

АСПЕКТЫ АВТОМАТИЗАЦИИ В ГИДРОМЕТЕОРОЛОГИЧЕСКОЙ СИСТЕМЕ ВИЗУАЛИЗАЦИИ ISOGRAPH

Ю.В. Алферов, В.В. Копейкин

*Гидрометеорологический научно-исследовательский
центр Российской Федерации
alferov@mecom.ru, help@tushino.com*

В течение нескольких лет в Гидрометцентре России проводится работа по развитию автоматизированной графической системы Isograph для визуализации метеорологических данных в операционных системах семейства MS Windows [1, 2]. К настоящему моменту система включает комплекс векторных географических данных, таких, как береговая линия материков и островов, внутренних водоемов, границы населенных пунктов, реки, дороги и т.д., и программные средства для их отображения, а также для визуализации метеоданных в форме полей и метеоданных, заданных в точке (наблюдения). При этом можно использовать несколько разных картографических проекций как для карт, так и при описании сеток полей метеоданных. Формы представления метеоданных также могут быть различными: изолинии, цветное закрашивание изоконтуров и нанеска значений или значков. Количество карт, одновременно размещенных на экране компьютера, не ограничено. Количество изобразительных слоев на одной карте, каждый из которых определяется некоторым видом данных, также не ограничено. Графическая система Isograph, таким образом, в комплексе с оперативными базами данных имеет признаки глобальной метеорологической геоинформационной системы (ГИС) [6].

Первоначально разрабатываемый графический комплекс предполагалось использовать для визуализации данных в научных исследованиях.

Программа была полностью интерактивной, т.е. управление изображением осуществлялось исключительно через меню и окна диалогов. Затем оказалось, что система полезна также и оперативным синоптикам для визуализации результатов численного прогнозирования погоды из внутренних баз данных Гидрометцентра России. С этого момента некоторые функции системы были автоматизированы, и определяющим принципом, строго соблюдающимся при разработке средств автоматизации, было облегчить работу с системой для пользователей-непрограммистов.

Для обеспечения ускорения построения карт в интерактивном режиме работы с системой Isograph используется три средства:

- возможность сохранения бланка карты в виде файла с параметрами региона и ссылками на необходимые географические данные;
- возможность сохранения заготовленных заранее комплектов параметров для последующей быстрой настройки изображения метеоэлементов, так называемые профили;
- возможность сохранения шаблонов карт с настроенным пользователем бланком и подобранными метеорологическими данными и профилями их изображения.

Все эти средства нацелены на использование пользователем-непрограммистом. Соблюдается принцип так называемого «визуального программирования». В данном случае этот процесс выглядит следующим образом. Пользователь строит карту, в интерактивном режиме подбирая параметры изображений и руководствуясь при этом стоящими перед ним задачами и собственным вкусом. На любом этапе работы над картой, когда пользователя удовлетворяет полученное изображение, ему необходимо вызвать соответствующее средство сохранения достигнутых параметров. Для сохранения бланка и профиля изображения метеоэлемента пользователю достаточно лишь назначить имя соответствующему объекту.

Процедура сохранения шаблона карты, т.е. полного состава изображительных слоев, более сложная. Затруднение возникает здесь из-за того, что система Isograph не имеет собственного хранилища данных, и доступ к метеоданным реализуется через механизм программного шлюза, как описывалось в [2]. Таким образом, виды запросов на метеоданные для различных хранилищ данных и, соответственно, для различных программных шлюзов оказываются различными, поэтому

