

В качестве рекомендаций для реализации научно-инновационной деятельности предлагаются следующие мероприятия:

- разработка механизма повышения эффективного использования научно-инновационного и инновационно-образовательного потенциала, разработка и использование положения о стимулировании научно-инновационной деятельности;
- организация эффективного взаимодействия с предприятиями региона с целью интеграции вузовской науки с промышленностью и ориентацией на разработку и производство высокотехнологичной и конкурентоспособной продукции;
- конкурсная поддержка молодежных и студенческих инновационных проектов и формирование инновационной тематики исследований для выполнения в рамках курсового и дипломного проектирования;
- обеспечение активизации инновационной активности, инновационного мышления студентов, аспирантов, молодых ученых и преподавателей;
- развитие научно-инновационного центра и технопарка;
- развитие и ведение единых баз данных по вузовским разработкам.

Реализация названных мероприятий позволит ускорить развитие инновационной деятельности в вузе для реализации исследовательской, инновационной и предпринимательской моделей его функционирования.

Список литературы

1. Наука России в цифрах – 2002 : стат. сб. – М. : ЦИСН, 2002.
2. www.mon.gov.ru
3. Россия в цифрах – 2002 : крат. стат. сб. / Госкомстат России. – М., 2002.
4. *Jacob, M., Lundqvist, M.* University based Innovation Systems: The Chalmers Experience. – Goteborg, Sweden: Chalmers University of Technology, 2001.

УДК 519.688

П. В. Токмурзин, аспирант;

А. Г. Ицков, кандидат физико-математических наук, доцент
Ижевский государственный технический университет

ОСОБЕННОСТИ РЕАЛИЗАЦИИ МЕДИЦИНСКИХ ИНФОРМАЦИОННЫХ СИСТЕМ НА ПРИМЕРЕ ПОДСИСТЕМЫ «АНАЛИЗ ПОКАЗАТЕЛЕЙ СМЕРТНОСТИ»

В статье рассмотрены требования к подсистеме анализа показателей смертности комплексной медицинской информационной системы. Основное внимание при проектировании подсистемы было уделено повышенной гибкости и настраиваемости. Приведены схемы таблиц подсистемы с обоснованием выбора СУБД, структура классов и их описание.

Требования к структуре подсистемы

Подсистема «Анализ показателей смертности» должна решать следующие задачи:

- 1) обеспечение возможности регистрации случаев смерти как в оперативном режиме, так и в режиме загрузки архивных данных (например, из архивной базы данных или старого программного обеспечения), в том числе и в режиме массовой вставки;

2) обеспечение возможности настраивать набор компонентов, входящих в подсистему МИС:

- расширение набора справочников, используемых подсистемой;
- создание новых аналитических отчетов;

3) предоставление средств для построения отчетности;

4) предоставление набора инструментов для проведения анализа данных, в том числе для data-mining-анализа.

На рис. 1 приведена схема контуров подсистемы «Анализ показателей смертности».

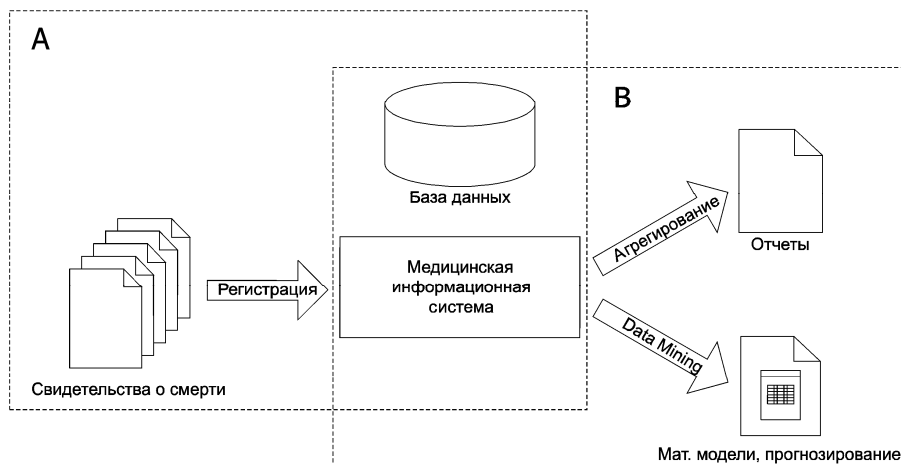


Рис. 1. Схема контуров подсистемы «Анализ показателей смертности»

На схеме обозначены контуры:

А – контур регистрации данных;

В – контур анализа данных.

