

***Инструментальные
средства
информационных
технологий***

Рассматриваются особенности и принципы построения версии 2.0 подсистемы ввода и редактирования геометрической информации автоматизированного комплекса НАДРА-3D.

© М.В. Белоус, В.С. Дейнека,
2009

УДК 004.925.8

М.В. БЕЛОУС, В.С. ДЕЙНЕКА

**ПОДСИСТЕМА ВВОДА
И РЕДАКТИРОВАНИЯ ГЕОМЕТРИЧЕСКОЙ
ИНФОРМАЦИИ ПРОСТРАНСТВЕННЫХ
СЛОИСТЫХ ТЕЛ ИНФОРМАЦИОННОЙ
ТЕХНОЛОГИИ НАДРА-3D**

Введение. В Институте кибернетики имени В.М. Глушкова НАН Украины разработан программный комплекс НАДРА-3D, предназначенный для моделирования процессов фильтрации жидкости, изменения напряженно-деформированного состояния в многокомпонентных трехмерных объектах, которые могут содержать произвольно ориентированные в пространстве тонкие включения [1 – 4], в том числе и пересекающиеся [5].

Комплекс состоит из двух частей – интерфейса пользователя (функционирует на персональном компьютере) [4] и решателя (функционирует на многопроцессорном комплексе СКИТ Института кибернетики имени В.М. Глушкова НАН Украины) [6].

В данной работе описана новая версия подсистемы ввода и редактирования геометрической информации (ПВиРГИ 2.0) интерфейсной части комплекса, разработка которой вызвана необходимостью адаптации системы НАДРА-3D для моделирования процессов в грунтовых средах, в которых ведется добыча полезных ископаемых. В частности – для моделирования влияния угольных шахт на процессы фильтрации жидкости и изменения напряженно деформированного состояния грунтового массива.

В версии 2.0 составляющей ПВиРГИ системы НАДРА-3D добавлены новые модели описания геометрии исследуемого объекта, расширены возможности подсистемы графического отображения информации, подсистемы управления инструментарием, расширено множество специализированных инструментов.

Основные подсистемы программного комплекса.

При запуске программы создается главная форма приложения и объект класса `nEngine`, реализующий системную компоненту интерфейсной части программного комплекса НАДРА-3D (рис. 1). Объект класса `nEngine` содержит системные переменные и указатели на объекты, реализующие различные подсистемы комплекса. Указатель на объект этого класса также содержится в большинстве подсистем комплекса. Таким образом, практически из любого места программы обеспечивается доступ к возможностям основных подсистем.

Во время работы программы события, генерируемые главной формой (например, нажатие клавиши клавиатуры, изменение положения курсора мыши, или перерисовка экрана) передаются объекту класса `nEngine` и далее обрабатываются соответствующей подсистемой.

Графическая подсистема.

Программный интерфейс графической подсистемы реализован в виде класса `nRenderer`. `nRenderer` – абстрактный класс, т. е. объявленные в нем методы определяются только в его потомках. Это позволяет использовать различные реализации подсистемы отображения графики (например, созданные самостоятельно, или с использованием графических библиотек OpenGL, [7] DirectX), имеющие одинаковый интерфейс вызова. Таким образом достигается независимость подсистем комплекса, использующих графическую подсистему, от деталей ее реализации.

Подсистема хранения данных о геометрии исследуемой области.

В программном комплексе НАДРА-3D геометрическая модель представляется как набор непересекающихся восьмиугольных зон, заполняющих пространство модели. При этом соседние восьмиугольники имеют общие вершины-узлы, ребра и стороны. На первом этапе создания геометрии модели формируется трехмерная прямоугольная сетка, состоящая из некоторого количества слоев n , m , k по каждой из осей координат x , y , z , после чего производится ее деформация путем применения операций перемещения, вращения, масштабирования к различным частям геометрии – узлам, ребрам, зонам, слоям зон и т. п.

Такая модель удобна для описания объектов, имеющих слоистую структуру и не очень сильные различия в геометрии слоев. Однако в ряде случаев применение подобной модели не целесообразно (вследствие значительного увеличения количества слоев сетки, необходимых для адекватного описания особенностей геометрии) или вообще невозможно. Например, в случае показанном на рис. 2.