и сохранить их автоматически не представляется возможным. При создании карты пользователем система Isograph «запоминает» имена примененных им бланка, шлюза и профилей расчерчивания метеоданных. Кроме того, в требованиях, сформулированных для разработчиков шлюзов, содержится пункт о необходимости сохранения запроса к данным, однозначно их определяющего, который затем передается вместе с другими необходимыми параметрами данных в ядро системы Isograph. Таким образом, в идеале к моменту сохранения карты система «знает» все обо всех используемых данных. То же относится и к данным, которые рассчитываются внутри Isograph. В этом случае запоминается информация о том, с помощью каких функций получены данные, из каких данных они получены и откуда те, в свою очередь, взяты. В результате при сохранении каждого изобразительного слоя пользователю предлагается своеобразная анкета, все графы которой уже заполнены, и достаточно лишь указать местоположение характеристик времени в соответствующих местах. Пример такого диалога для метеополя из оперативной базы данных Гидрометцентра России приведен на рис. 1. На рисунке в графах «анкеты» уже отмечены места для подстановки времени и даты при последующем вызове шаблона. Это группы букв, ограниченные символами решетки («#»). При реальной работе пользователю приходится выполнять эту разметку самостоятельно, что можно легко выполнить при помощи всплывающих меню. Здесь возникает лишь одна трудность: пользователю необходимо знать, где, какие и в каком формате временные параметры используются. Если названия карты и данных имеют достаточно произвольный вид, то форма запроса данных имеет строгую структуру, предусмотренную разработчиком шлюза. И, конечно, в этом случае за дополнительной информацией необходимо обратиться к разработчику либо к документации, предоставленной им.

Вызов всех описанных объектов производится системой после соответствующих команд оператора, подаваемых с помощью меню и окон диалогов. Вместе с тем пользователями системы уже давно высказывались пожелания организации полностью автоматического режима запуска системы для создания карт. Такой режим был бы полезен в оперативной работе для генерации карт по расписанию, для создания прогностического иллюстративного материала для веб-сайта и т.д. Если все перечисленные выше аспекты автоматизации могли быть

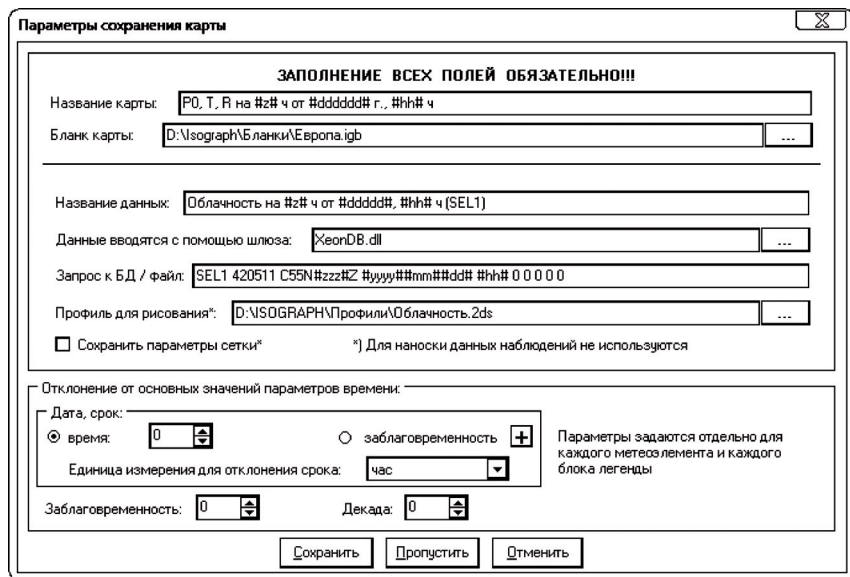


Рис. 1. Диалоговое окно с параметрами сохранения одного из изобразительных слоев шаблона карты.

максимально переложены на плечи разработчиков Isograph, то решение последней задачи предполагает со стороны пользователя системы разработку некоторых сценариев создания и обработки карт, анализ нестандартных ситуаций и т.п., а также от разработчиков – реализацию синтаксического разбора таких сценариев и их исполнения. Поэтому, к сожалению, от пользователя системы требуется способность составить план создания карт, манипулирования ими, реагирования на различные события и т.п., т.е. владения навыками программирования.

Итак, для автоматического создания карт и манипулирования ими разработан несложный язык программирования. При этом предполагается, что создаваемые карты были ранее сохранены в форме шаблонов и путем действий, описанных выше. Ядром этого языка являются операторы цикла с числовым счетчиком итераций и по списку строк. Допускается значительная глубина вложенности циклов. Также для реализации возможности анализа событий добавлен оператор условия.

