

УДК 004.438

Д.А. Заставной

ВСТРОЕННЫЙ ЯЗЫК СКРИПТОВ ДЛЯ GIS-СИСТЕМЫ WINMAP

Представлен язык, предназначенный для разработки скриптов для экспериментальной GIS-системы WinMap. Язык основан на объектно-компонентной модели. К основным функциям языка относятся доступ и манипулирование со структурами данных ядра, представляющих графические объекты, атрибутивную информацию, и настройка интерфейса оболочки системы.

Встроенный язык; язык скриптов; GIS-системы.

D.A. Zastavnoj

EMBEDDED LANGUAGE OF SCRIPTS FOR GIS-SYSTEM WINMAP

A specialized programming script language embedded in an experimental GIS-system WinMap is presented in the paper. The language based on a component object model features the access to the GIS kernel to manipulate graphical objects and attribute data as well as the variety of the system shell functions.

Embedded language; scripts; GIS-systems.

Встроенные языки для GIS-систем. Основными функциями геоинформационных систем является представление в базе данных графических пространственных объектов и ассоциированной с ней атрибутивной информации, воспроизведение этих объектов и выполнение различных поисковых операций и видов анализа, сочетающих традиционные для баз данных механизмы селекции с характерными для пространственных данных поисковыми операциями [1].

Современные GIS-системы ориентированы в первую очередь на работу в интерактивном режиме. Такие типичные операции, как ввод и редактирование пространственных данных и атрибутивной информации, выполнение поиска и анализа, выполняются пользователем в интерактивном (ручном режиме) при помощи соответствующих средств графического интерфейса, обращающихся к ядру системы. Использование только встроенных средств, однако, означает некоторую ограниченность в функциональных возможностях системы, в частности, в области поиска данных по произвольным критериям и произвольной обработки данных.

Для расширения возможностей GIS и частичного устранения ограничений указанного вида традиционно применяется два взаимно дополняемых подхода. Первый из них состоит в расширении функций ядра при помощи разработки плагинов – программных модулей, написанных на современных языках программирования, который взаимодействуют с ядром системы при помощи специальных программных интерфейсов, предусмотренных разработчиками системы. Эти плагины далее встраиваются в оболочку систему и доступны для использования при помощи GUI, наряду с базовыми средствами системы.

Второй подход состоит в использовании специализированных скриптовых языков, получивших название встроенных (англ. – embedded). Эти языки позволяют выполнять при помощи соответствующих команд различные действия с Ядром-системой, в основном дублирующие возможности, доступные через GUI, что существенно автоматизирует выполнение многочисленных специфических задач. Для ввода и исполнения команд предусмотрен встроенный интерпретатор. Этот подход является более доступным для конечного пользователя по сравнению с созданием плагинов.

Вопрос о разработке языков для пространственных баз данных и ГИС-систем не является новым (например, язык Avenue ArcView [2], MapBasic MapInfo [4], Oracle Spatial [3]); также следует упомянуть встроенные языки для пакетов 3D-графики (Maya MEL и JavaScript для SoftImage). В данной работе используется аналогичный подход для построения скриптового языка встроенного типа, предназначенного для использования в экспериментальной ГИС-системе WinMap. Основное отличие предложенного языка состоит в оформлении его синтаксических и функциональных средств на основе объектной модели.

Функциональные возможности встроенного языка GIS-системы. Встроенные языки GIS-систем традиционно реализуют следующие основные функциональные средства.

