

# Разработка информационных моделей для ГИС (Графических Информационных Систем) железнодорожного транспорта

Аспирант, Смирнов Сергей Николаевич

Сложность информационных моделей в современном информационном обществе предъявляет новые требования к информационным системам.

В недавнем прошлом главным требованием вычислительной системы была ее способность в достаточной степени решать определенную задачу, сейчас возникли новые требования, связанные с необходимостью интеграции технологий, данных и технических средств. Интеграция означает, что помимо большого множества составляющих элементов других технологий имеется некая концепция и методология, оптимально объединяющая это разнообразие элементов. Поэтому не следует путать применение суммы различных методов с их интеграцией.

Эта тенденция объясняет применение в сложных программных комплексах концепции "открытых систем" OLE, а также чрезвычайную популярность XML, как универсального механизма обмена данными и хранения данных и документов.

Мой интерес к методу R-функций вызван тем, что этот метод обработки графической информации лучше всего подходит для адаптации под новые требования интеграции приложений и совместной обработки разнородных данных.

Механизмы, предусмотренные этим методом, дают возможность не только выполнять описание сложных геометрических поверхностей, но и оптимально хранить и передавать графическую информацию в разнородной сетевой среде.

Геометрический облик объекта формируется представлением фрагментов как оболочечных R-операторов, представляющих собой процедуру вида [4]

$$R = \text{SPR}(IR, N, M', a_{m1}, a_{m2}, \dots, a_{mn})$$

Здесь имеется возможность ввести дополнительные элементы, расширяющие возможности обработки ГО. Это: во первых, дополнительные атрибуты, которые описывают ГО, а во вторых, дополнительные программные модули, которые анализируют атрибуты, описывающие объект и выбирают наиболее оптимальный алгоритм обработки изображения.

В качестве дополнительного элемента R-оператора предлагается ввести характеристику объекта IM, которая представляет собой расширение описания ГО. Прежде всего, рассмотрим, какую информацию она должна содержать, а потом её возможную логическую и программную реализацию. Дальше - как использовать этот механизм при обработке ГО.

Из многообразных требований, которым должна отвечать характеристика объекта IM, выделим следующие:

- информативность - подразумевает, что атрибут должен хранить достаточное количество информации для адекватного описания объекта;

- размер - параметр, конфликтующий с требованием информативности. Означает, что данные должны храниться в наиболее компактном виде для рационального использования памяти;
- простота использования - формат атрибута должен быть простым, понятным и удобным для использования разработчиком. Это позволит уменьшить количество ошибок на этапе создания программ, повысить надежность их функционирования;
- расширяемость - должен иметь возможность добавлять новые элементы описания, при этом не затрагивать существующие, и не требовать изменения уже созданных программных модулей.

Состав и структура характеристики определяется многими показателями, зависящими как от самого ГО, так и от процессов, осуществляющих его обработку.

Выделим такие атрибуты как:

- сложность кусочных поверхностей;
- количество кусочных поверхностей;
- сложность связей между кусочными поверхностями;
- режим представления;
- параметры сетевого транспорта;
- состояние объекта;
- дополнительные данные пользователя.

Существует несколько вариантов логической реализации, например, в виде числового параметра, массива битов, записи. Последний вариант предпочтительней, т.к. он, несмотря на то, что имеет не самые оптимальные размеры, наиболее удобен и универсален.

Реализация в терминах Object Pascal будет иметь вид:

```
type TIM =  
  record  
    iAmount : integer;  
    iRel      : integer;  
    iMode     : integer;  
    ...  
  end;
```

var IM: TIM;

Процедуру, осуществляющую построение изображения, включающего расширенное описание, предлагается представить как

$$R = \text{MSPR}(IM, IR, N, M', a_{m1}, a_{m2}, \dots, a_{mn})$$

Что касается обработки ГО, решено перенести предлагаемые ранее алгоритмы программ на новые принципы объектно-ориентированного программирования, т.к. используемые ранее подходы не соответствуют возросшей сложности задачи.

Новые технологии программирования позволяют не только ускорить разработку приложений, но и придать им новые свойства. Нельзя оставлять без внимания новые возможности программирования, т.к. эти технологии предоставляют новые, значительно более эффективные механизмы для "реализации в жизнь" результатов теоретических исследований. Они позволяют эффективнее использовать возможности теоретического аппарата.

В течение нескольких последних лет создавались разные методологии разработки программного обеспечения.

