

УДК 550.9:004.65

© К.В. Деев

К.В. Деев

ПЕРСПЕКТИВЫ РАЗВИТИЯ ГИС INTEGRO

Введение

В настоящее время во ФГБУ «ВНИГНИ» продолжается разработка и совершенствование программного комплекса ГИС INTEGRO. Это специализированная геоинформационная система, предназначенная для решения информационно-аналитических и прогнозных задач в области природопользования. Система позволяет работать с двумерными картами и трехмерными моделями геологических объектов. В отличие от ГИС общего назначения, в ГИС INTEGRO особое внимание уделено работе с геологическими объектами: разрезами, профилями и информацией по скважинам. Для отображения этих данных используются специальные окна просмотра. Окна для отображения информации по профилям позволяют показывать данные вдоль разреза в разных масштабах по осям глубины и длины вдоль профиля. Данные по скважинам масштабируются только вдоль оси глубины. Данные о местоположении линии профиля и устья скважины используются для синхронизации отметок курсора в этих окнах и в окне отображения обычной карты, а также при построении и отображении трехмерных моделей. В системе есть модули и утилиты для отображения данных гравитационной, магнитной и сейсморазведки, а также для обработки потенциальных полей. Демонстрационную 30-дневную версию системы можно скачать на сайте [1]. Работы по совершенствованию системы продолжаются. В данной статье рассматриваются четыре направления развития ГИС INTEGRO:

- 1) взаимодействие с базами данных,
- 2) внедрение скриптового языка,
- 3) импорт проектов ArcMap,
- 4) работа с первичными данными.

1. Взаимодействие с базами данных

Для хранения и обработки больших объемов данных в компьютерных технологиях повсеместно используют реляционные базы данных. Использование реляционных баз данных полезно по многим причинам. Оно способствует структурированию информации, обладает унифицированным быстрым доступом к данным посредством SQL-запросов, позволяет одновременно читать и записывать данные разным пользователям (многопользовательский режим), заботится о сохранности и согласованности данных с помощью механизмов триггеров и транзакций.

До недавнего времени базы данных использовались для хранения данных в виде текста и чисел. Оценив удобство подобного хранения информации, специалисты стали помещать в базы данных и картографические данные, и средства работы с ними. На сегодняшний день многие базы данных имеют встроенные средства или специальные программные расширения для работы с картографическими данными. К сожалению, устоявшегося стандарта, позволяющего единообразно работать с картографическими данными в базах данных разных производителей, нет. Будем надеяться, что он появится в дальнейшем и позволит работать с картографическими данными, подобно тому как SQL позволяет работать с числовыми и текстовыми данными.

На сегодняшний день, поддержка хранения картографических данных в реляционных базах данных устроена следующим образом. База данных позволяет в таблицах создавать поля специального типа для хранения картографических объектов. Каждая запись в таблице хранит геометрию картографического объекта в таком поле. Геометрия объектов чаще всего записываются в WKB-формате [2]. Остальные поля записи хранят числовые, текстовые, временные и т.д. характеристики картографического объекта. Для поля, в котором хранятся картографические объекты, чаще всего можно построить пространственный индекс. Пространственный индекс ускоряет некоторые геометрические операции над картографическими объектами, например быстрый выбор объектов, попадающих в прямоугольную область. Для работы с картографическими данными используется либо специализированный программный интерфейс базы данных, либо расширение языка SQL. Информация о географической проекции картографических данных обычно хранится в служебных таблицах базы данных. Географическая проекция часто записывается в виде кодов EPSG. Также в служебных таблицах обычно содержится перечень всех геометрических полей, таблицы, в которых они используются, и другая служебная информация.

В ГИС INTEGRO также возникла необходимость создать программные модули для работы с картографическими данными, хранящимися в реляционных базах данных. Для доступа к таким данным предлагается использовать библиотеку GDAL [3]. Эта библиотека с открытым исходным

кодом обладает целым рядом преимуществ. Она поддерживает большое количество разнообразных форматов хранения картографических данных (в том числе и в реляционных базах данных), бесплатна, активно развивается и поддерживается сообществом разработчиков программного обеспечения. Важным плюсом является то, что все различия в реализации хранения и доступа к картографическим данным скрыты за единым программным интерфейсом библиотеки. Это значительно снижает трудозатраты, но отнюдь не до нуля.