Разнесение задач по двум контурам не нарушает комплексности подсистемы, поскольку такая архитектура позволит функционировать подсистеме в едином информационном пространстве.

Непосредственно из поставленных задач следует разделение функций подсистемы по автоматизированным рабочим местам (АРМ):

- АРМ оператора – выполнение функций ввода данных (задача 1);
- АРМ разработчика – выполнение функций по настройке условно-постоянной (справочной) информации (задача 2);
- АРМ аналитика – проведение анализа накопленных данных (задачи 3, 4);
- АРМ администратора – управление пользователями и историей работы с объектами подсистемы.

Подобная организация АРМ позволит разделить ответственность между различными сотрудниками (а значит, и повысить ответственность при работе с подсистемой).

Каждому рабочему месту соответствует один из типов пользователей: «Оператор» (доступ только к журналу регистрации и всем справочникам подсистемы); «Разработчик» (доступ к метаданным подсистемы, настройке отчетов и построению моделей); «Аналитик» (построение отчетов и проведение анализа случаев смерти на основании построенных моделей); «Администратор» (полный доступ ко всем данным подсистемы).

Выбор СУБД

Если выбор среды разработки не столь критичен, поскольку на сегодняшний день большинство сред визуального проектирования приложений предоставляют сопоставимые возможности разработки (проектно-модульная структура приложения, состав компонентов, поддержка сетевых протоколов), то выбор системы управления базами данных, на которой будет функционировать подсистема, необходимо осуществлять с учетом ряда факторов.

Подсистема «Анализ показателей смертности» изначально задумывалась как многопользовательская: заносить свидетельства о смерти в подсистему могут несколько операторов, функции оператора и аналитика могут выполнять сотрудники разных отделов медицинских учреждений и т. п. Поэтому вариант использования настольной СУБД не рассматривался, т. к. многопользовательскую работу с такой СУБД можно реализовать только в режиме «файл-сервер». При этом для рабочей станции имитируется работа с сетевыми файлами, как будто они находятся непосредственно на рабочей станции (это делается средствами сетевой операционной системы). С точки зрения создания медицинской информационной системы эта модель имеет ряд недостатков: высокий сетевой трафик (передача множества файлов, необходимых приложению); узкий спектр операций манипулирования данными (поскольку данные ассоциированы с файлами); отсутствие адекватных средств безопасности доступа к данным (защита только на уровне файловой системы). Альтернативной моделью является технология «клиент-сервер».

Технология «клиент-сервер» – это технология работы в сетевой среде, при которой осуществляется распределение программы по двум логически различным компонентам, каждый из которых выполняет свои отдельные функции. Обычно клиент посылает запросы на сервер на выполнение от своего имени определенной работы. Задачей сервера является обработка запросов и возврат результатов клиенту [1].

В качестве СУБД для подсистемы «Анализ показателей смертности» был выбран продукт компании Microsoft – Microsoft SQL Server 2000. Можно выделить следующие причины принятия такого решения.

1. Наличие нескольких редакций продукта: от полной редакции SQL Server 2000 Enterprise Edition, обеспечивающей повышенную масштабируемость и надежность, до SQL Server 2000 Personal Edition для индивидуального использования, в том числе и на не-серверных операционных системах [2].

2. Поддержка клиент-серверной технологии. Функции подсистемы можно разделить на три группы:

- функции ввода и отображения данных;
- прикладные функции, характерные для данной предметной области (регистрация свидетельства о смерти, модификация справочной информации);
- фундаментальные функции хранения и управления информационными ресурсами (базами данных, файловыми системами).