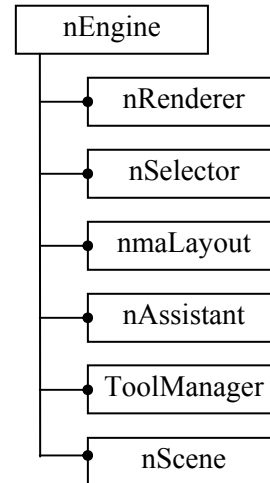


РИС. 1

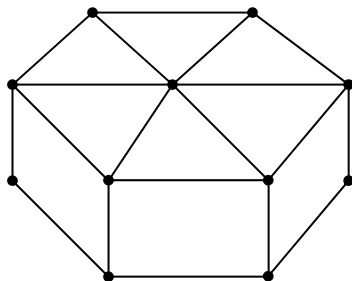


РИС. 2

Поэтому в версии 2.0. ПВиРГИ системы НАДРА-3D реализована более универсальная модель, представляющая моделируемую область в виде непересекающихся зон произвольной формы, заполняющих пространство модели. Это позволило значительно расширить возможности системы в плане описания моделируемых объектов, а также облегчить работу пользователя при создании и редактировании геометрии. Недостатком такого подхода явилось усложнение контроля целостности модели.

Рабочая модель геометрии моделируемой области представлена объектом класса `nScene`. Подсистема ПВиРГИ 2.0 ориентирована на одновременное использование разнотипных геометрических примитивов, представляющих различные фрагменты исследуемой области. Все классы, реализующие примитивы, – потомки класса `nSpatialObject`, в котором объявлены общие для всех примитивов свойства (такие, как идентификатор, имя, цвет, положение в пространстве, флаг выделения и т.п.), а также общие для всех примитивов методы (операции изменения положения, вращения, масштабирования, отображения в окне вывода и т.п.). В основном, эти методы являются чисто виртуальными, поскольку потомки класса `nSpatialObject` могут иметь сложную структуру (например, представлять собой группу примитивов) и заранее неизвестно как именно осуществляется их модификация или отрисовка. Класс `nScene` содержит список указателей на объекты класса `nSpatialObject`, и через предоставляемый ими интерфейс осуществляет базовые операции над всеми примитивами, а также управление выделением объектов.

Подсистема выделения объектов.

При выборе пользователем фрагментов модели, подлежащих редактированию, возможны следующие случаи:

- пользователь выделяет один из примитивов, формирующих геометрию исследуемой области (потомка класса `nSpatialObject`). В этом случае он может либо применить к нему модификаторы перемещения, поворота, масштабирования, либо перейти в режим изменения параметров примитива (при этом системой автоматически определяется и активизируется соответствующий инструмент редактирования);

- в режиме изменения параметров примитива пользователю могут понадобиться функции выбора его составляющих, поскольку представляющие геометрию объекты могут иметь сложную структуру или сами состоять из нескольких разнотипных объектов. Например, при редактировании сетки необходимы такие операции, как выбор группы точек, группы сторон, или группы участков поверхности. В этом случае обработка выделенных элементов осуществляется активным инструментом редактирования.

Поскольку в обоих случаях первым этапом обработки запроса на выделение является определение попадания примитивов в отмеченную пользователем область окна просмотра и ввод параметров модификаторов, в системной компоненте реализованы сервисные классы: `nSelector` и `nAssistant`. Класс `nSelector` содержит методы, осуществляющие тестирование попадания различных графических примитивов в заданную пользователем область окна вывода. Как и `nRenderer`, `nSelector` – абстрактный интерфейсный класс, а конкретные алгоритмы реализуются в его потомках. Класс `nAssistant` представляет собой инструмент, позволяющий пользователю переключаться между модификаторами выделения, перемещения, поворота и масштабирования, осуществлять ввод параметров этих модификаторов в окнах ввода либо с помощью перемещения курсора мыши, управлять ограничением модификации в пространстве вдоль осей координат. При этом сам объект класса `nAssistant` не производит изменений в выделяемых примитивах, а только передает активному инструменту запрос о проведении некоторого действия и необходимые параметры.

Подсистема управления инструментарием.