1. Традиционные возможности языков программирования, к которым относятся работа с переменными, управляющие конструкции и обработка ошибочных ситуаций.
2. Наличие «стандартных» библиотек для выполнения каких-либо традиционных действий, например, для работы с файлами ОС, консольного ввода/вывода, и простейших средств оконного диалога.
3. Средства для доступа к конфигурации графической оболочки системы, в частности, для добавления новых функций к существующему меню системы; эта возможность необходима для настройки системы для применения в конкретных областях.
4. Доступ к графическим объектам и ассоциированной с ними информацией – к координатным данным, стилям прорисовки и т.д.
5. Средства для синхронизации действий, выполняемых средствами GUI и при исполнении скриптов; в частности, доступ к выделенным объектам из скриптов, и наоборот, возможность выделения объектов из программного контекста.
6. Выполнение традиционных пространственных поисковых операций типа «поиск объектов по вхождению в некоторый объект».
7. Представление атрибутивной информации в SQL-базах данных, исполнение SQL-запросов из среды встроенного языка и интеграция значений SQL-выборки с поисковыми механизмами оболочки системы.

В данной работе основное внимание уделено описанию структуры языка в целом и функциональных возможностей языка, предназначенных для манипулирования графическими объектами, конфигурации GUI-интерфейса системы и интеграции с ним среды исполнения скриптов.

Объектная модель системы WinMap. Система WinMap является экспериментальной разрабатываемой геоинформационной системой. Архитектурно эта система представляет собой набор компонентов, встроенных в GUI-оболочку, которые реализуют базовые функции, к которым относятся представление в SQL БД пространственных и атрибутивных данных, растеризация графических объектов и их воспроизведение в графических окнах, и выполнение основных поисковых функций объектов по географическим и атрибутивным свойствам.

Система реализована на основе платформы Microsoft .NET и FRAMEWORK 2. Архитектура построена на основе объектно-контейнерной модели, представленной (в упрощенном виде) на рис. 1.

Термин «объектно-контейнерная модель» означает, что все структуры данных образуют иерархию вложенных одиночных объектов и контейнеров (коллекций) объектов. Каждый объект имеет собственные свойства, контейнеры с вложенными объектами и методы.

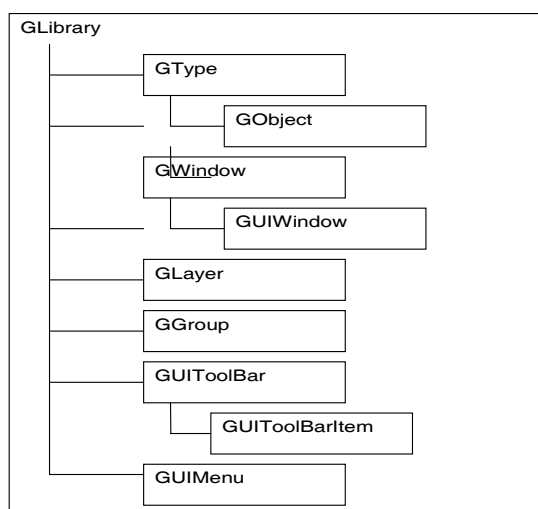


Рис. 1. Объектно-контейнерная модель системы WinMap

Объектом верхнего уровня является объект **GLibrary**, который соответствует экземпляру пространственной БД и описывает ее общие свойства. В него непосредственно вложена коллекция экземпляров **GType**, инкапсулирующие определения типов графических объектов, которые описывают их категорию (точечный, линейный и др.), параметры прорисовки, параметры масштабирования и т.д. Каждый экземпляр **GType** в свою очередь является контейнером для множества экземпляров графических объектов, непосредственно представляющих пространственные данные. Объекты **GWindow** описывают абстрагированные участки поверхности, воспроизводимой в конкретном графическом окне **GUIWindow**. Объект **GLayer** соответствует понятию слоя в традиционной терминологии ГИС-систем. Наконец, объекты **GGroup** являются коллекциями графических объектов вне зависимости от их типов, сгруппированными вместе по каким-либо семантическим признакам.

Типы **GUISMenu**, **GUIToolBar** и **GUIToolBarItem** представляют объекты для создания и конфигурации интерфейса при помощи меню и панелей инструментов, настраиваемых в GUI-интерфейс через языковую среду.

Кроме того, с экземплярами **GObject** ассоциированы записи с атрибутивной информацией; в данной статье языковые средства для манипулирования атрибутивной информацией и использование SQL-запросов подробно не рассматриваются.