Современная наука программирования рекомендует при разработке сложных приложений применять объектно-ориентированную методологию, согласно которой наилучшим способом разделения сложной системы на подсистемы считается создание таких моделей, которые фокусируют внимание на объектах, найденных в самой предметной области. Эти модели образуют объектно-ориентированную декомпозицию системы.

После такой декомпозиции по результатам ее объектно-ориентированного анализа выполняется объектно-ориентированное проектирование, которое, в свою очередь, создает фундамент для окончательной реализации системы с использованием методологии объектно-ориентированного программирования.

Концепция объектно-ориентированного программирования возникла не вдруг. Идея использования программных объектов развивалась разными исследователями в течение многих лет.

Объектно-ориентированное программирование - это новый способ подхода к программированию. Такое программирование, взяв лучшие черты структурного программирования, дополняет его новыми идеями, которые переводят в новое качество подход к созданию программ. Наиболее важное понятие языков объектно-ориентированного программирования - это понятие объекта.

Объект формирует связь между кодом и данными. Только те функции, которые объявлены в классе, имеют доступ к приватной части класса.

Объектно-ориентированное программирование имеет дело с объектами, включает в себя создание объектов, которые объединяют данные и правила обработки этих данных.

Объекты могут включать в себя частные, закрытые, приватные (private) данные и правила их обработки, доступные только объекту и его наследникам, а также общие (public) данные и правила, которые доступны объектам и модулям в других частях программы. Важной чертой объектно-ориентированного программирования является наследование, т.е. возможность создавать иерархическую последовательность объектов от более общих к более специфическим, частным. В этой иерархии каждый объект наследует характерные черты объектов-прародителей, объектов, предшествующих ему.

Объектно-ориентированный язык - язык программирования, на котором программа задается описанием поведения совокупности взаимосвязанных объектов. Объекты обмениваются запросами; реагируя на полученный запрос, объект посылает запрос другим объектам, получает ответы, изменяет значения своих внутренних переменных и выдает ответ на полученный запрос. Механизм запросов в объектно-ориентированных языках отличается от механизма процедур в процедурных языках тем, что при выполнении запроса объектом непосредственно могут быть изменены только значения переменных этого объекта".

Таким образом, языки объектно-ориентированного программирования содержат в себе следующие основные черты: наличие объектов и инкапсуляцию данных, наследование, полиморфизм, абстракцию данных.

Предлагаемый программный компонент построен на основе комплекса программ SLOGO [3]. Существующие элементы программ переработаны и объединены в класс TRFn, наследник от класса TObject, при этом модифицированные процедуры стали методами класса, а переменные, массивы, и т.д. - его свойствами или членами класса. Сюда же включены и новые функции.

Реализация в терминах Object Pascal будет иметь вид:

```
type
TRFn = class(TObject)
private
    XH: integer;
    XK: integer;
    YH: integer;
    YK: integer;

protected
    function MSPR(IM: TIM; JR: byte, IK: integer, IS:
integer, JB: byte): TRF;
    function RT(A: real; B: real ; IR: byte, L: integer; K:
integer): TRF;

public
    procedure LokRF(N: integer; X: integer: Y: integer: Z:
integer);

end;

var
RFn: TRFn;
```

## ЛИТЕРАТУРА

1. Цикритзис Д., Лоховски Ф. Модели данных. – М.: Финансы и статистика, 1986.
2. Цветков В.Я. Информатизация: создание современных информационных технологий. – М.: ГКНТ, ВНТРИЦентр, 1990.
3. Коптева Л.Г., Васильев В.А., Уколов И.С. Математические модели поверхностей составных объектов для цифрового анализа изображений в системах синтеза визуализации., НТО, договор № 38С, этап 1, ЛПУ ЛА Научного Совета по кибернетике АН СССР., М., 1985, ДСП
4. Коптева Л.Г. Представление составных трехмерных объектов для использования в САПР // Изв. Вузов. Сер. Авиационная техника. -Казань, 1985, -№3.
5. А.К. Агапеев, Дегтярев О.В. Создание программных средств для разработки комплексов математического и полунатурного моделирования авиационных систем на многопроцессорных структурах и сетях ЭВМ // Отчет, ГосНИИАС 131 (13653)-93, 1993г.

Аспирант, Смирнов Сергей Николаевич,  
Россия, Москва, РГОТУПС, e-mail: qs@FreeMail.ru