При разработке подсистемы управления инструментарием программного комплекса НАДРА-3D основное внимание уделялось вопросам.

1. Поскольку система предназначена для моделирования процессов в реальных больших объектах, имеющих сложную структуру, отдельное внимание следует обратить на удобство работы с моделью на макроуровне. Так как исследование поведения моделируемого объекта предполагает проведение расчетов при различных вариантах значений физических параметров или различных вариантах геометрии (возможно, локальных) необходимо обеспечить достаточную гибкость инструментария редактирования и предоставить пользователю возможность оперативного изменения параметров модели.

2. При создании геометрии моделируемого объекта пользователю приходится работать с разнотипной информацией, которая определяет конечный вид геометрии. Например, при проектировании некоторого объекта производится моделирование его влияния на процессы в грунтовой массе. В этом случае пользователю необходимо по результатам геологических исследований построить геометрическую модель грунтового массива, учитывающую конфигурацию залегания пород с различными физическими свойствами, а после внести в нее изменения, отображающие влияние исследуемого объекта. Этот окончательный вариант модели геометрии и является входными данными для дальнейших этапов моделирования.

Как правило, пользователь будет производить моделирование для различных возможных конфигураций проектируемого объекта, а значит, ему понадобится создавать различные модели геометрии. При этом значительная часть исходной информации, описывающая геометрию грунтового массива без влияния объекта, остается неизменной. Подобные особенности должны учитываться при проектировании специализированных инструментов создания и редактирования геометрии исследуемой области.

3. Поскольку моделируемый объект может иметь сложную форму, в процессе моделирования пользователю приходится использовать различные геометрические примитивы и вспомогательные объекты различных типов. С точки зрения их редактирования, можно разделить уровень объекта и уровень подобъекта. Редактирование на уровне объекта предполагает операции перемещения, поворота и масштабирования, которые могут быть применены к любому из объектов созданных пользователем. Редактирование на уровне подобъекта предполагает изменение его специфических параметров. Например, для цилиндра – изменение высоты и радиуса основания, для тела, ограниченного поверхностью из сетки треугольников, – изменение положения ее узлов. При этом инструментарий для редактирования на уровне подобъекта может иметь сложную структуру. Например, позволять переключение между различными уровнями иерархии объекта (точки, ребра, участки поверхности), осуществлять интерактивное (с помощью мыши или клавиатуры) выделение объектов соответствующего уровня иерархии и выполнение доступных действий над ними.

4. Пользователь на любом этапе моделирования в любой момент должен иметь возможность осуществить следующие операции: добавить в модель новый объект желаемого типа, редактировать имеющийся объект, удалить имеющийся объект. Взаимодействие пользователя с представлением моделируемой геометрии и поддержание ее целостности осуществляется только посредством интерфейса программного комплекса.

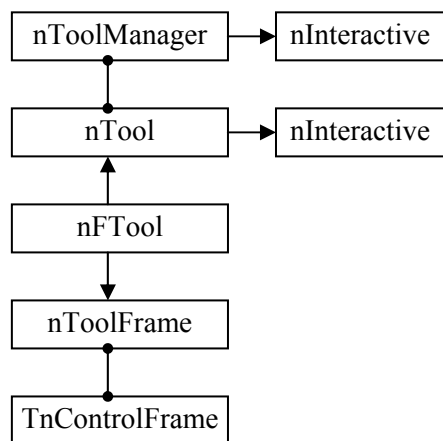


РИС. 3

Иерархия классов, реализующих подсистему управления инструментами программного комплекса НАДРА-3D, показана на рис. 3.

Базовый класс иерархии инструментов – класс nInteractive, представляющий собой интерфейс обработки событий мыши или клавиатуры (перемещение курсора мыши, нажатие клавиш и т.п.). Все методы класса – виртуальные и при необходимости переопределяются его потомками.

Класс nTool описывает интерфейс некоторого абстрактного инструмента. Кроме обработки событий мыши и клавиатуры, унаследованных от nInteractive, nTool определяет виртуальные методы

активации / деактивации, блокирования / разблокирования инструмента, методы отображения инструмента в окнах просмотра, метод обработки сообщений. Кроме того, класс nTool содержит указатель на класс nToolManager, реализующий подсистему управления инструментарием.