Базовый сценарий работы с системой WinMap включает следующие этапы.

1. Загрузка в оболочке одного из экземпляров **GLibrary**, выполняемая при входе в систему, что сопровождается загрузкой вложенных контейнеров **GType** и **GWindow** и настройкой оконного интерфейса (меню и панелей инструментов).
2. Создание (по мере необходимости) на основании существующих экземпляров **GWindow** объектов **GUIWindow**, соответствующим графическим окнам с изображением объектов в соответствующем масштабе; при открытии окна происходит извлечение объектов из БД в кеш системы и растеризация в окне.
3. Интерактивные действия с воспроизведенными объектами, к которым относятся редактирование и создание новых объектов, поиск по различным критериям, переход от графических объектов к атрибутивной информации и наоборот, и др.

Скриптовый язык системы WinMap. Рассмотрим основные возможности представляемого языка. К основным типам относятся базовые типы для хранения числовых значений (Integer, Float), строк (String), значений-дат (DateTime) и специализированные типы. К ним относятся классы, перечисленные в предыдущем разделе, и некоторые другие, например: GPoint (точка в глобальных координатах) и GScreenPoint (точка в экранных координатах). Со значениями этих типов можно выполнять основные арифметические и строковые операции и операции сравнения. Для создания экземпляров специализированных классов используется оператор new вместе с соответствующим конструктором; далее объектная переменная может использоваться для вызова членов класса (методов и т.д.). Определение собственных классов не предусмотрено.

Управляющие конструкции включают условный оператор if/else, цикл for/break/continue, конструкция foreach, используемая для перебора элементов вложенных контейнеров, конструкция try/catch для обработки ошибочных ситуаций (исключений), и некоторые другие.

Вложенные контейнеры имеют унифицированную структуру и способы обработки, полностью соответствующие коллекциям в среде Framework. Так, для добавления нового элемента, удаления существующего и поиска используются стандартизированные методы Add(), Delete() и Find(), доступа по имени – индексное выражение «[<имя элемента>]» и т.д.

Исполнение кода скрипта выполняется в одном из двух режимов – в «присоединенном» режиме, когда скрипт выполняется в контексте уже открытого объекта GLibrary, или в автономном режиме, при котором создается или открывается еще не загруженный объект GLibrary, который далее может быть присоединен к оболочке.

Ниже приводятся фрагменты кода, иллюстрирующие возможности языка и способы его применения, комментарии к ним.

Пример 1. Создание пустого репозитория GLibrary и типа графического объектов GType, создание экземпляра этого типа с координатами, надписью и записью в атрибутивной таблице, и сохранение репозитория в БД.

```
GLibrary aCity = new GLibrary( dataProvider, "Карта города" );
GType streetLine = new GType( PolyLine, DefaultStyle );
aCity.Types.Add(streetLine );
GObject aStreet = StreetLine.Create(
    { new GPoint( 10010, 20020 ), new GPoint(10110, 20020 ) ,
      new GText( "ул. Парковая" ) );
aStreet.Attributes.Add( new SQLStreet( "улица", "Парковая", 5.23 ));
streetLine.Add( aStreet );
aCity.Save();
```

При создании экземпляра GLibrary используется вспомогательный объект dataProvider, инкапсулирующий параметры доступа к внешней базе данных для сохранения данных; в настоящее время поддерживаются системы Oracle, MS SQLServer и MS Access, а также собственный формат.

Экземпляр графического объекта в этом примере создается при помощи метода Create(), параметрами которого являются массив контрольных вершин полилинии, и объект GText, представляющий ассоциированную надпись. Кроме того, в коллекцию Attributes добавляется связанная запись с атрибутивной информацией, представленная объектом SQLStreet.

Скрипты подобного вида могут использоваться, в том числе для импорта данных; для этого достаточно автоматически создавать поток команд создания объектов.

Пример 2. Создание и модификация меню и панелей инструментов.