Переменные внутри программы (скрипта) на этом языке не требуют описания. Помимо стандартных и для других языков программирования числовых и строковых констант, введены специальные константы вида год/мес/день/час/мин/сек для обозначения даты и времени. Допустимо сокращать конечную часть такой константы, например, 2011/04/15 – 15 апреля 2011 года. Также введены константы целого типа с постфиксной литерой для обозначения временных единиц: у – годы, m – месяцы, d – дни, h – часы, i – минуты, s – секунды, например, 2d – два дня (двое суток), 3m – три месяца и т.п. Из общераспространенных операторов есть еще, конечно, оператор присваивания. Определены также специальные операторы, реализующие известную функциональность Isograph: создание и удаление карты, печать и сохранение карты, выстраивание карт рядами. Также есть арифметические операторы-функции. Есть операторы получения и преобразования даты и времени. Также имеются некоторые другие служебные операторы. Таким образом, основное назначение операторов языка – создание карт по готовым шаблонам и манипуляции ими. Полный список зарезервированных слов языка и их значение приведены в таблице.

Таблица

Зарезервированные слова языка манипулирования картами

Функции манипулирования картами и переменными	
Create	Создание карты
Destroy	Удаление последней созданной карты
DestroyAll	Удаление всех созданных карт
BuildToRow	Выстраивание карт рядами
Export	Запись последней созданной карты в файл графического формата
ExportAll	Запись всех созданных карт в файл графического формата
Print	Печать последней созданной карты
PrintAll	Печать всех созданных карт
ClearVarList	Очистка списка переменных скрипта
ClearMapList	Очистка списка карт скрипта без уничтожения самих карт
DeleteFromList	Удаление карты из списка без ее уничтожения
MapError	Статус последней созданной карты: 0 – карта создана успешно, 1 – создана карта с неполным составом слоев, 2 – карта не может быть создана.

Продолжение таблицы

Операторы для работы с датой и временем	
Now	Текущие поясные дата и время (из компьютера)
UTC	Текущие дата и время по ВСВ (при условии правильной настройки операционной системы пользователя)
Srok	Ближайший к заданному времени прошедший синоптический срок
Управляющие инструкции	
If	Оператор условия
EndIf	Окончание оператора условия
For	Оператор цикла
Next	Окончание оператора цикла, переход на следующую итерацию
Break	Выход из текущего цикла
Halt	Немедленное прекращение выполнения скрипта без закрытия Isograph
Exit	Немедленное прекращение выполнения скрипта и закрытие Isograph
Арифметические операторы	
Inc	Сложение
Dec	Вычитание
Mult	Умножение
Div	Деление
Trunc	Целая часть числа
Frac	Дробная часть числа
Abs	Абсолютная величина числа
Дополнительные зарезервированные символы	
!	Комментарий
#	Ссылка на параметр скрипта
Дополнительные слова для реализации цикла и условия	
To	Составляющая арифметического цикла. Пример: For A=1 To 5 Step 2
Step	Составляющая арифметического цикла
Of	Составляющая цикла по списку строк. Пример: For A of (B, "text2", C)
Not	Отрицание логической переменной
Then	Составляющая оператора условия

Логика обработки скрипта следующая. При его запуске создается пустой список карт. При каждом выполнении команды создания карты вызывается соответствующая процедура в Isograph, а ссылка на новую карту добавляется в список карт скрипта. В процессе работы возможно использование как команд работы с комплектом карт, так и команд работы с последней созданной картой. Карты можно сохранять в файл, выводить на печать и удалять из Isograph. Также возможна модификация комплекта карт командой удаления из него ссылки на последнюю созданную карту либо полной очисткой списка карт. При этом из Isograph карты не удаляются. Полная очистка списка карт используется для начала работы с формально новой группой карт. Удаление из списка карт последней созданной карты позволяет при необходимости исключить карту, созданную с ошибкой.

