

СТРУКТУРА ПРОГРАММНЫХ СРЕДСТВ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО МОДЕЛИРОВАНИЯ ИНТЕГРИРОВАННОЙ НАВИГАЦИОННОЙ СИСТЕМЫ

Потапенко В.А.
Кафедра ЭВМ, ДонГТУ
eline@eline.dn.ua

Abstract

Potapenko V. The structure of software for object-oriented simulation of the integrated navigation system. The organization of the integrated navigating system model is overviewed. Purposes and tasks of modeling as well as their embodiment features are described together with the opportunities for the further development.

1. Цели и задачи моделирования

В настоящее время активно ведутся работы по разработке интегрированных навигационных систем (ИНС), призванных поднять на качественно новый уровень функции по управлению подвижными объектами, в том числе речными судами [2,3,7]. Такие системы должны способствовать повышению уровня безопасности движения на водных путях, снижению энергозатрат и экологической нагрузки на природу речных бассейнов [5]. Кроме того, есть потребность в создании модели ИНС для экспериментального обеспечения соответствующих исследовательских и проектных работ, а также – для построения учебных и испытательных тренажеров. Таким образом, вопросы построения интегрированных навигационных средств являются достаточно перспективными, учитывая тот факт, что среди уже разработанных систем не существует ни одной, которую можно было бы успешно применить для речного судоходства Украины. В связи с этим на кафедре ЭВМ ДонГТУ были поставлены задачи по разработке собственной модели ИНС, максимально приближенной по функциональности к уже существующим системам.

Для успешного функционирования реальной ИНС необходимо обеспечить решение ряда взаимосвязанных задач. **Первая** – это определение места расположения судна. При этом может использоваться комплекс программно-аппаратных средств, на вход которого поступает информация от радиолокационной станции (РЛС), спутниковой системы GPS, направленных видеокамер и других источников информации. Выходом такой системы являются координаты положения судна. **Вторая задача** заключается в оценке текущей навигационной ситуации. На основе информации, получаемой от РЛС, необходимо произвести анализ обстановки и принять решение по воздействию на органы управления объектом либо, если система работает в полуавтоматическом режиме, выдать информацию о наиболее приемлемых вариантах дальнейшего движения. Таким образом, определяется **третья задача** ИНС – планирование движения судна, решение которой есть входная информацией для **четвертой задачи** – задачи управления. Управляющие сигналы генерируются для рулей и дросселей двигателей судна. **Дополнительными задачами** являются вопросы построения комплексных систем, в которых информация, включающая в себя электронную карту реки, погодные условия, навигационную обстановку, дополнительную информацию, накапливается на специальном сервере и в режиме реального времени передается на объекты движения (речные суда), которые, в свою очередь, также предоставляют информацию о себе на сервер.

2. Структура програмного забезпечення

В соответствии с поставленными выше задачами, была организована работа по разработке модели интегрированной навигационной системы. В качестве инструмента разработки был выбран принципиально новый объектно-ориентированный язык высокого уровня JAVA, разработанный фирмой SUN в 1995 г. Одним из главных отличий данного языка является переносимость откомпилированных программ на любой тип ЭВМ, поддерживающих виртуальную JAVA-машину. Этот факт дает возможность абстрагироваться от программно-аппаратной платформы, на которой будет работать реальная ИНС. Кроме того, в JAVA широко развиты встроенные сетевые средства, что позволяет решать задачи построения распределенных систем. Упрощенная структура программного обеспечения разработанной модели ИНС с перечислением основных реализованных объектов приведена на рисунке 1.

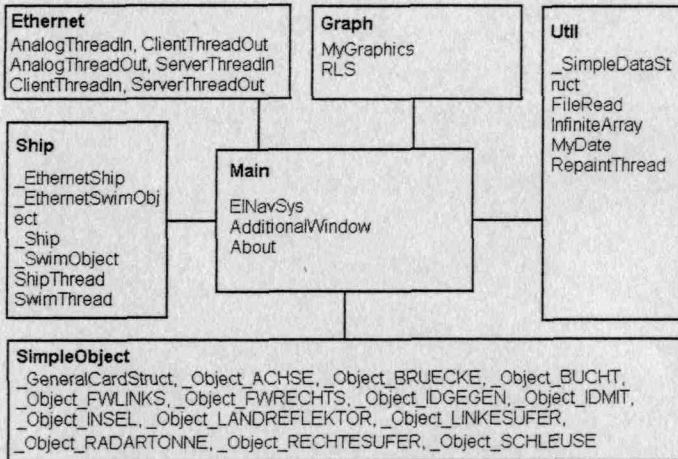


Рисунок 1 – Структура програмного забезпечення

Система состоит из пакетов (Main, Ship, Ethernet, ...), в каждом из которых описаны классы, реализующие соответствующие функции. На приведенном рисунке названия пакетов выделены жирным шрифтом.

