве заданы связи и функции преобразования структур данных между системами. Указанная модель описывает единое интегрированное информационное пространство с формальной точки зрения.

С учетом построенной модели предложен общий алгоритм обработки поступившего в хранилище пакета данных для загрузки (рис. 9). На приведенной схеме представлены все основные этапы алгоритма обработки пакета данных, которые поступили для обновления (сохранения). Были рассмотрены случаи, когда в одном пакете данных поступил набор связанных между собой объектов и требуется наведение связей между ними до сохранения в хранилище.

Следует отметить, что в предлагаемой схеме практически линейный порядок обработки данных, а также возможность сохранить только те данные и связи, которые проходят многочисленные проверки. Сами проверки настраиваются администраторами системы (системными аналитиками) и отличаются как для каждого единого интегрированного информационного пространства, так и для каждого реестра (вида объекта).

Рассмотрим этапы алгоритма обработки данных при сохранении их в базе данных ЕИП.

1. Импорт пакета данных. ЕИП получает данные от некоторой информационной системы.

2. Конвертирование XML файла по стандарту HL7. Парсинг XML для получения данных.

3. Автоматическая валидация формата и полноты данных. На данном этапе в качестве фильтра данных применяется специально настраиваемый отдельный модуль, который производит сравнительно простой анализ качества поступающих данных на предмет наличия явных ошибок, которые можно выявить автоматически.

4. Сохранение текущих версий объекта и его связей. Речь идет об использовании системного историзма централизованного хранилища. Данная функция должна выполняться автоматически при изменении централизованного хранилища.

5. Сохранение объектов в централизованном хранилище. Производится сама процедура сохранения объектов.

6. Формирование и отправка обратного пакета с отчетом об ошибках. На данном шаге производится анализ всех произведенных операций. Весь отчет формируется в единый пакет данных и отправляется в информационную систему-источник.

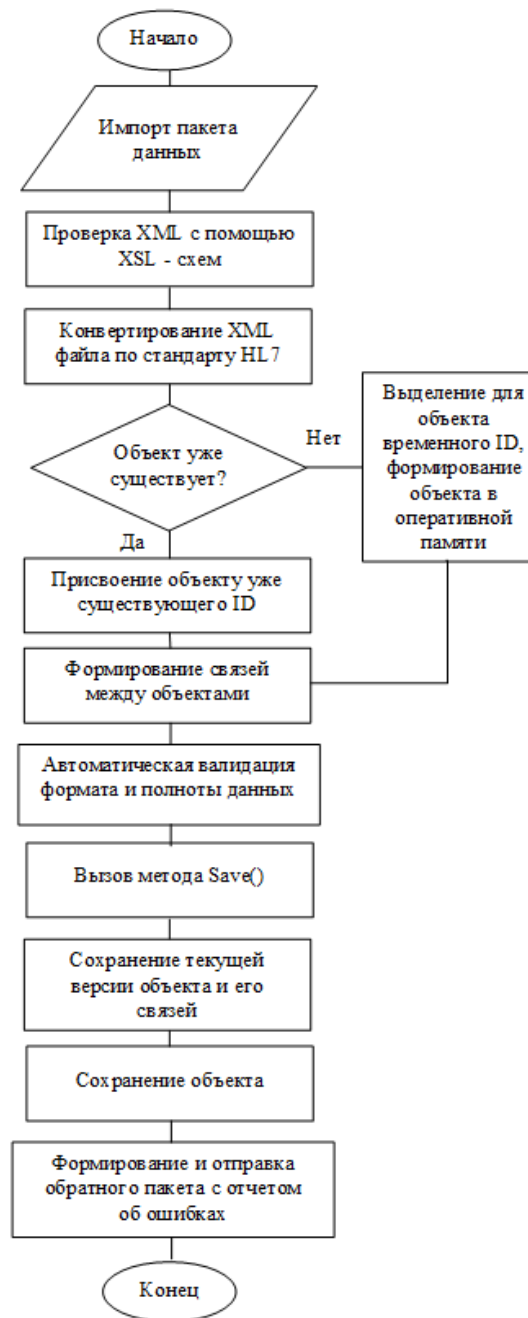


Рис. 9. Общий алгоритм сохранения данных

Программная реализация

Разработка программного продукта является сложным процессом, направленным на реализацию заданных функций. Поэтому для описания функций системы используем стандарты объектно-ориентированного языка UML (Unified Modeling Language) [16-18]. Для создания UML – диаграмм выбран продукт компании Rational. На (рис. 10) показан фрагмент диаграммы классов. Данная диаграмма демонстрирует, как в интеграционной платформе реализована схема хранения медицинской информации пациентов. Основным классом является

ся класс Patient (хранимый класс данных о физ. лице). Вокруг Patient в классах F003 и F025 собрана вся медицинская информация о пациенте. Классы SampleLog и ResultTarget хранят результаты медицинских и лабораторных исследований. Для регистрации в системе физическому лицу необходимо заполнить следующую регистрационную форму (рис. 11).