Класс `nToolManager` содержит список указателей на доступные в данный момент времени инструменты системы (потомки `nTool`), а также осуществляет централизованное управление операциями блокирования и активации инструментов. При вызове методов, унаследованных `nToolManager` от `nInteractive`, вызываются соответствующие методы активного незаблокированного объекта `nTool`. Для управления активацией каждому инструменту комплекса при проектировании присваивается некий уровень иерархии `index`. При запросе на активацию инструмента с уровнем иерархии `index = i`, если инструмент не заблокирован, `nToolManager` деактивирует активный инструмент, производит блокировку инструментов с уровнями `index = j < i` и активирует запрашиваемый инструмент. При деактивации инструмента с уровнем `i` происходит разблокирование инструментов с уровнями `j < i`. Такая иерархическая организация активации/деактивации инструментов удобна в таких случаях, когда необходимо, например, активизировать инструментарий навигации камеры во время редактирования геометрии. Тогда все события мыши/клавиатуры передаются обработчикам инструментов навигации, а весь остальной инструментарий временно недоступен.

Очевидно, что в большинстве случаев обработки событий мыши/клавиатуры не достаточно для построения инструмента: необходима возможность вводить числовые значения параметров, переключаться между различными режимами инструмента, или работать с редактируемым объектом на различных уровнях его иерархии (точки, ребра, участки поверхности и т.п.). То есть необходимо использовать контекстные (зависящие от выбранного инструмента) панели с окнами текстового ввода и кнопками. Такие инструменты в комплексе НАДРА-3D – потомки класса `nFTool`.

Как видно из рис. 3, класс `nFTool` является потомком класса `nTool`, перенимая таким образом всю его функциональность, и класса `nToolFrame`, обеспечивающего интерфейс работы с контекстными панелями. Это позволяет осуществлять гибкое управление внутри потомков `nFTool` доступными пользователю панелями ввода информации в зависимости от текущего состояния инструмента. Кроме того, класс `nFTool` содержит список указателей типа `nFTool*`, т. е. его потомки могут содержать в себе несколько инструментов, также являющихся потомками `nFTool`. Такая организация необходима для создания инструментов редактирования сложных объектов, содержащих в себе несколько разнотипных примитивов.

Класс `TnControlFrame` предназначен для централизованного управления отображением потомков `nToolFrame` в рабочем окне программы.

Подсистема графического отображения информации.

Одной из важных составляющих продуктивной работы с системой трехмерного моделирования является удобное, управляемое пользователем отображение графической и числовой информации о редактируемой модели.

Для визуализации данных геометрической модели в ПВиРГИ 2.0 программного комплекса НАДРА-3D (как и в ее предшественнике) реализовано несколько видов проекций (двумерные проекции на плоскости XY , XZ , YZ , параллельная и перспективная трехмерные проекции), а также интерактивная система навигации в окнах просмотра (система позиционирования камеры просмотра).

Главное отличие подсистемы визуализации от предыдущей версии состоит в ее разделении на подсистему, реализующую непосредственно графические алгоритмы и осуществляющую отображение графических примитивов (класс `nRenderer`), подсистему управления окнами просмотра, отображения инструментария и геометрии моделируемого объекта (класс `nmaLayout`). Связано это с тем, что предыдущая версия ПВиРГИ предполагала представление геометрии моделируемого объекта в виде трехмерной сетки шестигранных макрозон, подсистема визуализации была оптимизирована для отображения объектов такого типа. В версии 2.0 ПВиРГИ комплекса НАДРА-3D поддерживается создание геометрии произвольного вида, поэтому и понадобился более гибкий инструмент графической визуализации. Иерархия классов подсистемы графического отображения информации показана на рис. 4.

