**Особенности в проектировании и практической разработке медицинской информационной системы**

Гусев А.В., член-корр. РАМН Дуданов И.П., Романов Ф.А., Дмитриев А.Г., Карельский научно-медицинский центр СЗО РАМН, Петрозаводск

В последние годы в России появился целый ряд уникальных разработок в области комплексных медицинских информационных систем (МИС), предназначенных для автоматизации работы учреждений здравоохранения. Одними из самых интересных являются: ин-формационная система "Интерин" (Институт программных систем РАН), МИС "Артемида", МИС "Амулет" и некоторые другие. Эти системы не только разработаны, но и активно раз-виваются - распространения и признания в практическом внедрении они получили за по-следние 2-3 года. В литературе опубликованы положительные отзывы коллективов клиник самого разного профиля и масштабов об опыте применения информационных систем в рабо-те [1, 3, 5, 10, 13, 14, 17-20]. Наметилась тенденция на комплексное решение разносторонних задач лечебного учреждения, что особенно радует и свидетельствует о качественном росте отечественных разработчиков в области медицинской информатики.

Однако при более глубоком изучении этого процесса все сильнее выделяется сущест-венная проблема: несмотря на наличие глубоко проработанных программных решений, практически отсутствует опыт полного перехода на электронный принцип хранения и обра-ботки информации в лечебном учреждении [8, 11]. Имеется ряд серьезных преград, через ко-торые не могут перешагнуть даже современно оснащенные клиники в своем стремлении от-казаться от бумажных носителей информации и повысить эффективность в организации сво-ей работы. Все они могут быть объединены в несколько групп.

Во-первых, существующая в России правовая база не обеспечивает должного уровня юридической защиты медицинских работников, применяющих информационные технологии в повседневной практике. Во-вторых, финансовые ресурсы большинства учреждений здра-воохранения пока не могут позволить им приобрести достаточное количество компьютерной техники и дорогостоящего программного обеспечения для комплексной автоматизации. По-этому этот процесс успешно протекает лишь в некоторых, зачастую далеких от медицинской направленности, разделах работы ЛПУ: статистика, бухгалтерия, автоматизация работы ад-министративного звена и т. д. В третьих, в России практически отсутствует школа, которая бы готовила профессионалов высокого уровня в области разработки и внедрения именно комплексных медицинских информационных систем, людей по определению работающих на стыке сложнейших наук - медицины и прикладной математики. Для становления отечест-венной школы в этой области творческим коллективам необходимо обмениваться наблюде-ниями и мнениями в разработке программных продуктов, накапливая тем самым специаль-ные знания и формируя потенциально выгодные направления в поиске эффективных реше-ний разработки и внедрения комплексных МИС.

Коллектив авторов имеет 5-летний опыт в разработке медицинской информационной системы. За это время изучена на практике эффективность различных подходов. Применя-лись Microsoft FoxPro разных версий, CA Clipper, Lotus Notes, начиная с версии 4.6, СУБД Microsoft SQL Server, Microsoft Access, Borland Paradox, MySQL и IBM DB2. Апробирован вариант написания программного обеспечения на Borland Delphi, сервлеты на Java, применя-лись Lotus Designer и мультиплатформенный Lotus Script и некоторые другие технологии. Серверная часть системы работала под управлением Microsoft Windows NT Server, Microsoft Windows 2000 Server и Red Hat Linux 6.0. В качестве клиентской операционной системы применялись все версии Windows, начиная с Microsoft Windows 95. Кроме инструментария, была проведена работа по изучению эффективности различных методик проектирования ба-зы данных МИС. В итоге мы остановились на применении принципа объектно-реляционной парадигмы в проектировании БД МИС [4]. Кратко концепция этого подхода выражается в том, что в силу особенностей предметной области необходимо разрабатывать информацион-ную систему на базе синтеза двух, различных по своей природе, систем управления базами данных (СУБД) - объектно-ориентированной и реляционной. Только на основе этого синтеза удается исключить недостатки обоих СУБД и использовать их достоинства. В качестве ос-новной СУБД используется Lotus Domino Server R6.0.3 для функционирования объектной части МИС и MySQL 4.0.17 для реляционной составляющей системы. Разработка программ-ного обеспечения ведется в среде Lotus Designer R6.0.2 и Borland Delphi 6 Professional SP3. В ходе изучения эффективных способов создания приложений для системы найдено несколько, на наш взгляд, интересных находок.