На сегодняшний день в ГИС INTEGRО разработана и тестируется первая версия редактора векторных данных, позволяющая работать со слоями, хранящимися в базе данных. Эта версия позволяет редактировать картографические объекты из базы данных, изменяя их геометрию и атрибуты. Измененные объекты сохраняются во внутреннем буфере программы. При завершении сеанса редактирования пользователь может записать эти изменения в базу или полностью отказаться от них. Для редакции геометрии используется тот же набор инструментов, что и для редакции геометрии данных, хранящихся в SHAPE-файле. Эти инструменты позволяют совместно редактировать сразу несколько слоев исходных данных. С их помощью можно удалять старые и создавать новые объекты, изменять форму объектов, поддерживая при этом топологическую корректность и совместимость редактируемых слоев (совместные границы должны быть одинаковы в разных объектах, полигоны не должны накладываться друг на друга, линейные объекты в месте пересечения должны иметь общую точку и т.д.). Однако первая версия редактора имеет некоторые недостатки, которые необходимо далее исправить. Во-первых, первая версия редактора не позволяет отменять изменения, сделанные пользователем в сеансе редактирования. Если мы работаем в ГИС INTEGRО с геометрией, сохраненной в SHAPE-файле, то в процессе редактирования мы можем сделать один или несколько «шагов» назад, и данные вернуться в состояние до применения последних операций редактирования. При работе с базой данных мы такой возможности пока не имеем. Во-вторых, редактор атрибутов данных, хранящихся в базе, нуждается в доработке. Пока что он не позволяет легко выбирать несколько объектов, не отображает информацию о количестве выбранных объектов, не позволяет упорядочивать объекты в соответствии со значениями атрибутов и т.д.

Работы по доработке редактора уже ведутся разработчиками ГИС INTEGRО. Для удобной работы с картографическими данными, хранящимися в базе данных, предстоит сделать следующие доработки:

1. Помимо инструментов просмотра и редакции уже существующих картографических слоев в базе данных, ГИС INTEGRО необходимы инструменты, позволяющие создавать новый слой в базе данных, и инструменты, позволяющие импортировать существующие картографические данные в базу.
2. В данный момент разработка и тестирование проводится с использованием базы данных PostgreSQL [4]. Желательно, чтобы ГИС INTEGRО работала с как можно большим числом различных баз данных. Используемая библиотека GDAL предоставляет унифицированный доступ к различным базам данных, но наверняка есть какая-то специфика, неучтенная в процессе разработки. Поэтому разрабатываемые средства нужно протестировать и доработать для других баз данных. В первую очередь это нужно сделать для наиболее распространенных баз данных MS SQL Server [5] и Oracle [6].
3. В ГИС INTEGRО существует много утилит, работающих с векторными данными. Подавляющее большинство из них работают с данными в формате SHAPE. Постоянно конвертировать данные из базы в SHAPE-формат и обратно неудобно. Нужно переделать существующие утилиты таким образом, чтобы они работали с данными базы напрямую или конвертировали их в SHAPE-формат в фоновом режиме.

2. Внедрение скриптового языка

В процессе обработки картографических данных зачастую возникают ситуации, когда встроенных в ГИС средств обработки данных не хватает. Например, есть отдельные процедуры обработки данных, которые для получения конечного результата нужно последовательно вызывать одну за другой. Другой пример: есть набор файлов исходных данных, и к каждому файлу нужно применить одну и ту же процедуру обработки. В этих случаях хочется написать небольшую программу, которая выполнила бы эти действия. Такая программа называется скриптом.

Для написания скриптов могут использоваться разные языки программирования. Для внедрения в ГИС INTEGRО разумнее всего, по-моему, использовать Python [7]. Назовем несколько причин, по которым выбор языка Python кажется оправданным.