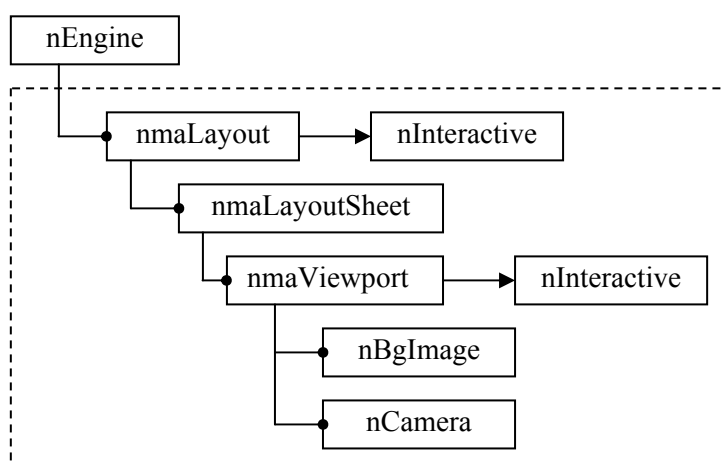


РИС. 4

Класс `nmaLayout` представляет собой реализацию разметки “рабочего стола” пользователя и осуществляет управление доступными окнами просмотра, перехват событий мыши / клавиатуры, отображение моделируемой геометрии и инструментария. Рабочие окна просмотра сгруппированы в настраиваемые пользователем “листы” (объекты класса `nmaLayoutSheet`). Например, для редактирования трехмерных областей удобно использовать порты просмотра в двумерных

проекциях, для просмотра результатов – перспективные, отображающие редактируемую область с определенных точек пространства, для редактирования фоновых изображений – окно XU проекции. Чтобы установить нужную конфигурацию окон просмотра пользователю достаточно активизировать соответствующий “лист” разметки. Количество, размер и положение окон в листах разметки задаются в текстовых файлах настроек и могут редактироваться пользователем, позволяя конфигурировать для каждой конкретной задачи оптимальный вид “рабочего стола”.

Окна просмотра – объекты класса `nmaViewport`, реализующие такие стандартные возможности: выбор типа проекции, управление положением точки взгляда и его направлением, отрисовка моделируемой сцены и используемого инструментария, управление настройками отображения. Класс `nmaViewport` содержит объект класса `nCamera`, осуществляющий управление проецированием и навигацией, и объект класса `nBgImage`, позволяющий использовать в качестве фона рабочего окна оцифрованные изображения.

Кроме того, через указатель на `nEngine`, рабочее окно получает доступ к объекту класса `nRenderer`, системным объектам `nScene` и `nToolManager`, содержащим соответственно все созданные пользователем графические примитивы и все доступные инструменты. При вызове функции отрисовки окна `nmaViewport` производит установку настроек проецирования, рисует фоновое изображение, после чего вызывает функции отрисовки объектов `nScene` и `ToolManager`, которые в свою очередь вызывают функции отрисовки соответствующих графических объектов и инструментов. Отображение потомков класса `nSpatialObject` осуществляется функцией `virtual void Draw(short mode, nRenderer *Renderer)`. Как видно из ее описания – алгоритм отображения каждого примитива в окне просмотра определяется внутри описывающего его класса, используя функции графической подсистемы `nRenderer`.

Подсистема управления фоновыми изображениями. Вышеотмечено, что программный комплекс НАДРА-3D позволяет использовать отсканированные изображения в качестве фона окна просмотра (класс `nBgImage`). При этом предполагается, что данное отсканированное изображение является картой, т. е. известны ее масштаб, разрешение сканирования (`dpi`) и точка привязки к координатной системе моделируемой области. При отображении изображения на основании этих данных и настроек окна просмотра осуществляется соответствующее масштабирование исходного раstra. Отметим также, что поскольку реальные геологические карты имеют большой физический размер, их не всегда возможно (целесообразно) сканировать как одно изображение. Поэтому класс `nBgImage` поддерживает загрузку отдельных фрагментов карты (из файлов формата `*.jpg`) и их последующую сборку в одно изображение.


```

<имя_раздела_1>
[имя_подраздела_1]
имя_параметра_1=значение_параметра_1
имя_параметра_2=значение_параметра_2
имя_параметра_3=значение_параметра_3
...
[имя_подраздела_2]
имя_параметра_1=значение_параметра_1
имя_параметра_2=значение_параметра_2
имя_параметра_3=значение_параметра_3
...
<имя_раздела_2>
[имя_подраздела_1]
имя_параметра_1=значение_параметра_1
имя_параметра_2=значение_параметра_2
имя_параметра_3=значение_параметра_3
...
    
```

РИС. 5

выбор нужного сообщения внутри программы осуществляется по индексу или строковому имени. Это позволяет поддерживать многоязыковый интерфейс – для смены языковых настроек достаточно заменить соответствующие конфигурационные файлы. Их структура показана на рис. 5. Разбор конфигурационного файла осуществляется объектом класса `nIniReader`, который позволяет по имени раздела, подраздела и параметра получить нужное значение.