Во-первых, одной из самых существенных причин увеличения стоимости МИС мы считаем высокую стоимость самой разработки. Изучив причины этого явления, мы пришли к выводу, что не последнюю роль в этом сыграла традиция создания медицинских информа-ционных систем на основе так называемых АРМ-ов (автоматизированных рабочих мест). Причем зачастую под АРМ-ом понимается именно клиентское программное обеспечение, хотя изначально этот термин имел более широкое толкование [10]. Чаще всего разработка АРМ-ов ведется по следующей методике: разработчики выбирают некоторую общую задачу (например, создание электронной истории болезни для стационара), проектируют структуру базы данных, разрабатывают приложение для работы с ней. Нередко это приложение выпол-няется в виде нескольких версий - АРМ главного врача, АРМ регистратора, АРМ лаборанта и т. д. Разработка систем в 65% случаев ведется на Borland Delphi. При этом даже на выпуск очень сырой версии АРМа тратится 4-8 месяцев. Затем столько же времени уходит на обкат-ку. Вместе с тем, по нашим оценкам, разработчику приходится 10-20% времени тратить на создание специфичного для медицинской области кода. Остальная часть, причем самая тру-доемкая и ответственная, приходится на разработку механизмов, обеспечивающих целост-ность данных, подсистему безопасности и администрирования МИС, связь с периферийным медицинским оборудованием и т. д.

Однако, не вызывает сомнений, что эти решения значительно уступают промышлен-ным решениям для корпоративного рынка, над которыми трудятся лучшие специалисты и которые прошли многолетнюю проверку. В связи с этим вызывают недоумение подобные попытки "изобрести велосипед". На наш взгляд, разработка МИС не должна осуществляться созданием и дальнейшей интеграцией отдельных АРМов. Для создания МИС необходимо применять готовые программные платформы для групповой работы, уже имеющие в своем арсенале мощные средства для мультиплатформенной разработки программы, готовые тех-нологии для развертывания и управления подсистемой безопасности. В своей работе мы вы-брали пакет групповой работы Lotus Notes/Domino, разрабатываемый в настоящее время корпорацией IBM. Эта платформа является за рубежом стандартом "де факто" для создания мощных информационных систем, ориентированных на обработку электронных документов и мы считаем, что она наиболее подходит для создания медицинских информационных сис-тем.

Работа над созданием МИС "Кондопога" на основе Lotus Notes/Domino версии 4.6 на-чата в сентябре 1999 года. Через 2 месяца МИС, включающая подсистемы работы врача, клинической и биохимической лаборатории, функциональной и рентгенологической диагно-стики, аптеки и планирования рабочего времени была поставлена в эксплуатацию. А еще че-рез 2 месяца лечебное учреждение, использующее систему, полностью перешло на элек-тронный способ хранения информации, отказавшись от бумажных носителей.

Вторым важным решением явился отказ от проектирования базы данных МИС по функциональному назначению, когда для отдельной задачи (например, подсистема лабора-тории, функциональной диагностики, консультационная и т. д.) создавалась своя база дан-ных. Хотя такой подход имеет ряд преимуществ, главным из которого является снижение требований к аппаратной мощности сервера за счет разделения потоков пользовательских запросов к отдельным БД. Однако было избрано проектирование БД МИС таким способом, что вся информация собиралась вокруг пациента и хранилась физически в одной БД.

Однако количество таких БД в МИС является вариабельным и зависит от количества функциональных групп пользователей, имеющихся в ЛПУ. Так, в стационаре это может быть выделенная БД для каждого отделения или корпуса. Для поликлиники - это могут быть БД, разделенные по участкам обслуживания. Кроме того, в этих БД специальным образом хра-нится только актуальная информация, а неиспользуемые данные помещаются в БД архива. Для решения ряда задач может быть принята либо связанная с объектно-ориентированным ядром реляционная БД, либо специальным образом сконструированные представления, ко-торые мы называем регистрами (рис. 1).