Конфигурация меню и панелей инструментов содержится в соответствующих контейнерах, вложенных в объект `GLibrary`, и соответственно хранятся в базе данных вместе с прочей информацией. После загрузки возможна модификация существующей конфигурации, которая сводится к перебору соответствующих коллекций `GUIMenu` и `GUIToolBar`, добавлению новых элементов и изменению или удалению существующих. При добавлении нового элемента указывается его название или идентификатор иконки и название файла со скриптом, который будет вызываться при вызове данного пункта меню или значка панели инструмента.

```
GUIMenu customexport = new GUIMenu( "Экспорт" );
aCity.Menus[ "File" ].Add( customexport );
customexport.Add(
    new GUIMenuItem( "в CSV", "/menuscripts/csvexport" ) );
```

Эти строчки кода используются для создания нового вложенного меню в существующем разделе `"File"` с пунктом указанного содержания.

Работа с панелями инструментов выполняется аналогичным способом; следующий фрагмент кода иллюстрирует создание новой «плавающей» панели инструментов.

```
GUIToolBar customtb = new GUIToolBar( "Выгрузка" );
aCity.Toolbars.Add(customtb);
aCity.Toolbars[ "Выгрузка" ].style = FloatStyle;
customtb.Add(new GUIToolBarItem(resourceID, "Экспорт в CSV",
    "/menuscripts/csvexport" ) );
aCity.UpdateGUI();
```

Последний метод используется для обновления интерфейса системы в соответствии с выполненными изменениями.

Пример 3. Доступ из среды исполнения команд скриптов к результатам поиска объектов и «выделенным» объектам.

Поиск объектов может выполняться средствами GUI-интерфейса системы по различным критериям, включая географические, атрибутивные, и т.д. Вне зависимости от использованного критерия и способа поиска, результат поиска – множество найденных объектов – представляется в виде коллекции, доступной из среды языка как объект типа `GGroup` (графическая коллекция), хранимой в свойстве `FoundGObject` текущего объекта `GLibrary`. Аналогичным образом для представления «выделенных» на графических окнах объектов предусмотрено свойство `SelectedGObjects`.

Концепция группы объектов, которую можно сохранять в пространственной базе данных и позднее вызывать для выполнения каких-либо действий с содержащимися в ней объектами, является новшеством GIS WinMap по сравнению с другими системами.

В следующем примере выполняется подключение к загруженному в систему репозиторию, перебор всех объектов всех типов и включение в группу выделенных объектов, которые попадают в указанный прямоугольник, а затем – воспроизведение на открываемом графическом окне выделенных объектов.

```

GLibrary aCity = GLibrary.Load(dataProvider, "Карта города" );
foreach(GType type in aCity.Types )
    foreach(GObject gobj in type.Objects )
        if( gobj.Inside( new GRect(10000, 20000, 1000, 1000 ) )
            Selected.Add( gobj );
GWindow gwin = FindSutableGWindow( Selected );
if( gwin == null )
    {
        ErrorMessage( "Нет подходящего географического окна" );
        return; }
GUIWindow guiwin = gwin.CreateGUIWindow( );
guiwin.Show();
GGroup tobosaved = new GGroup( Selected, "Объекты из квадрата" );
SavedGroups.Add( tobosaved );

```

Последние две строчки кода иллюстрируют возможность сохранения произвольной группы объектов в БД; сохраняются, естественно, не сами объекты, а сведения об их членстве в группе.

Пример 4. Разработка экспорта данных в произвольном формате.

Последний пример иллюстрирует доступ к найденным объектам и выгрузка из координат в отдельные файлы; имя файла формируется из названия типа и номера объекта.

```

foreach( GObject gobj in FoundGObject.GObjects )
{
    string filename = gobj.Type.Name + gobj.ID;
    TextFile tf = new TextFile( filename );
    foreach( GPoint point in gobj.Points )
        tf.WriteLine( point.X + ",\t" + point.Y );
    tf.Close();
}

```

Здесь коллекция Points графического объекта содержит пространственные координаты объекта.