Заключение. В работе рассмотрены особенности реализации подсистемы ввода и редактирования геометрической информации ПВиРГИ 2.0 программного комплекса НАДРА-3D. Основное внимание уделено идеологии ее построения и архитектуре. Приведены схемы и описание некоторых важных составляющих ПВиРГИ (подсистем хранения и выделения объектов, управления инструментарием, отображения графической информации). Представленные в работе результаты могут быть полезны при разработке систем автоматизированного проектирования.

М.В. Білоус, В.С. Дейнека

ПІДСИСТЕМА ВВОДУ ТА РЕДАГУВАННЯ ГЕОМЕТРИЧНОЇ ІНФОРМАЦІЇ ПРОСТОРОВИХ ШАРУВАТИХ ТІЛ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ НАДРА-3D

Розглядаються особливості та принципи побудови версії 2.0 підсистеми вводу та редагування геометричної інформації автоматизованого комплексу НАДРА-3D.

Подсистема управления текстовыми сообщениями. В процессе работы системы кроме визуализации данных диалог с пользователем предполагает использование различной текстовой информации: сообщений о возникших ошибках, всплывающих подсказок, надписей на кнопках и панелях инструментов, протоколирования выполняемых действий. Подобная текстовая информация в системе НАДРА-3D хранится в соответствующих файлах настроек, а

M.V. Bilous, V.S. Deineka

INPUT-EDITING SUBSYSTEM OF 3D LAYER-LIKE BODIES GEOMETRIC INFORMATION
IN AUTOMATED PROGRAM COMPLEX NADRA-3D

Main features and construction principles of input-editing subsystem (version 2.0) of 3D layer-like bodies geometric information in automated program complex NADRA-3D are considered.

1. *Вещунов В.В., Дейнека В.С.* Подсистема ввода и редактирования информации программного комплекса Nadra 3D // Компьютерная математика. – 2005. – № 1. – С. 25–34.
2. *Вещунов В.В.* Подсистема хранения и оперирования данными программного комплекса Nadra 3D // Компьютерная математика. – 2005. – № 3. – С. 50–58.
3. *Вещунов В.В., Дейнека В.С.* Подсистема визуализации трехмерных объектов программного комплекса Nadra 3D // Компьютерная математика. – 2006. – № 2. – С. 40–50.
4. *Сергиенко И.В., Дейнека В.С., Вещунов В.В.* Информационная технология NADRA 3D исследования процессов многокомпонентных грунтовых сред // Кибернетика и системный анализ. – Киев: Ин-т кибернетики НАН Украины. – 2006. – № 6. – С. 157–174.
5. *Дейнека В.С., Белоус М.В.* О численном моделировании пространственного деформирования среды с сосредоточенной массой и расклинивающим давлением на пересекающихся включениях // Доповіді НАН України. – 2008. – № 10. – С. 35–42.
6. *Дейнека В.С., Белоус М.В.* Розв'язання задачі напружено-деформованого стану багатоконпонентного тіла з великим об'ємом повнозв'язних даних на системі ПЕОМ-СКІТ // Пр. міжнар. конф. "Питання оптимізації обчислень" (ПОО - ХХХІІ), 19 – 23 вересня 2005 р., смт. Кацивелі, Україна, Крим, С. 72.
7. *Френсис Хилл.* Open GL. Программирование компьютерной графики. – С.-П.: Питер, 2002. – 1082 с.

Получено 15.12.2008

Об авторах:

Белоус Максим Владимирович,

аспирант Института кибернетики имени В.М. Глушкова НАН Украины,
e-mail maksbilous@ukr.net

Дейнека Василий Степанович,

доктор физико-математических наук, профессор, академик НАН Украины,
заведующий отделом Института кибернетики имени В.М. Глушкова НАН Украины.
e-mail vdeineka@ukr.net