Рассмотрим назначение каждого из пакетов подробнее.

Пакет Main. Это главный пакет, в котором осуществляется управление всей системой в целом, а также решаются вопросы взаимодействия между остальными пакетами.

Пакет Ethernet. Реализует сетевой обмен по протоколу TCP/IP между распределенными в сети процессами моделирования. Обмен входными/выходными значениями с пакетом Main осуществляется через векторы (Vectors).

Пакет Graph. В данном пакете реализованы функции по формированию изображения на экране, а также вспомогательные графические функции.

Пакет Util. Включает набор вспомогательных структур и функций, необходимых для работы главного пакета Main.

Пакет SimpleObject. В данном пакете описаны структуры электронной карты реки (Achse, FwLinks, FwRechts и т.д.). В каждом из объектов реализован метод по формированию изображения конкретного элемента карты.

Пакет Ship. В данном пакете определены свойства управляемых объектов, а также реализованы независимые потоки (Threads), представляющие собой участников движения.

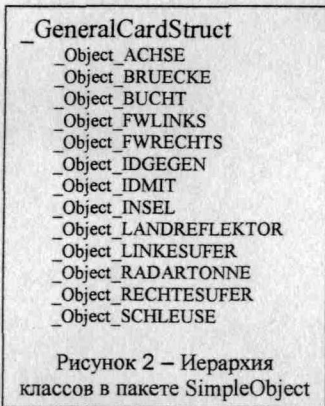
Таким образом, предложена двухуровневая структура ПО, во главе которой стоит пакет **Main**. Данный пакет берет на себя основные функции по управлению работой программы, осуществлению обмена данными, организацию пользовательского интерфейса. Ключевым классом данного пакета является **EINavSys**, который наследует класс **Frame** из пакета **JAVA AWT**. Работа программы начинается с момента создания экземпляра класса **EINavSys**.

3. Применение объектно-ориентированного подхода

В настоящее время практически все средства разработки приложений поддерживают концепцию объектно-ориентированного программирования (ООП). Язык **JAVA**,

используемый при написании модели ИНС, полностью реализован с учетом требований ООП. На примере разработанной системы можно проследить те преимущества, которые предоставляет данный подход в программировании. Для начала рассмотрим пакет **SimpleObject**. Иерархия классов представлена на рисунке 2. В данном пакете содержится базовый класс **_GeneralCardStruct**, в котором описаны общие характеристики для фрагментов, описывающих контур реки. Все переменные данного класса объявлены статическими, так как они являются общими для всех экземпляров данного класса. Класс содержит ряд методов, которые должны быть переопределены в производных классах, важнейшим из которых является метод **draw()**, отвечающий за формирование изображения на экране данного

элемента реки. Далее в пакете содержатся классы, описывающие элементы реки (левый/правый берег, острова, мосты и т.д.). Данные классы наследуют базовый **_GeneralCardStruct** и дополняют его информацией, специфичной для конкретного элемента. Преимущества такого подхода следующие. Во-первых, для каждого элемента описания русла реки можно определить частные свойства и методы по работе с ним. Во-вторых, при добавлении новых элементов потребуется произвести минимум изменений в тексте программы: необходимо всего лишь добавить класс для работы с этим элементом и процедуру для его чтения с диска. При этом подключение произойдет автоматически. Это становится возможным благодаря использованию базового класса, который служит интерфейсом между основной программой и классами элементов реки. В памяти компьютера информация о конфигурации реки хранится в векторе. Вектор – новый тип данных, реализованный в **JAVA**. По своей организации он подобен одномерному массиву, так как доступ к элементам осуществляется по индексу. Однако на этом сходства заканчиваются. Вектор предоставляет более гибкие средства по хранению информации, одним из главных достоинств которого является возможность хранения разнотипных данных. В результате это дало возможность построения следующего массива данных, изображенного на рисунке 3.



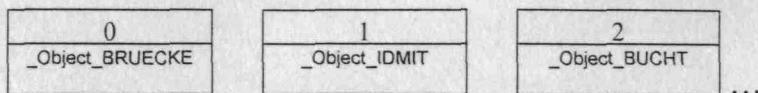


Рисунок 3 – Организация хранения информации в памяти компьютера

В программе осуществляется приведение каждого из элементов вектора к базовому типу **_GeneralCardStruct** и вызов требуемого метода. В результате, если метод в наследуемом классе был переопределен, то вызовется именно он, а не метод базового класса. Следовательно, достигается необходимая гибкость при формировании изображения на экране компьютера, так как заранее конфигурация реки неизвестна.