1. Простой синтаксис. Это значит, что язык легко освоить.
2. Наличие большого количества программных библиотек для научных вычислений. В силу разных причин сложилась такая ситуация, что Python стал общепринятым языком для научных

вычислений. Поэтому многие библиотеки для обработки числовых данных написаны или имеют интерфейсы на языке Python.

3. Просто встраивать. Разработчики Python приложили усилия, чтобы интерпретатор языка можно было встраивать в другие приложения.
4. Python – популярный язык программирования. Это означает, что, возможно, пользователь уже знаком с этим языком. Кроме того, существует много обучающей литературы по языку и огромное сообщество, использующее Python.

Для проверки возможности встраивания интерпретатора Python в ГИС INTEGRO была написана небольшая тестовая утилита. Она позволяет подключить интерпретатор Python из популярного менеджера пакетов языка программирования Python-Miniconda3 [8]. Это весьма удобно, так как этот менеджер позволяет легко устанавливать множество других Python-библиотек, в том числе и библиотек для обработки данных, без всякой дополнительной настройки. В рамках работы над этой утилитой написана небольшая подпрограмма на Python. Она позволила получить доступ программным объектам ГИС INTEGRO из языка Python. Для связи с объектами ГИС INTEGRO, имеющими интерфейс на языке Си, используется пакет CTYPES [9]. Альтернативные пути (использование Cython, Boost Python library, Python SDK) сложнее и требуют строгой привязки к конкретной реализации интерпретатора Python. Тестовая утилита позволила выполнить скрипт, в котором создавался геометрический объект ломаная линия, в нее добавлялись точки с заданными координатами, и полученная ломаная сохранялась в SHAPE-файл. Для проверки связи с другими модулями Python тестовая утилита выполнила скрипт, который с помощью директивы «import» подключил модули Numpy и Matplotlib, часто используемые специалистами по анализу данных, создал двумерный массив, заполнил его случайными значениями и отобразил его в виде точек на графике в отдельном окне. Однако, для того чтобы перейти от тестовых разработок к настоящей работе со скриптами Python в ГИС INTEGRO, нужно сделать еще очень многое.

Список дальнейших работ в этом направлении выглядит следующим образом:

1. Продумать и реализовать на Python интерфейс доступа к объектам ГИС INTEGRO. Вероятно, для этого придется написать промежуточные (Proxy) объекты на Си++ и уже их импортировать в Python. Это придется сделать потому, что исходный интерфейс объектов ГИС INTEGRO слишком сложен, чтобы пользоваться им напрямую из интерпретатора Python.

2. Нужно предоставить пользователю, который будет писать скрипт, описание объектов ГИС INTEGRO и возможных операций над ними, то есть справочную документацию. Для этого можно воспользоваться программой Doxygen [10], которая из специально отформатированных комментариев к исходному коду позволяет создать справочную документацию по программному интерфейсу.
3. Усовершенствовать пользовательский интерфейс работы со скриптами на Python. На данный момент в тестовой утилите пользователь может в поле ввода написать несколько строк на Python и выполнить их, нажав на кнопку «Выполнить». Нужно уметь загружать и сохранять скрипты на диск. Хорошо бы уметь подсвечивать синтаксис (хотя бы ключевые слова).

3. Импорт проектов ArcMap

Большое количество работ с картографическими данными проводилось и проводится в программе ArcMap фирмы ESRI. Естественно, хочется иметь возможность импортировать в ГИС INTEGRO проекты, построенные в этой программе.

Обе ГИС имеют сходную структуру картографического проекта. Картографический проект состоит из набора карт. Карта состоит из набора слоев. Слои представляют собой геометрические данные, связанные с ними атрибуты и правила их отображения.

Геометрические данные и связанные с ними атрибуты могут быть использованы напрямую или переведены в другой формат средствами ArcMap или ГИС INTEGRO. При этом ГИС INTEGRO использует библиотеку GDAL, упомянутую выше.