После одноразовой регистрации в системе пользователь может войти в систему под зарегистрированными логином и паролем. Для этого Пользователю необходимо ввести логин и пароль на форме авторизации в системе. После успешной авторизации пользователь входит в систему под своим

логином и паролем и получает доступ к индивидуальной информации (рис. 12, 13). К общей информации относятся регистрационная информация (ФИО, дата рождения, пол, адрес, ИИН), а под индивидуальной информацией понимается медицинская информация, которая включает в себя данные из амбулаторной карты.

Интеграционный инструмент реализован на базе СУБД Caché. СУБД Caché поддерживает стандарт XML, протокол SOAP и веб-сервисы, что позволяет разрабатывать приложения, обладающие высокой масштабируемостью, производительностью и надежностью.

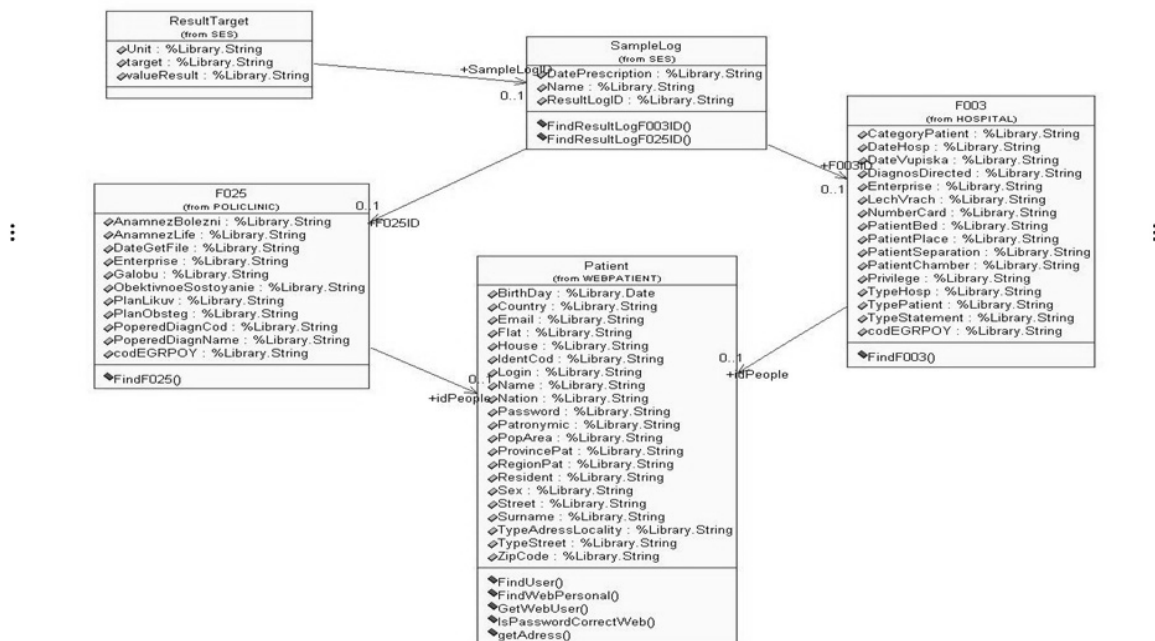


Рис. 10. Фрагмент диаграммы классов

Рис. 11. Форма регистрации пользователей системы

Рис. 12. Регистрационные данные пользователя

| № п/п дослідження | Мета дослідження | Результат дослідження | Одиниця виміру |
|-------------------|------------------|-----------------------|----------------|
| 1 | Са | 1 | ммоль/л |
| 2 | Альбуміни | 2 | г% |
| 3 | Гамма | 3 | - |

Рис. 13. Медицинские данные пользователя

Основными технологиями, которые были использованы при разработке системы, являются технологии СУБД Caché: ZEN Caché, Caché Object Script, а также клиентский язык Java Script. Использование, в качестве сервера, базы данных СУБД Caché позволило применить объектно-ориентированный принцип проектирования информационной системы и использовать различные способы доступа к данным.

Заклучение

В заключение отметим, что в данной публикации предложена архитектура единого интегрированного пространства в основу которой положен подход консолидации данных, математическая модель ЕИП, подсистема интеграции и обмена медицинской информацией. Разработан прототип программного обеспечения, реализующий предложенный подход, метод и алгоритм.

Литература

1. Закон України від 07.07.2011 № 3612-VI "Про порядок проведення реформування системи охорони здоров'я у Вінницькій, Дніпропетровській, Донецькій областях та м. Києві" [Електронний ресурс]. – Режим доступу: <http://www.moz.gov.ua> – 25.07.2012г.

2. Гулиев, Я.И. Концептуальные принципы интегрированной системы управления медицинской помощью и единого информационного пространства [Текст]/ Я.И. Гулиев, В.Л. Малых // Тр. междунар. конф. «Программные системы: теория и приложения», ИПС РАН, Переславль-Залесский, 2006: В 2 т. / Под ред. С.М. Абрамова. - М.: Физматлит, - Т. 1. - С. 27-49.

3. Краюшкин, В. Виртуальная консолидация данных [Электронный ресурс] / В. Краюшкин. – Режим доступа: <http://www.citforum.ru/nets/storage/consolidation/> – 25.07.2012 г.