Далее рассмотрим организацию управляемых объектов. Предполагается, что участниками движения на виртуальной реке будут, как собственно моделируемые объекты, так и другие (встречные и попутные суда, лодки, шлюпки и т.д.). Ввод дополнительных участников движения дает возможность реализовать более реалистичную картину текущей навигационной обстановки. В результате, в пакете **Ship** были реализованы классы **_Ship** и **_SwimObject**. Класс **_Ship** содержит необходимую информацию о моделируемом объекте, такую как текущие координаты, скорость, габаритные размеры и т.д. В ходе работы программы допускается создание неограниченного количества таких классов. При добавлении нового участника движения в конструкторе класса потока осуществляется создание нового объекта и добавление его в список моделируемых. Программа организована таким образом, что моделирование движения может осуществляться на отдельной системе. Данная возможность позволяет упростить текст программы, перенести сложные расчеты на специализированное программное обеспечение, в том числе осуществить распределенные вычисления для повышения производительности работы системы. Для работы с сетевыми объектами реализовано два класса: **_EthernetShip** и **_EthernetSwimObject**.

4. Описание работы программы

Общая структура разработанной программной среды моделирования представлена на рисунке 4. В целом работа программы выглядит следующим образом. При запуске система считывает информацию из файла базы данных (БД) электронной карты реки. Данный файл является текстовым и содержит информацию о координатах составных частей, при помощи которых описывается контур реки. Формат файла детально описан в источнике [1]. Далее происходит создание объектов (экземпляров классов) тех типов, которые встретились во входном файле (**_Object_ACHSE**, **_Object_BRUECKE**, ...). В зависимости от установленных пользователем параметров, на дисплей карта реки может выводиться в различном масштабе и при различной степени детализации. Электронная карта является источником информации для блока управления моделируемыми объектами. На основе этой же информации генерируется и изображение от радиолокационной станции (РЛС) [4, 10]. В интерфейсе пользователя имеется диалоговое окно, в котором реализованы функции по созданию/удалению моделируемых участников движения. При этом движение каждого из участников может осуществляться в одном из двух режимов: автоматическом или ручном. В автоматическом режиме осуществляется движение по идеальной траектории. В ручном режиме пользователь при помощи органов управления осуществляет координату движения самостоятельно. Кроме управления моделируемыми судами, пользователю предоставлены возможности по манипулированию количеством дополнительных плавающих объектов, воссоздающих более реальную навигационную обстановку. Одной из особенностей разработанной системы является то, что она может

взаимодействовать с другими программами, которые могут, например, выполнять функции по решению систем дифференциальных уравнений, описывающих динамику поведения моделируемых объектов. В качестве примера было осуществлено распределенное взаимодействие с программой AnalogComputer, на которой производился расчет угла поворота судна при заданных параметрах скорости движения и угла поворота руля.

