

УДК 181.3

Н.В. Куликов (6 курс, каф. ИУС), Б.В. Абрамов (гл. сис. аналитик, ДЕКОСП),  
Д.Ф. Дробинцев, ст. преп.

## ПРОЕКТИРОВАНИЕ СИСТЕМЫ, РЕАЛИЗУЮЩЕЙ ПРЕОБРАЗОВАНИЕ ДАННЫХ ЧЕРЕЗ ОБЪЕКТНОЕ ПРЕДСТАВЛЕНИЕ

В настоящее время на рынке функционирует множество информационных систем (ИС) различной специализации. В процессе эксплуатации и эволюции ИС рано или поздно возникает необходимость переноса данных. Данная задача является частным случаем более общей проблемы совместимости ИС.

Выделим основные предпосылки возникновения данной проблемы:

- Перенос данных из старой версии ИС в новую.
- Перенос данных из одной ИС в другую ИС.
- Перенос данных между подсистемами одной ИС.

При этом перенос может быть как единовременным (миграция), так и непрерывным (синхронизация) в процессе эксплуатации ИС.

В большинстве случаев логическая и физическая структура хранения данных в источнике и приемнике различна и реализована в реляционной СУБД. Поэтому задача переноса должна решаться вместе с задачей преобразования данных. Таким образом, в процессе конвертации должны реализовываться следующие функции:

- преобразование логической структуры хранения данных;
- преобразование значений данных;
- селекция данных;
- перенос данных из одной СУБД в другую.

Причем эти функции должны выполняться как по отдельности, так и в комплексе.

Рассмотрим традиционный и предложенный подходы к решению данной проблемы. При традиционном подходе осуществляется последовательная конвертация данных из одного множества связанных таблиц в другое. Процесс конвертации описывается некоторым алгоритмом, который жестко привязан к конкретным ИС или подсистемам. Данный подход не является универсальным, так как требуется заново переписать алгоритм при конвертации новых структур данных.

В целях построения более гибкого и настраиваемого средства был предложен объектно-ориентированный подход к описанию и реализации процесса конвертации. Данный подход позволяет выделить и зафиксировать общую объектную структуру данных для ИС одного класса. Имеются в виду ИС, решающие одинаковый класс задач и работающие с объектами из одной предметной области. Для описания классов объектов и связей между ними было решено использовать иерархическую структуру. В перспективе это позволит, увеличивая уровень абстракции, описать в объектном виде предметные области информационных систем практически всех классов.

В свете объектно-ориентированного подхода данные – это множество статичных объектов, которые можно объединить в один или несколько классов по сходному набору признаков (свойств). Соответствие табличного и объектного представления данных выглядит следующим образом:

- Поле или функция полей – Свойство.
- Запись и связанные с ней записи – Объект.
- Master-таблица и связанные с ней таблицы-справочники – Класс.
- Detail-таблица, связанная с master-таблицей – Подкласс (наследует свойства класса или подкласса-родителя).

- Множество несвязанных таблиц – Группа классов.

Функция может быть: математической для числовых данных, строковой для символьных данных и т.д. Любая запись в master-таблице может быть связана только с одной записью из таблицы-справочника и с несколькими – из detail-таблицы.

В дальнейшем информация об объектах может быть либо сохранена в специальном файле, либо записана в соответствующие таблицы базы данных приемника.

Преимущества описанного решения в следующем:

- перенос объекта целиком (включая вложенные объекты) без потери какой-либо информации, находящейся в разных таблицах;
- построение иерархии классов, которая не будет меняться для однотипных ИС;
- простота настройки конвертора путем построения дерева классов и подклассов с использованием визуального конструктора;
- возможность представления результата конвертации в объектном виде и удобное их восприятие.

Кроме этого, данный подход предоставляет возможность промежуточного сохранения результата конвертации в отдельном файле, что позволит разнести во времени и в пространстве конвертацию данных из таблиц источника в объектную форму и запись результата в таблицы приемника.

Из недостатков можно выделить:

- увеличение длительности процесса конвертации;
- невозможность построения объектной модели для сильно денормализованной реляционной структуры.

Рассмотрим два варианта логической реализации объектного представления данных, каждый из которых имеет свои достоинства и недостатки. В первом случае объекты являются автономными (агрегированными) и хранят все данные внутри себя. Это приводит к следующему:

- отсутствие связей между объектами;
- простота реализации;
- быстрый доступ к данным объекта;
- денормализованность структуры (хранение избыточной информации).

Во втором случае объекты содержат в себе не другие объекты, а только указатели на них. Это приводит к следующему:

- появление связей между объектами;
- гибкость полученной структуры (изменение какого-либо объекта не требует изменения всех объектов, указывающих на него);
- компактность (занимает малый объем памяти);
- сложность реализации;
- увеличение вероятности ошибки при большом количестве связей;
- замедление доступа к данным объекта.

В рамках данной работы была программно реализована система конвертации данных (для ОС Windows), в основе которой лежит первый вариант объектного представления данных. При разработке был применен объектно-ориентированный анализ и проектирование. В системе реализована возможность визуального конструирования дерева классов и просмотра результата конвертации в виде дерева объектов.

Эксплуатация опытной версии системы подтвердила гибкость примененного подхода.

Дальнейшее развитие системы может включать поддержку связей между объектами, что обеспечит большую эффективность.