Рис. 1. Укрупненная схема объектно-реляционной базы данных медицинской информационной системы

Проектирование структуры БД, таким образом, позволяет достичь стабильно малого объема БД МИС в течение практически всего срока ее эксплуатации, а тем самым обеспе-чить максимально возможную производительность работы МИС. Так, начиная с 1999 года база данных историй болезни пациентов, проходящих реабилитацию в медицинском центре, имеет объем 26-29 Мбайт. При этом вся информация за время работы центра сохранена, а скорость работы остается стабильно высокой. Сложностью указанной методики является то, что программное обеспечение информационной системы должно поддерживать любое коли-чество физических баз данных в ядре системы, объединенных в одну логическую структуру.

Таким образом, необходимо разработать алгоритмы всех программ МИС так, чтобы они могли корректно работать с базой данных текущих документов, состоящей из одной или не-скольких частей. Это вызвано тем, что в некоторых случаях программам необходимо обра-ботать данные не только по отдельной части базы данных, но и по всей имеющейся в ней информации. В связи с этим необходимо перед каждым обращением к серверу выполнять ряд последовательных шагов:

определить, какое количество физических баз данных и их имен соответственно установ-лено на сервере;

определить возможность доступа к каждой базе данных в отдельности;

выполнить соответствующий запрос к каждой базе данных, указав в правильном формате полный адрес, включающий имя сервера и имя базы данных на нем;

обработать и запомнить полученный ответ;

повторить шаги 2-4 для каждой базы данных;

сложить накопленные ответы и обработать их, как единый пакет информации.

Доказано [5, 10, 16, 17, 19, 20, 22], что для эффективного решения такой задачи необхо-димо исключить в текстах программ МИС прямое обращение к базам данных. Взамен этого предложено использовать специальное промежуточное программное обеспечение, называе-мое сервисами middleware. Схема работы МИС на базе сервисов middleware показана на ри-сунке 2. С целью определения эффективности разработки системы с применением объектно-ориентированной технологии на основании использования сервисов middleware, авторами был выполнен анализ работы по созданию информационной системы в период 1999-2001 гг. Были получены следующие данные (таблица 1).

Таблица 1

Использование однотипного программного кода в различных подсистемах МИС

|  |  |  |
| --- | --- | --- |
| Подсистема | Общий код | Уникальный код |
| % от всего | % времени на разработку  | % от всего | % времени на разработку  |
| Документы ИБ и АК | 45 | 10 | 55 | 90 |
| Лабораторная подсистема | 74 | 38 | 26 | 62 |
| Статистика | 17 | 9 | 83 | 91 |

Как представлено в таблице 1, в некоторых видах ПО доля повторяющегося кода со-ставляет значительную часть. Исключение его из этапа разработки нового приложения по-зволило сократить среднее время создания новой программы с 3,8 до 2,9 недель (на 23,68%). Кроме этого, использование проверенных библиотек позволило значительно, на 35-50%, со-кратить количество последующих исправлений ошибок. Фактически вся основная работа над ошибками была связана с исправлением уникального кода приложения, в редком случае (в среднем 5-7 ошибок на 100 исправлений) исправлением используемых middleware-сервисов.

Таблица 2

Анализ ошибок и исправлений в МИС

|  |  |  |
| --- | --- | --- |
| Подсистема | До применения middleware | После применения middleware |
| Количество ошибок в неделю  | Время ис-правления ч/неделю  | Количество ошибок в неделю  | Время ис-правления ч/неделю  |
| Микроядро системы  | 26 | 4,5 | 2,9 | 3,5 |
| Статистика | 8 | 6,2 | 1,3 | 0,8 |
| Лабораторная подсистема | 1,2 | 3,4 | 0,2 | 1,2 |
| Внешние программы | 13 | 14,2 | 2,5 | 6,5 |
| Календари | 3 | 4,4 | 0,4 | 1,2 |

Дополнительно с широким применением базовых библиотек класса middleware, выпол-ненных в виде динамически подсоединяемых библиотек (dynamic link libraries DLL), было предложено встроить во все основные функции единый обработчик ошибок. В случае фа-тального прекращения работы какой-то функции middleware он пересылал системе резуль-тат, переданный по умолчанию, и дополнительно отправлял максимально возможную ин-формацию разработчику по e-mail. Это позволило сократить время, необходимое на анализ и исправление ошибки в среднем на 45-55%. Нередко исправление ошибки производилось уже до того, как пользователь сообщал об этом программистам [10, 14].