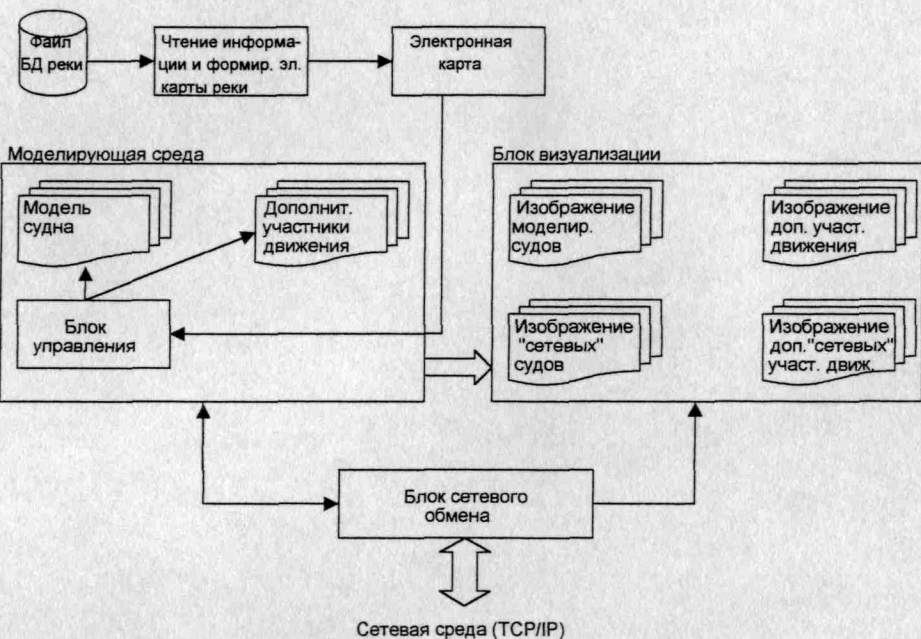


Рисунок 4 – Общая структура среды моделирования

5. Анализ производительности

В общем случае на работу ИНС накладывается важное ограничение: система должна работать в реальном масштабе времени. Однако, если ИНС выступает в качестве консультанта, то возможны варианты, допускающие увеличения периода обработки информации. На разработанной модели ИНС были проведены эксперименты с целью определения быстродействия ее работы при различных параметрах [1]. В результате было установлено, что значительные вычислительные ресурсы отбирает блок визуализации (рис. 4). Данный факт является следствием обработки большого объема информации об электронной карте реки. С подобными проблемами столкнулись и разработчики реальных интегрированных навигационных систем, в том числе в институте системной динамики и автоматического управления ISR Штутгартского университета. В связи с этим программой предусмотрено управление детализацией выводимого изображения. В зависимости от настроек, определенных пользователем, для увеличения производительности на экран может выводиться не вся информация. Кроме того, ресурсоемким является процесс синтеза имитируемого изображения от радиолокационной станции.

6. Заключение

Данная модель является первым шагом на пути построения реально работающих систем для речного судоходства Украины. В ходе работы над системой был получен опыт организации и решения подобного рода задач, определены основные моменты, требующие углубленного анализа и исследований. В качестве основных путей по дальнейшему развитию системы можно выделить следующие. Одной из главных является задача повышения производительности работы. Решением может являться применение более быстродействующих языков программирования при разработке системы, например С++. Перспективной представляется и идея построения специализированного сервера вычислений, с которым будут взаимодействовать объекты управления (клиенты), посылающие на сервер входную информацию (информация от датчиков и измерительных приборов) и получающие назад вектор управления. Необходимо также дополнить систему недостающими блоками обработки информации, важнейшим из которых является блок планирования траектории движения и принятия решений о воздействиях на органы управления.

Литература

1. Потапенко В.А. Розробка та дослідження методів та засобів паралельного моделювання в мережному середовищі / Диссертация на соискание квалификации магистра, ДонДТУ, 1999 г. – 88 с.
2. Gilles E.D., Neul R., Plocher T., and Kabatek U.: Ein integriertes Navigationsystems fur Binnenschiffe. *Automatisierungstechnik* 38 (1990), S.202-209, 247-257.
3. Sandler M., Kabatek U. and Gilles E.D.: Application of an electronic chart in an integrated navigation system for inland ships, in: EURNAV 92(London, 1992) Royal Institute of Navigation and the German Institute of Navigation (DGON).
4. Красюк Н.П. «Вплив тропосфери і підстильної поверхні на роботу РЛС», Москва, «Наука», 1992р., 125с.
5. Навигация. Под ред. Ю.К. Баранова. СПб.: Лань, 1997.
6. Технические средства судовождения. М.: Транспорт, 1990.
7. Sandler M., Wahl A., Zimmermann R., Faul M., Kabatek U. and Gilles E.D. Autonomous guidance of ships on waterways Robotics and Autonomous System 18 (1996), p.327-335.
8. Кривошеев С.В., Потапенко В.А. «Подходы к моделированию работы интегрированных навигационных систем для судов внутреннего и смешанного плавания»/ научные труды Донецкого государственного технического университета. Серия: Информатика, кибернетика и вычислительная техника, (ИКВТ-99) выпуск 6:- Донецк: ДонДТУ, 1999г., стр. 115-120.
9. Аноприенко А.Я., Кривошеев С.В., Потапенко В.А. Моделирование процесса обработки информации в интегрированной навигационной системе // Тези доповідей міждержавної науково-методичної конференції "Комп'ютерне моделювання 30 червня - 2 липня 1999 р., м. Дніпродзержинськ. - Дніпродзержинськ. - 1999. - С. 114-115.
10. Святный В.А., Аноприенко А.Я., Кривошеев С.В., Потапенко В.А. Имитация радиолокационной информации в интегрированной навигационной системе // Сборник трудов VI международной конференции в г. Севастополе 13-18 сентября 1999 г. "Машиностроение и техносфера на рубеже XXI века". Том 3. - Донецк: ДонДТУ. - 1999. - С. 12-15.