Естественно, что подобным образом можно реализовать экспорт данных в разнообразных форматах.

Заключение. В работе описаны основные особенности встроенного скриптового языка для манипулирования пространственными данными геоинформационной системы WinMap, в частности, описаны общая структура языка, основные типы данных и способы работы с ними, и интеграция среды выполнения скриптов со средствами графической оболочки системы. Некоторые вопросы, в частности детальное описание методов работы с атрибутивной информацией, а также интеграция языка с SQL-запросами, осталась за рамками рассмотрения и нуждается в дополнительных исследованиях.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Шаши Шекхар, Санжей Чаула. Основы пространственных баз данных. – М.: Кудриц-Образ, 2004. – 327 с.
2. Amir H. Razavi. Arcview Gis/Avenue Developer's Guide. – OnWord Press. 1999. – 452 p.
3. Oracle Spatual Developer's Quide. Oracle. 2007. – 836 p.
4. Журавлев В. и др. (перевод). MapBasic. Среда разработки. Руководство пользователя. – New York, MapInfo Corp. 2000 г. – 285 с.

Статью рекомендовал к опубликованию д.ф.-м.н. В.Ф. Кравченко.

Заставной Дмитрий Александрович

Федеральное государственное образовательное учреждение высшего профессионального образования «Южный федеральный университет».

E-mail: pilidi@math.sfedu.ru.

344090, г. Ростов-на-Дону, ул. Мильчакова, 8а.

Тел.: +78632975111; факс: +78632975113.

Кафедра информатики и вычислительного эксперимента; к.т.н.; ассистент.

Zastavnoy Dmitry Aleksandrovich

Federal State-Owned Educational Establishment of Higher Vocational Education "Southern Federal University".

E-mail: pilidi@math.sfedu.ru.

8a, Milchakova Street, Rostov-on-the-Don, 344090, Russia.

Phone: +78632975111; fax: +78632975113.

The Department of Informatics and Computing Experiment; Cand. of Eng. Sc.; Assistant.

УДК 004.932.72'1

М.С. Соколов

**ПРИМЕНЕНИЕ АЛГОРИТМА ТОПОЛОГИЧЕСКОГО АНАЛИЗА
ПРИ РЕШЕНИИ ЗАДАЧ ГОРОДСКОГО ХОЗЯЙСТВА В МГИС**

Статья посвящена вопросам автоматизации решения задач городского хозяйства. Дано определение роли муниципальных и геоинформационных систем в составе единой информационной системы управления муниципалитетом. Сформулировано обоснование необходимости наличия конкретно двух этих подсистем и обосновано разграничение в обслуживании типов информации. Рассмотрен подход к применению топологических отношений при выполнении группового пространственного анализа картографических объектов. Предложено ввести понятие массива из матриц топологических отношений, хранящих взаимосвязи между объектами. Это описание позволяет выполнять быстрый отбор картографических объектов в соответствии с заданными пользователем правилами.

Топологическое отношение; муниципальная геоинформационная система; матрица топологических отношений; правило отбора; картографический объект.

M.S. Sokolov

**APPLICATION OF THE TOPOLOGICAL ANALYSIS ALGORITHM
AT THE DECISION OF MUNICIPAL ECONOMY PROBLEMS IN MGIS**

Article is devoted questions of automation for the decision of municipal economy problems. Definition of a role of municipal and geoinformation systems as a part of a uniform information control system of municipality is made. The substantiation of presence necessity two these subsystems is formulated and differentiation in service of types of the information is proved. The approach to application of topological relations is considered at performance of the group spatial analysis of cartographical objects. It is offered to enter concept of a file from matrixes of the topological relations storing interrelations between objects. This description allows to carry out fast selection of cartographical objects according to the rules set by the user.

The topological relation; municipal geoinformation systems; matrix of topological relations; selection rule; cartographical object.

Современные геоинформационные (ГИС) технологии находят применение в самых различных сферах деятельности человека, как в связи с его профессиональными потребностями в пространственной информации, так и познавательными потребностями [1,2].