4. Интеграция данных и хранилища [Электронный ресурс]. – Режим доступа: <http://citcity.ru/12101/> - 25.07.2012 г.

5. Подходы к созданию федеративных Хранилищ данных [Электронный ресурс]. – Режим доступа: <http://zeus.sai.msu.ru:7000/consulting/BI/warehousing/> - 25.07.2012 г.

6. Service Oriented Architecture (SOA) [Электронный ресурс]. – Режим доступа: <http://www.inln.ru/it/soa/> - 3.09.2012 г.

7. Сервис-ориентированная архитектура [Электронный ресурс]. – Режим доступа: <http://citforum.ru/internet/webservice/soa/> - 3.09.2012г.

8. Стандарт "Уровень 7" [Электронный ресурс]. – Режим доступа: http://www.ctmed.ru/DICOM_HL7/hl7/hl7_1_2.htm - 5.09.2012 г.

9. Проблемы интеграции данных [Электрон-

ный ресурс]. – Режим доступа:

<http://zeus.sai.msu.ru:7000/consulting/BI/integrdata/index.shtml> - 5.09.2012 г.

10. Опыт применения продуктов Informatica в зарубежных банках [Электронный ресурс]. – Режим доступа: <http://www.iso.ru/iournal/articles/541.html> - 25.04.2012 г.

11. IBM WebSphere - From Wikipedia, the free encyclopedia [Электронный ресурс]. – Режим доступа: <http://en.wikipedia.org/wiki/WebSphere> - 25.04.2012 г.

12. Разработка расширенных клиентских приложений Eclipse для платформы WebSphere [Электронный ресурс]. – Режим доступа:

<http://www.ibm.com/developerworks/ru/library/0608xw/> - 25.04.2012 г.

13. Websphere для новичков [Электронный ресурс]. – Режим доступа:

<http://www.ibm.com/developerworks/ru/vwebsphere/new>

to/ - 25.04.2012 г.

14. Microsoft BizTalk Server - From Wikipedia, the free encyclopedia [Электронный ресурс]. – Режим доступа: <http://en.wikipedia.org/wiki/BizTalk> - 25.04.2012 г.

15. Business process management - From Wikipedia, the free encyclopedia [Электронный ресурс]. – Режим доступа:

http://en.wikipedia.org/wiki/Business_process_management - 25.04.2012 г.

16. Поллис, Г. Разработка программных проектов на основе Rational Unified Process (RUP) [Текст] / Г. Поллис, Л. Огастин. – К.: Бином-Пресс, 2009. – 256 с.

17. Боггс, У. UML и RationalRose [Текст] / У. Боггс, М. Боггс. – М.: Лори, 2008. – 600 с.

18. Киммел, П. UML. Основы визуального анализа и проектирования [Текст] / П. Киммел. – СПб.: ИТ Пресс, 2008. – 272 с.

Поступила в редакцию: 22.09.2012

Рецензент: д-р техн. наук, проф., зав. каф. «Информационные управляющие системы» О.Е. Федорович, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина.

ІНТЕГРАЦІЙНА СИСТЕМА НА ОСНОВІ ЕЛЕКТРОННОГО ПРОТОКОЛУ ОБМІНА МЕДИЧНИМИ ДАНИМИ HEALTH LEVEL 7

О.Б. Лещенко, А.В. Гук, Ю.О. Лещенко

Пропонується підхід інтеграції медичних даних в єдине сховище. Використовується модифікований метод консолідації даних для забезпечення постійного обміну даними між системами. Розроблена підсистема здатна об'єднати медичні, страхові, інформаційні системи та забезпечити обмін повідомленнями між ними, надати можливість використання і управління медичною інформацією її власниками. В якості основного стандарту обміну медичними даними використовується Health level 7. Розглянуто сервіс-орієнтований підхід до інтеграції медичних, страхових та інших інформаційних систем, які здатні приймати та обробляти повідомлення зашифровані за допомогою стандарту HL 7.

Ключові слова: лікувально-профілактичний заклад, медична інформаційна система, інтеграція, Health level 7, web-сервіси.

INTEGRATION PLATFORM BASED ON THE PROTOCOL EXCHANGE OF HEALTH DATA HEALTH LEVEL 7

A.B. Leshchenko, A.V. Guk, Yu.A. Leshchenko

It is proposed approach integrating medical data in a single repository. Using a modified method of data consolidation to ensure regular exchange of information between systems. Is implemented subsystem that is capable to combine medical, insurance information systems, to provide the messaging between them, to give an opportunity for the use and management of medical information by its owners. As a basic standard of health data exchange has been selected HL7. Considered a service-oriented approach to the integration of medical, insurance, and other information systems are capable to receive and process messages encrypted with the standard HL 7.

Keywords: healthcare facility, medical information systems integration, Health level 7, web-services.

Лещенко Александр Борисович – канд. техн. наук, доцент, доцент кафедры информационных управляющих систем, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

Гук Андрей Васильевич – магистрант кафедры информационных управляющих систем, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.

Лещенко Юлия Александровна – инженер кафедры информационных управляющих систем, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.