Немного сложнее обстоят дела с импортом растровых данных. Если в проекте используются растры в обычных растровых форматах (TIFF, JPEG, BMP и т.д.), то ГИС INTEGRO может подключать такие данные напрямую. Если же растр сохранен в базе геоданных GDB, собственной разработки фирмы ESRI, то доступ к ним можно получить на текущий момент только с помощью экспорта из программы ArcMap.

Остальная информация из картографического проекта: список карт, их названия, слои из которых они состоят, группировка слоев, стили отображения объектов в этих слоях – при этом не импортируется. Это большая проблема сегодняшних ГИС-технологий. Нет унифицированных способов передать эту информацию.

К сожалению, формат файла картографического проекта ArcMap является закрытым. Фирма ESRI не предоставляет его описания. Однако программисты

из открытого сообщества расшифровали некоторые части этого формата и выложили эту информацию в открытый доступ [11]. Основываясь на этих данных, в ГИС INTEGRО была создана утилита, частично импортирующая простые проекты из ArcMap.

Существующая версия этой утилиты позволяет импортировать общую структуру проекта и слои, основанные на SHAPE-файлах. Слои импортируются со стилевым оформлением, но без настроек автоподписей. Для доработки этой утилиты необходимо:

1. Тестировать утилиту на файлах картографических проектов разного состава и разных версий программы ArcMap. При этом стараться исправить выявленные ошибки и неточности.
2. Научиться импортировать настройки автоподписей слоев.
3. Научиться импортировать настройки растровых слоев.
4. Научиться импортировать настройки слоев проекта, основанные на данных из базы геоданных ArcMap – GDB.
5. Научиться импортировать статические подписи и графику из проектов ArcMap.
6. Попробовать импортировать растровые данные из базы геоданных GDB напрямую, без использования ArcMap. При реализации этого пункта можно основываться на работе [12], содержащей сведения о способе хранения растровых данных в GDB-файлах.

Полного описания формата картографического проекта нет. Работа ведется во многом «на ощупь». Поэтому полной уверенности, что эти работы удастся полностью выполнить, нет. Небольшой опыт использования существующей версии утилиты дает некоторую надежду. С ее помощью были удачно импортированы несколько простых файлов проекта ArcMap. Эти проекты содержали только слои, созданные на основе SHAPE-файлов и растровые слои. Растровые слои игнорировались, а векторные вместе со стилевым оформлением были корректно импортированы в ГИС INTEGRО.

Однако при импорте большого проекта с данными, хранящимися в разных источниках (в том числе в базе геоданных), нас постигла неудача. В этом проекте было много слоев, основанных на данных из GDB. Многие из них были распознаны программой импорта некорректно. Это повлекло за собой сбой в программе импорта. Сейчас ведется работа над исправлением этих ошибок.

4. Работа с первичными данными

Под первичными данными здесь понимается информация, которую получает геолог в ходе

непосредственных наблюдений на исследуемой территории. То есть это некоторый набор точек наблюдения, иногда объединенных в маршрут наблюдений. О каждой точке есть некоторая фиксированная информация: координаты точки, время проведения наблюдений, данные о наблюдателе и др. – и данные о результатах наблюдений. Это может быть текстовое описание, измерение какого-либо параметра, фотография с места наблюдения и т.д. Для хранения этой информации во ВСЕГЕИ была разработана программа Sherpa [13]. Эта программа сохраняет данные в базу данных, созданную на основе компактной встраиваемой СУБД с открытым исходным кодом SQLite [14]. Естественно, хочется иметь доступ к этой информации из ГИС INTEGRО. Для достижения этой цели необходимо выполнить следующие работы:

1. Создать модули для удобной работы с СУБД SQLite. В ГИС INTEGRО уже есть модуль для чтения данных из этой СУБД. Нужно его развить и дополнить функциями записи данных.
2. Разобраться в структуре базы данных первичной информации, разработанной во ВСЕГЕИ. Ситуация осложняется тем, что существующий вариант структуры еще не окончательный и подвергается во ВСЕГЕИ небольшой доработке.
3. Разработать для ГИС INTEGRО программные модули, которые позволяли бы создавать специальные картографические слои, отображающие точки наблюдений и линии маршрутов из базы данных первичной информации.
4. Разработать окна интерфейса пользователя для отображения информации о результатах наблюдений в данной точке.
5. Разработать окна интерфейса пользователя для редакции и пополнения базы первичных данных.