Переменные скрипта также попадают в список переменных. Тип переменной определяется по ее первому упоминанию. Изменять тип переменной внутри скрипта нельзя, если не вызвать ClearVarList, но в этом случае потеряются значения всех переменных. Таким образом, фактически после вызова ClearVarList и ClearMapList внутри того же файла начинается новый скрипт.

Запуск скрипта в пакетном режиме может быть осуществлен командой:

```
Ig.exe "<имя скрипта>" "<параметры>"
```

Здесь Ig.exe – имя программы Isograph, например:

```
Ig.exe "Европа" "date=2010/12/01 z1=24 z2=72 stp=12"
```

Запуск подобной задачи по расписанию выполняется при помощи штатного Планировщика заданий MS Windows.

Был разработан интерпретатор этого языка. Основным предназначением интерпретатора является синтаксический разбор текста программы (скрипта), распознавание конструкций языка, детектирование ошибок и указание на них, а также вызов соответствующих функций графической системы в случае отсутствия ошибок.

Итак, интерпретирование скрипта представляет собой синтаксический анализ текста путем выделения из него последовательности предложений языка, после чего интерпретатор выполняет действие,

соответствующее данному предложению. Выделение предложений производится путем сравнения последовательности элементарных конструкций, выделяемых из текста, по типам с последовательностями элементарных конструкций, указанными в шаблонах предложений языка [3–5, 7, 8].

Выделение элементарных конструкций из текста скрипта производится при помощи двух модулей. Первый модуль осуществляет получение очередного символа из буфера текста скрипта и помещает этот символ в буфер символа. Если буфер текста скрипта оказывается пуст, то модуль устанавливает флаг конца текста. Второй модуль осуществляет сборку элементарных конструкций из символов, находящихся в буфере символа.

Сборка элементарных конструкций производится путем добавления очередного символа к уже частично собранной элементарной конструкции до тех пор, пока не встретится символ, недопустимый для данного типа конструкции. После чего конструкция считается полностью собранной. Тип элементарной конструкции определяется в процессе ее сборки.

В процессе выделения из текста скрипта как элементарных конструкций, так и предложений неявно производится проверка текста скрипта на наличие синтаксических ошибок, в случае обнаружения которых генерируется соответствующее сообщение.

С целью облегчения написания скрипта для пользователя-непрограммиста была создана специальная программа Генератор скриптов (рис. 2). Окно программы содержит несколько панелей, в которых можно как набирать текст вручную (для опытных пользователей), так и создавать его путем перетаскивания иконок команд и заполнения «анкеты» параметров команды, что удобно для непрограммистов. При втором способе структура программы более наглядна, а текст ее генерируется автоматически в соответствии со структурой, получающейся при работе пользователя.

Основные возможности Генератора скриптов:

1. Разработка скрипта при помощи графического интерфейса:

- скрипт отображается в виде дерева. В таком представлении четко прослеживается вложенность структур. Текст может быть легко переконструирован благодаря возможности перемещения структур любой сложности внутри дерева;