Функции распределяются между логическими компонентами приложения и затем между компьютерами в сети. Все функции третьей группы передаются ядру СУБД, которая выносится на сервер. Обращения к серверу теперь делаются в виде запросов к СУБД, языком запросов является SQL. Роль СУБД возрастает, т. к. СУБД обеспечивает не только обращение, но и обработку данных – выборку данных в соответствии с запросом. Часть прикладных функций также выносится на сервер в виде хранимых процедур и триггеров. Возможности по их созданию также

обеспечиваются SQL-серверами. Приложению на клиентской станции остаются функции первой группы и частично второй [3].

3. Наличие у разработчика статуса Microsoft Certified Professional (MCP) по результатам экзаменов, входящих в трек Microsoft Certified Database Administrator (MCDBA).

Схема базы данных

При разработке подсистемы МИС «Анализ показателей смертности» особое внимание было уделено реализации инструментов разработки. Настраиваемость подсистемы является одним из основных требований к информационным системам и медицинским информационным системам в частности. Применительно к подсистеме МИС это качество программного продукта позволит изменять состав справочников подсистемы (менять состав полей существующих справочников, а также добавлять новые), создавать новые аналитические отчеты, модифицировать модели оценки риска (уточнять существующие и строить новые модели) без повторной перекомпиляции приложения. Такой подход позволит сопровождать подсистему силами медицинского учреждения, без привлечения разработчика.

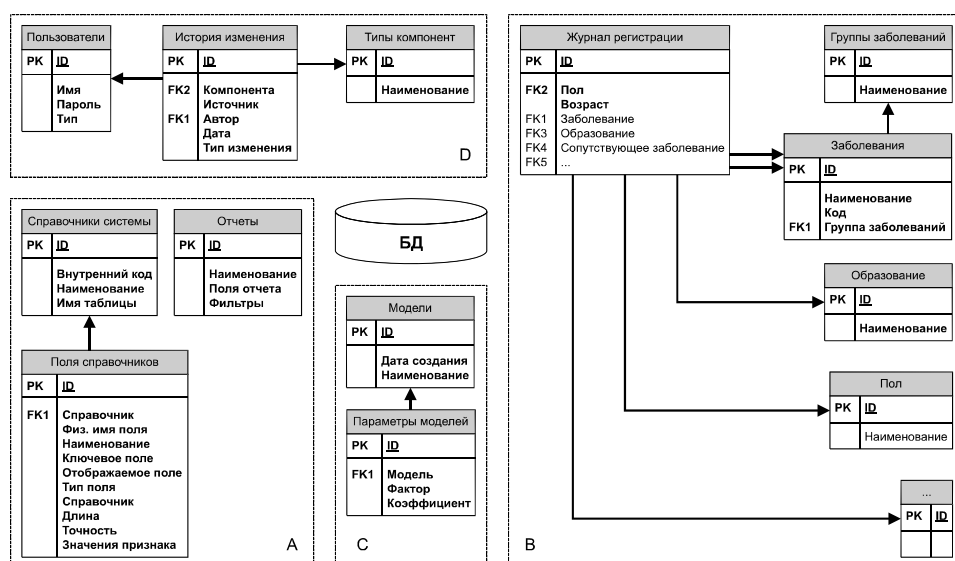


Рис. 2. Структура таблиц подсистемы «Анализ показателей смертности»

В подсистеме МИС «Анализ показателей смертности» (рис. 2) таблицы можно разделить на 4 группы.

1. А – таблицы разработчика. К ним относятся метаданные справочников (состав и структура справочной информации подсистемы) и отчетов. Структура системы позволяет использовать иерархические справочники.

2. В – таблицы оператора. Центральной таблицей, которая и служит основанием для проведения анализа, является журнал регистрации случаев смерти. Для заполнения справочных разрезов (аналитик) журнала используется система справочников, настраиваемая разработчиком подсистемы. На рисунке приведена только часть таблиц, т. к. состав справочников может настраиваться.

3. С – таблицы аналитика, в которых содержатся данные о построенных моделях с учетом хронологии.

4. D – таблицы администратора. Функции таблиц – хранение информации о пользователях подсистемы и истории изменения объектов разработки.