Заключение

В данный момент работа ведется по всем четырем направлениям развития ГИС INTEGRО. Наиболее важными и востребованными из них являются «взаимодействие с базами данных» и «импорт проектов ArcMap». На этих направлениях уже созданы первые работающие модули, находящиеся на этапе тестирования. К середине этого года мы планируем выпустить версию ГИС INTEGRО, позволяющую работать с существующими слоями в базе PostgreSQL, а к концу года сделать первые модули для создания и импорта новых слоев в эту базу данных. С направлением «импорт проектов ArcMap» ситуация менее ясная. Первая тестовая версия утилиты импорта уже находится в общем доступе. Однако оценить

трудозатраты по ее доработке достаточно сложно по причинам, описанным выше.

Направления «внедрение скриптового языка» и «работа с первичными данными» находятся пока на этапе перехода от экспериментальных разработок к рабочим модулям. Первые тестовые варианты рабочих модулей ожидаются к середине года.

Ключевые слова: ГИС INTEGR0, базы данных, Python, первичная геологическая информация.

ЛИТЕРАТУРА

1. <http://www.gis-integro.ru/support/> (дата обращения: 24.01.2020).
2. https://en.wikipedia.org/wiki/Well-known_text_representation_of_geometry (дата обращения: 24.01.2020).
3. <https://gdal.org> (дата обращения: 24.01.2020).
4. <https://postgis.net> (дата обращения: 24.01.2020).
5. <https://docs.microsoft.com/ru-ru/sql/relational-databases/spatial/spatial-data-sql-server?view=sql-server-ver15> (дата обращения: 23.01.2020).
6. <https://www.oracle.com/database/technologies/spatialandgraph.html> (дата обращения: 23.01.2020).
7. <https://www.python.org> (дата обращения: 22.01.2020).
8. <https://conda.io/en/latest/miniconda.html> (дата обращения: 24.01.2020).
9. <https://python-scripts.com/extending-python-with-c-libraries> (дата обращения: 23.01.2020).
10. <http://www.doxygen.nl> (дата обращения: 22.01.2020).
11. <https://github.com/nyalldawson/slyr> (дата обращения: 22.01.2020).

12. https://github.com/rouault/dump_gdbtable/wiki/FGDB-Spec (дата обращения: 21.01.2020).
13. https://vsegei.ru/ru/info/normdocs/prog_ggk200-ggk1000/sherpa/ (дата обращения: 22.01.2020).
14. <https://www.sqlite.org/index.html> (дата обращения: 22.01.2020).

REFERENCES

1. <http://www.gis-integro.ru/support/> (date of access: 24.01.2020).
2. https://en.wikipedia.org/wiki/Well-known_text_representation_of_geometry (date of access: 24.01.2020).
3. <https://gdal.org> (date of access: 24.01.2020).
4. <https://postgis.net> (date of access: 24.01.2020).
5. <https://docs.microsoft.com/ru-ru/sql/relational-databases/spatial/spatial-data-sql-server?view=sql-server-ver15> (date of access: 23.01.2020).
6. <https://www.oracle.com/database/technologies/spatialandgraph.html> (дата обращения: 23.01.2020).
7. <https://www.python.org> (date of access: 22.01.2020).
8. <https://conda.io/en/latest/miniconda.html> (date of access: 24.01.2020).
9. <https://python-scripts.com/extending-python-with-c-libraries> (date of access: 23.01.2020).
10. <http://www.doxygen.nl> (date of access: 22.01.2020).
11. <https://github.com/nyalldawson/slyr> (date of access: 22.01.2020).
12. https://github.com/rouault/dump_gdbtable/wiki/FGDB-Spec (date of access: 21.01.2020).
13. https://vsegei.ru/ru/info/normdocs/prog_ggk200-ggk1000/sherpa/ (date of access: 22.01.2020).
14. <https://www.sqlite.org/index.html> (date of access: 22.01.2020).