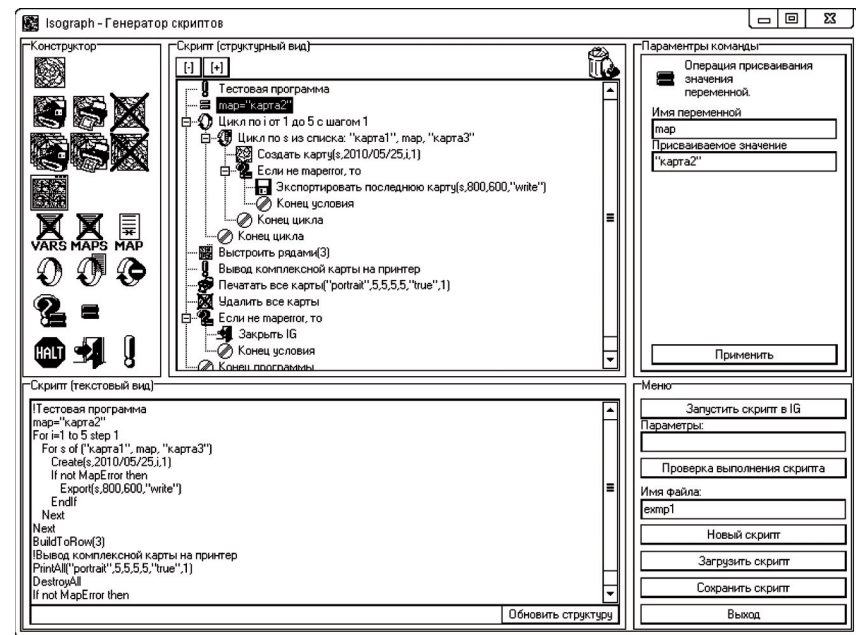


Рис. 2. Окно Генератора скриптов.

- добавление новой команды производится путем перетаскивания соответствующего значка, расположенного на панели «Конструктор», в нужное место дерева. Такой способ исключает возможность появления синтаксических ошибок при написании команд. Также отпадает необходимость помнить названия и синтаксис написания команд;

- редактирование параметров команды осуществляется при помощи диалоговой панели «Параметры команды». После изменения всех необходимых параметров и нажатия кнопки «Применить» происходит проверка корректности задания параметров;

- значки ветвей дерева, соответствующие различным командам, добавляют наглядности тексту;

- в окне «Скрипт (текстовый вид)» при каждом изменении текст скрипта автоматически модифицируется.

2. Набор текста скрипта вручную:
– возможно редактирование текста скрипта вручную в окне «Скрипт (текстовый вид)»;

– после внесения очередного изменения в текст есть возможность отобразить его в дереве скрипта путем нажатия кнопки «Обновить структуру». Одновременно будет произведена проверка синтаксиса написания команд и их параметров, а также вложенности структур.

3. Проверка выполнения скрипта:

– при нажатии на кнопку «Проверка выполнения скрипта» будет произведена полноценная проверка скрипта путем эмуляции работы скрипта в программе Isograph.

Такая проверка обеспечивает контроль правильности использования переменных и значений параметров.

4. Программа имеет возможность создания, сохранения и загрузки скриптов.

Итак, система визуализации Isograph имеет в своем составе средства для автоматизации действий пользователя как в интерактивном режиме работы, так и позволяющие производить расчет карт в пакетном режиме обработки данных. Все эти средства разработаны с учетом необходимости облегчить использование их для пользователей-непрограммистов.

Работа выполнялась при финансовой поддержке Федерального агентства по науке и инновациям в рамках Федеральной целевой программы «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2012 гг.» по составной части темы «МЕГАПОЛИС – интегрированные технологии оценки загрязнения атмосферы крупных городов в региональном и глобальном масштабах на основе аэрокосмического и наземного мониторинга для уменьшения негативных последствий антропогенных воздействий».

Список использованных источников

1. Алферов Ю.В. Принципы построения автоматизированной графической системы для визуализации полей метеоэлементов в научных исследованиях // Труды Гидрометцентра России. – 2000. – Вып. 334. – С. 180–189.

2. Алферов Ю.В. Автоматизированная графическая система для визуализации результатов численных прогнозов // Труды Гидрометцентра России. – 2003. – Вып. 338. – С. 119–124.

3. Ахо А.В., Сети Р., Ульман Д.Д. Компиляторы: принципы, технологии и инструменты. – М.: Вильямс, 2003. – 768 с.

4. Вирт Н. Построение компиляторов. – М.: ДМК Пресс, 2010. – 192 с.

5. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб.: Питер, 2001. – 368 с.

6. Геоинформатика. Толковый словарь основных терминов / под ред. А.М. Берлянта и А.В. Кошкарева. – М.: ГИС-Ассоциация, 1999. – 204 с.

7. Свердлов С.З. Языки программирования и методы трансляции. – СПб.: Питер, 2007. – 637 с.

8. Шилдт Г. Теория и практика C++. – СПб.: BHV-Санкт-Петербург, 2001. – 416 с.

Поступила в редакцию 20.05.2011 г.