Необходимо отметить, что применение модели программного обеспечения системы на основе использования общих сервисов middleware позволяет применять эволюционный ме-тод, называемый в литературе Spiral Model [12, 18]. При этом возможно внедрение новых версий информационной системы путем простого подмена базовых сервисов на новые вер-сии. Эти версии могут работать как со старой информационной системой, так и с новой, без необходимости повторного обучения персонала или исправлений в структуре существующей базы данных.

Таким образом, применение сервисов middleware позволило в среднем увеличить появ-ление новых версий программ с 4 до 7 в месяц (на 75%), снизив удельную стоимость каждой новой версии на 22%. Применение указанных технологий позволило разработать систему со значительной экономией. Так, разработка крупнейшей отечественной МИС "Интерин" длит-ся 9 лет, штат разработчиков насчитывает 25 человек. Разработка ИС, в которой принимают участие авторы, осуществляется 4 года и только 2 последних из них в ней постоянно участ-вует 2 программиста. Приняв, что данная МИС содержит только 50% от возможностей МИС "Интерин", зарплата одного программиста составляет около $300 (долл. США), а работа ве-дется 11 месяцев в году, получена экономия по сравнению с традиционными технологиями около $75900 в год. Таким образом, за 4 года работы стоимость разработки МИС на основе объектно-реляционного подхода составила 5,3% от суммы, которая потребовалась бы для создания МИС с применением традиционного подхода.

Рис. 2. Схема работы промежуточного программного обеспечения и его место в структуре программ медицинской информационной системы

На этапе, когда ИС становится пакетом многочисленных программ, остро встает вопрос их поддержки. Актуальность ее растет вместе со сроком эксплуатации и ростом количества пользователей. Наряду с начальными капитальными затратами, администрирование инфор-мационной системы составляет значительную цифру в смете расходов ЛПУ [3, 5, 10, 14]. Применение сложных комплексных информационных систем требует высококвалифициро-ванного штата программистов и администраторов [12, 15]. С ростом количества подключен-ных к базе данных системы пользователей растет и сложность ее обслуживания. В таблице 3 приведены средние еженедельные затраты времени работы администратора МИС, получен-ные в результате хронометрических исследований в медицинском центре.

Таблица 3

Трудозатраты администратора МИС

|  |  |  |  |
| --- | --- | --- | --- |
| № | Вид деятельности | % общего времени | Примечание |
| 1 | Обслуживание вызовов пользователей на местах  | 38 |   |
| 2 | Установка пакетов исправлений | 17 |   |
| 3 | Отправка сообщений об ошибках разработчикам системы | 7.8 |   |
| 4 | Плановое обслуживание клиентских ПК (дефраг-ментация диска, проверка на вирусы) | 6,9 |   |
| 5 | Знакомство с литературой | 5 |   |
| 6 | Анализ журналов безопасности | 4 |   |
| 7 | Установка новых приложений | 3,5..45,7 | 45,7% при вне-дрении системы |
| 8 | Контроль за новыми версиями ПО и публикация-ми исправлений | 3,2 |   |
| 9 | Исправление сбоев в клиентских операционных системах | 0,1-2,1 | Максимум при Windows 95/98 |
| 10 | Внесение исправлений в системные справочники, текущие изменения настроек приложений | 0,3-8,6 | Максимум - при внедрении |
| 11 | Прочие | 14,2 |   |

Использование встроенных в middleware-приложений глобальных обработчиков оши-бок позволяет сократить время по п. 3 (отправка отчетов об ошибках) практически до нуля, т. к. вся ключевые отчеты система формирует и отправляет автоматически. Обслуживание вызовов пользователей, исправление локальных сбоев на компьютерах пользователей и вне-сение исправлений в справочники системы являются практически неуправляемыми факто-рами. Плановое обслуживание компьютеров, анализ журналов работы системы, чтение спе-циальной литературы являются постоянными величинами, истекающими из функциональ-ных обязанностей администратора системы.