Состав подсистемы и структура классов

В подсистеме «Анализ показателей смертности» применены методы представления информации, используемые в большинстве современных учетных информационных систем.

Для каждого из 4 типов АРМ подсистемы начальной формой для работы является проводник компонент. Проводник представляет собой древовидную структуру папок (например, «АРМ оператора» – «Справочники» – «Классификатор районов»). Каждый листовый элемент такой структуры представляет собой ссылку на одну из компонент подсистемы, каждая из которых выполняется в отдельном исполняемом файле. Структура и состав компонентов настраивается в xml-файле специального формата, что позволит менять внешний вид проводника в зависимости от нужд пользователей (так, например, оператору нет необходимости видеть в «своем» проводнике компоненты администратора).

Поскольку структура классов подсистемы весьма обширна, приведем описание отдельных ее компонентов.

Справочники

Структура классов для справочников приведена на рис. 3.

При открытии любой из компонент подсистемы (в том числе и справочника) определяются параметры подключения к базе данных, сама возможность подключения (ограничения по правам типов пользователей). Свойства подключения описывает класс TComponentParams.

В момент открытия справочника в любом из режимов (карточка, список, выбор записи) на основании параметров командной строки (при запуске из проводника) или на основании главной формы (при открытии из другого справочника) определяются параметры справочника. Эти параметры инкапсулирует класс TReferenceInfo.

Состав полей справочника определяется динамически, на основании метаданных (раздел таблиц А на схеме БД). Все свойства полей собраны в классе TReferenceFieldInfo. В подсистеме поддерживаются поля нескольких типов: строка (с указанием длины), целое число, дробное число (с указанием точности), дата, признак (указываются возможные значения), справочник (ссылка на запись справочника, что позволяет строить также и иерархические справочники). Состав полей для справочника определяется в момент его открытия и представляется экземпляром класса TReferenceFields.

Для визуальной работы со справочниками используются два потомка класса TForm:

- TReferenceList – класс, описывающий справочник в форме списка. Класс инкапсулирует все методы работы с записями справочников: добавление, удаление, копирование, редактирование. При разработке универсальной формы использовались компоненты пакета ExpressQuantumGrid, поэтому функции фильтрации, группировки и сортировки данных также будут присущи любому справочнику подсистемы. Состав полей определяется на основании метаданных; поля хранятся в экземпляре класса TReferenceFields. Эти же поля используются при формировании формы-карточки справочника.

