

ПРОГРАММНАЯ СРЕДА РАЗРАБОТКИ ПРИЛОЖЕНИЙ ВИРТУАЛЬНОГО ОКРУЖЕНИЯ АВАНГО

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Житко А. П., Шафар Н. А.

Хорошко В.В. – канд.техн.наук, доцент

Аванго [1] – это среда программирования для создания распределённых интерактивных приложений в виртуальном окружении. Она использует язык программирования С для определения двух категорий объектных классов. Узлы обеспечивают объектно-ориентированный интерфейс графов сцены, который допускает представление и обработку сложной геометрии. Датчики обеспечивают Аванго связь с реальным миром и используются для ввода данных с внешних устройств в приложение.

Все объекты Аванго – полевые контейнеры, представляющие информацию о состоянии объекта как набор полей. Они поддерживают универсальный потоковый интерфейс, который позволяет объектам и информации о их состоянии быть записанным в поток, и последующую реконструкцию объекта из этого потока. Такой интерфейс один из основных строительных блоков, используемых для реализации распределения объектов.

Аванго использует связи между полями для формирования графа потоков данных, ортогонального к графу сцены, который используется, чтобы задать поведение, и учитывает интерактивные приложения.

В качестве основы для Аванго выбран SGI Performer [2] – программный интерфейс для создания интерактивных графических приложений, обеспечивающий их максимальную производительность. Расширенные графические возможности, такие как отбор видимых поверхностей, переключение между уровнями детализации и взаимодействие с графическим оборудованием, все выполняется Performer. Performer может использовать несколько процессоров и графических конвейеров, если это допускается оборудованием.

Помимо С интерфейса, Аванго использует интерпретируемый язык Scheme (Схема) [3]. Scheme – универсальный язык программирования, произошедший от Алгола и Лиспа. Это язык высокого уровня, поддерживающий операции со структурными данными типа строк, списков и векторов. Все объекты высокого уровня в Аванго могут быть созданы и управляемы посредством Scheme.

Интерфейс сценариев Аванго предлагает двухуровневый подход к разработке приложений. Сложные и критичные для исполнения возможности реализуются в С с помощью создания подклассов, расширяющих классы Аванго. Само приложение в этом случае – только набор Scheme сценариев, которые создают требуемые объекты Аванго, вызывают их методы, устанавливают значения их полей и определяют отношения между ними. Сценарии могут быть написаны, проверены и отлажены прямо во время работы приложения. Это значительно уменьшает время разработки и обеспечивает разработчиков мощной средой создания прототипа приложения.

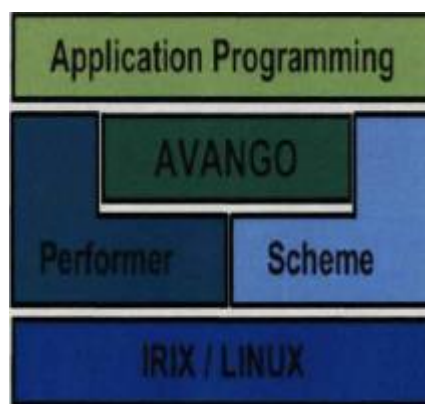


Рисунок. 1 – Связь Аванго с другими системами

Эффективная реализация универсального потокового интерфейса для разнородных объектов требует дополнительной мета-информации об атрибутах объектов и их типах, а также о способе обращения к этим атрибутам без точного знания типа содержащегося объекта. Язык

программирования C не рассматривает классы как объекты, так что эта мета-информация труднодоступна на языковом уровне. Performer, например, использует функцию-член интерфейса, чтобы обратиться к атрибутам состояния объекта. Симметричная пара функций установки и взятия значения существует для каждого атрибута. Установка одного атрибута может, как побочный эффект, изменить другой атрибут этого объекта. Однако, никакая абстрактная информация о числе атрибутов, их типе и их значениях не может быть получена от объекта через интерфейс Performer. Чтобы устранить этот недостаток, в Аванго использован подход, принятый системой Inventor и который также можно найти в спецификации VRML [4]. Аванго использует поля как контейнеры для атрибутов состояния объекта. Поля содержат основные типы данных и предоставляют универсальный потоковый интерфейс. Они реализованы как общедоступные члены класса и таким образом унаследованы производными классами. Они непосредственно доступны клиентским классам и являются главным интерфейсом Аванго к атрибутам состояния объекта.

Поля бывают двух типов. Однозначные поля содержат одно значение данных основного типа, в то время как мультиполя содержат вектор произвольного числа значений. Аванго интенсивно использует Стандартную Библиотеку Шаблонов (STL). Мультиполевые значения реализованы как STL векторы полевого типа. Поля реализованы посредством C шаблонов, которые допускают параметризацию avField() класса требуемым типом данных. Как пример полевого интерфейса, часть определения шаблона класса для однозначного поля приведена здесь:

```
template<class T>
class avSingleField : public avField {
public:
void setValue(const T& value);
const T& getValue() const;
};
template<class T>
ostream& operator<<(ostream& stream,
const avSingleField<T>& field);
template<class T>
istream& operator>>(istream& stream,
avSingleField<T>& field);
```

Доступ к полемому значению обеспечивается методами getValue() и setValue(). Для каждого полевого класса существует пара потоковых операторов, позволяющих преобразовать полевое значение в последовательную потоковую форму и восстановить полевое значение из потока.

Аванго объекты являются полевыми контейнерами, которые представляют состояния объекта как набор полей. Полевой контейнер может быть опрошен о числе содержащихся в нем полей и об их ссылках. Соответствующие части интерфейса полевых контейнеров имеют вид:

Это, вместе с универсальным потоковым интерфейсом полей, позволяет Аванго обеспечивать потоковые функциональные возможности на уровне полевых контейнеров без того, чтобы знать точный тип основного объекта. Это распространяется на все классы, выведенные из avFieldContain.

Шаблонный класс avLink<> определяет эффективный указатель на объекты Аванго, который скрывает детали подсчёта ссылок от разработчика и существенно уменьшает сложность написания свободного от исключений кода.

Интерфейс полевых контейнеров Аванго обеспечивает прочную основу для последовательной универсальной разработки таких передовых черт, как организация потоков данных, создание сценариев и распределение. Эти возможности унаследованы любыми специфичными для приложения расширениями, добавленными разработчиком. Поскольку узлы Аванго являются подклассификацией узлов Performer, узлы обоих типов могут быть свободно объединены при построении графа сцены.

Список использованных источников:

1. Tramberend H., Avocado: A Distributed Virtual Reality Framework /7 Proc. of the IEEE Virtual Reality. - 1999. IEEE
2. Rohlf J., Helman J. IRIS Performer: A High Performance Multiprocessing Toolkit for Real Time 3D Graphic. // In: A. Glassner (ed): Proc. of the 21st annual conference on Computer graphics and interactive techniques — 1994. ACM. - pp.381-394.
3. Dybvig R. K., The Scheme programming language: ANSI Scheme. — P T R Prentice-Hall, Englewood Cliffs, NJ 07632, USA, second edition, 1996.272 p.
4. R. Carey and G. Bell. The VRML 2.0 Annotated Reference Manual. -Addison-Wesley, Reading, MA, USA, 1997. — 504 p.