Следовательно, управляемыми факторами, способными сократить (перераспределить) трудозатраты технического персонала на обслуживание системы являются: установка при-ложений системы, установка исправлений (в т. ч. самой ИС и общесистемного ПО), контроль за новыми версиями ПО. Причиной высоких показателей в этих категориях является тради-ционный способ установки прикладного ПО: администратор сети запускает инсталляцион-ные пакеты на компьютерах пользователей, а затем по мере появления и т. н. "заплатки" к ним. Учитывая высокие показатели в выходе новых версий отдельных программ МИС на ба-зе ООП, 1 администратор может обслуживать до 22-25 пользователей. По нашему опыту, время от появления пакета исправлений до его полной инсталляции на всех компьютерах се-ти может составлять от 2-3 суток при работе 50 станций до 4-7 суток при работе 100-150 станций. Этот факт чреват тем, что злоумышленник может воспользоваться этим промежут-ком для нарушения системы безопасности или другого нанесения вреда, если он знает меха-низм ошибки, которую планируется исправить "заплаткой".

Анализируя эти проблемы, было предложено использовать технологию установки и об-новления приложений [8, 11, 14]. Суть ее работы состоит в следующем: в системе имеется выделенная база данных дистрибутивов приложений. Все команды на запуск приложений используют в своей работе специальный сервис, предоставляемый системой. Ей передается команда на запуск приложения, содержащая код программы и параметры ее запуска. Всю необходимую работу выполняет система, используя следующий алгоритм (рис. 3).

Определяется, имеется ли описание программного продукта с переданным кодом в цен-тре программ. Если описание не найдено, выдается сообщение об ошибке.

Вычисляются из описания программы необходимые данные, в частности: номер версии ПО, имя исполняемого файла и т. д.

Проверяется, имеется ли вызываемое ПО на компьютере пользователя: если нет, произ-водится инсталляция программного обеспечения - из базы данных извлекаются необхо-димые файлы и настройки, создается программная папка и выполняется копирование файлов и т. д. После окончания процесса установки исполняемый файл запускается.

Если вызываемое ПО имеется на компьютере пользователя, проверяется его версия и сравнивается с версией в центре программ. В случае, если на локальном компьютере со-держится устаревшая версия, производится обновление программных файлов в зависимо-сти от настроек одним из следующих способов:

метод переустановки. Он является наиболее простым решением. При его использова-нии, однако, механизм синхронизации должен оценить следующие факторы: доступность обновленного дистрибутива, его объем и время копирования на данную рабочую станцию, права доступа данного пользователя к нужному дистрибутиву, возможные ограничения, накладываемые администратором сети на данный дистрибутив;

метод обновления. Он более трудоемкий. Его суть в том, что на данном компьютере приложение не переустанавливается целиком, а лишь перезаписывается его исправленная часть. Этот метод имеет преимущество в том, что объем данных для исправления значи-тельно меньше по сравнению с полным дистрибутивом.

метод исправления справочников. Применяется, если приложение само не изменило свою версию, однако его справочник устарел по сравнению с эталоном системы.

Если все проверки пройдены, исполняемый файл запускается.

Рис. 3. Алгоритм работы подсистемы установки и обновления программ

Применение данной технологии позволило сократить время, необходимое на обновле-ние клиентского программного обеспечения с 2-3 дней до 5-10 сек. (в среднем), снизить за-траты ЛПУ на администрирование информационной системы на 47,8% за счет снижения трудозатрат администратора системы и возможности совмещения ставок программиста и администратора. Ежегодная экономия составляет около $72 на одного пользователя.

**Список литературы**

Айламазян А. К., Гулиев Я. И. Данные, документы и архитектура медицинских инфор-мационных систем. http://interin.botik.ru

Айламазян А. К., Гулиев Я. И., Матвеев Г. Н., Турна И. А., Белова И. А. ИС КОТЕМ-2001: Требования, проблемы, решения. http://interin.botik.ru

Андерсон К., Минаси М. Локальные сети. Полное руководство: Пер. с англ. - К.: ВЕК+, ЭНТРОП, Спб.: КОРОНА принт, 1999. - 624 с.

Андреев А. М., Березкин Д. В., Кантонистов Ю. А. Выбор СУБД для построения инфор-мационных систем корпоративного уровня на основе объектной парадигмы // СУБД 1998. - № 4-5. - С.26-50.