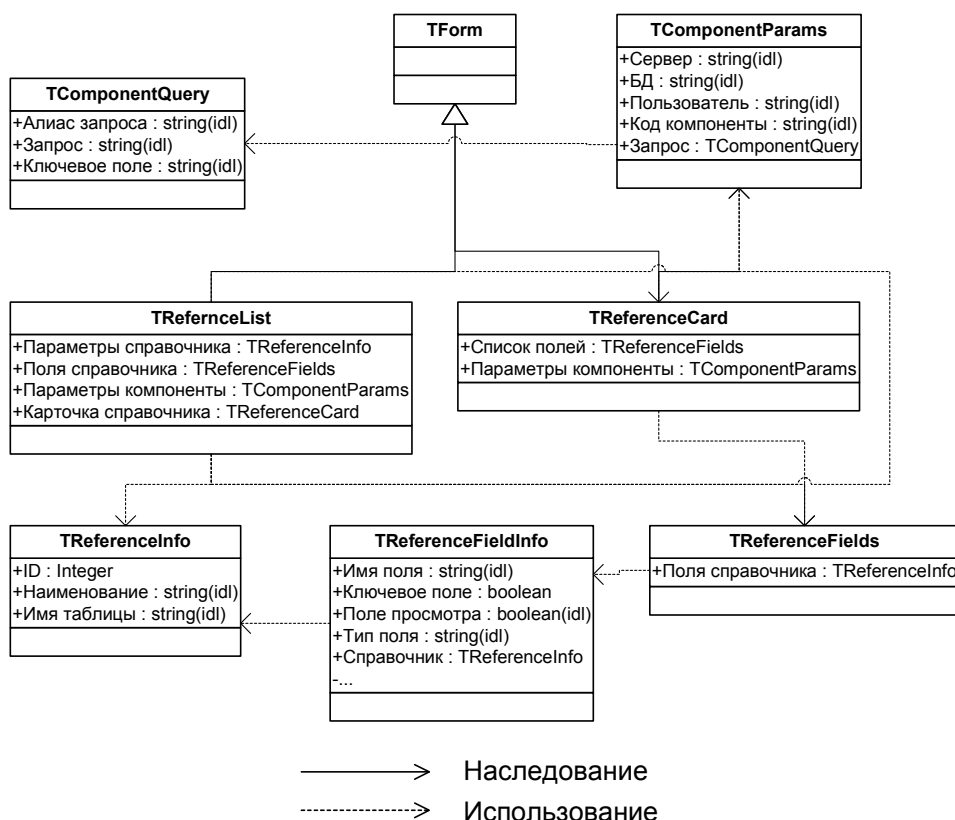


Рис. 3. Структура классов справочной информации подсистемы «Анализ показателей смертности»

• TReferenceCard – класс, инкапсулирующий работу с записью справочника в форме карточки. Элементы управления на форме располагаются динамически в соответствии с метаданными справочника, а также автоматически формируются ограничения на их возможные значения (например, запрет вводить в реквизит типа «Дата» произвольные целые числа). Важную роль при этом играют реквизиты типа «Справочник». При работе с карточкой справочника поддерживаются следующие способы выбора записи: выбор из полного списка справочника, использование поиска по части наименования и указание точного значения внутреннего кода. Для формирования запросов к другим справочникам при этом используется объект класса TComponentQuery.

Отчеты

Аналитические отчеты предназначены для анализа показателей смертности в разрезе справочников подсистемы. Отчеты представляют собой сводные таблицы и диаграммы Microsoft Excel.

Для работы с отчетами используется класс TAnReport. Класс содержит свойства, определяющие параметры отчета: фильтры, отображаемые поля по столбцам

и строкам (разрезы, в которых будет производиться анализ данных) и типы выводимых показателей (относительные или абсолютные), а также тип агрегирования итоговых значений.

Использование такого механизма позволит получить все документы, используемые для отчетности, например: «Уровень и структура смертности по классам болезни», «Смертность населения по причине смерти», «Распределение числа умерших по дням года» и т. п.

Модели

Подсистема среди прочих инструментов для анализа данных предоставляет такую функцию, как классификация социальных групп населения по группам риска. Классификация производится при помощи бинарных моделей.

Базовым классом для построения моделей является класс `TBinaryModel`, инкапсулирующий методы построения моделей на основании указанной обучающей выборки, а также методы проверки моделей на контрольных выборках. В качестве выборки может быть указан любой экземпляр класса `TDataSet`, что позволяет применять модель к любым наборам данных с использованием фильтрации и определять различные наборы факторов для анализа.

Примером реализации определенного класса бинарных моделей является класс `TLogitBinaryModel` (наследник класса `TBinaryModel`), также реализованный в подсистеме. О примерах применения такого класса моделей к показателям смертности можно прочитать в [4].

Заключение

В данной статье были рассмотрены требования к подсистеме анализа показателей смертности комплексной медицинской информационной системы. Основной упор при разработке подсистемы был сделан на повышенную гибкость и настраиваемость (в том числе без перекомпиляции всего проекта), что объясняет большое количество используемых таблиц и разветвленную структуру классов. Приведены схемы таблиц подсистемы с обоснованием выбора СУБД, структура классов и их описание для реализованной подсистемы.

Список литературы

1. Шэнк, Джеффри Д. Технология «клиент–сервер» и ее приложения. – М. : Лори, 1995. – 418 с.
2. Проектирование и реализация баз данных Microsoft SQL Server 2000 : учеб. курс MCSE. – М. : Русская редакция, 2001. – 704 с.
3. Microsoft SQL Server 7.0. – СПб. : БХВ Санкт-Петербург, 1999. – 720 с.
4. Токмурзин, П.В. Логистическая модель анализа риска для показателя суицида на территории Удмуртской Республики // Вестник Удмуртского университета. – 2006. – № 1.