Губин И. М., Тарасов В. В., Антонов Р. А. и другие. Разработка и внедрение новой авто-матизированной информационной системы ЦКБ // Кремлевская медицина. Клинический вестник, 2000. - №4. - С.51-54.

Гусев А. В., Романов Ф. А., Дуданов И. П.. Опыт разработки медицинской информаци-онной системы // Медицинский академический журнал, 2001.- №1.- Приложение 1.- С.18.

Гусев А. В., Романов Ф. А., Осиик Т. А.. Применение медицинской информационной системы в работе клинических лабораторий медицинского центра // Медицинский ака-демический журнал, 2001. - № 1. - Приложение 1. - С.19.

Гусев А. В., Дуданов И. П. Оценка 3-летнего опыта разработки и внедрения информаци-онной системы: выводы и перспективы // Медицинский академический журнал, 2002. - Том 2. - Приложение 2. - С.56-57.

Гусев А. В., Дуданов И. П., Романов Ф. А. Информационная система в медицине - кон-цептуальная модель.http://surgery.karelia.ru

Гусев А. В., Романов Ф. А., Дуданов И. П., Воронин А. В., Информационные системы в здравоохранении. ПетрГУ. Петрозаводск, 2002. - 120 с.

Дуданов И. П., Гусев А. В., Романов Ф. А., Воронин А. В. и соавт.. Информационная система в здравоохранении - концептуальная модель // Сердечно-сосудистые заболева-ния. Бюллетень НЦССХ им. А. Н. Бакулева РАМН. Том 3. - № 11. - 2002. - С.332.

Дуданов И. П., Гусев А. В., Романов Ф. А., Воронин А. В. Информационные системы в здравоохранении // Медицинский академический журнал, 2002. - № 1.- Том 2.- Стр.58-77.

Дуданов И. П., Гусев А. В., Романов Ф. А., Кемпи С. И. Региональная информационная система "Кондопога" // Сердечно-сосудистые заболевания. Бюллетень НЦССХ им. А. Н. Бакулева РАМН. 2002. - Том 3. - № 11. - С.335.

Дуданов И. П., Гусев А. В., Романов Ф. А., Кемпи С. И. и соавт. Создание "паспорта здо-ровья" больных с сердечно-сосудистыми заболеваниями с использования информацион-ной системы // Медицинский академический журнал, 2003. - Том 3. - № 3. - С.125-133.

Ильмаст А. В., Марусенко К. М., Моисеев Е. В. Некоторые вопросы технологии разра-ботки МИС // Медицинский академический журнал, 2002. - Том 2. - Приложение 2. - С.85-86.

Принципы проектирования и разработки программного обеспечения. Учебный курс MCSD: Пер. с англ. - М.: Издательско-торговый дом "Русская редакция", 2000. - 608 с.

Шеррер Жан-Рауль. Информационные системы в здравоохранении: технология и орга-низация // Кремлевская медицина. Клинический вестник, 2000. - № 4. - С.15-17.

Claudio G. A. da-Costa, MD, Rodrigo P. Quaresma, BE and Renato M. E. Sabbatini, PhD. A Software Engineering Approach to the Development of Computer-Based Patient Record Sys-tems. http://home.nib.unicamp.br/~claudiog/slides/seandcpr/seandcpr.htm

Ramamoorthy C. V., Prakash A., Tsai W. T., Usuda Y. Software Engineering: problems and perspectives. Computer. Outubro. 1984. - P.191-209.

Reidsema C., Szczerbicki E. A Blackboard database model of the design planning process in concurrent engineering. Cybernetics and Systems: An International Journal, 2001. - 32. - Р.755-774.

Sherrer J. R., Lovis C., Baund R., Borst F., Spahni S. Integrated Computerised Patient Records: The DIOGEN2 Distributed Architecture Paradigm with Special Emphasis on its Middleware Design. In User Acceptance of health Telematics Applications (I Iakovidis et al., Eds) IOS Press, Technology and Informatics 56. - P.15-31.

Spahni S., Sherrer Jr. Sauquet D., Sottile PA. Consensual trends of optimizing the constitution of middleware. ACM SIGCOMM Computer Communication. - 1998. - V.28, №5. - P.76-90.