

Рис. 1. Узагальнена структура тренажера

Основним елементом цього тренажера є автоматизоване робоче місце оператора АС УПР, на якому відтворюються умови імітованого процесу. АРМО АС УПР являє собою виконаний в натуральну величину макет робочої зони реального об'єкта або сам реальний об'єкт.

Тренажер для підготовки операторів АС УПР повинен дозволяти інструкторові змінювати ситуації, вводити нові або додаткові умови, що ускладнюють управління об'єктом або створюють перешкоди. Тому подальші дослідження необхідно направити на розробку автоматизованої системи оцінки операторської діяльності, до якої включені засоби реєстрації, аналізу, оцінки дій оператора, модифікацію умов проведення тренувань і їх застосування на тренажерах

Список використаних джерел:

1. Шукшунов В. Е., Циблев В. В., Потоцкий С. И. Тренажерные комплексы и тренажеры : технологии разраб. и опыт эксплуатации. – М. : Машиностроение, 2005. – 383 с.
2. Красовский А. А. Основы теории авиационных тренажеров. М.: Машиностроение, 1995. – 304 с.
3. Шукшунов В. Е., Бакулов Ю. А., Григоренко В. Н. Тренажерные системы. – М.: Машиностроение, 1981. – 256 с.

Ходарина К.В.

*кандидат технических наук, доцент,
Азовский морской институт*

Национального университета «Одесская морская академия»

ПРОЕКТ ВИРТУАЛЬНОЙ СРЕДЫ ДЛЯ ИНТЕРАКТИВНЫХ СИСТЕМ В МОРСКОЙ ОТРАСЛИ

Первые попытки использования компьютерной техники для организации технического обслуживания судов были предприняты в начале 1980-х годов, когда в Норвегии была разработана первая версия информационной системы

AMOS-D, которую с 1985 г. развивала и внедряла компания SpecTec AS. Фирма STN Systemtechnik (Германия) интегрировала систему для управления технологическими процессами на судах с отдельными элементами административного управления. Фирма SpecTec AS предлагает программный продукт AMOS Business Suite (деловой комплект) и функционирует как холдинг, имеющий 20 офисов в разных странах. В то же время большинство поставщиков стали предлагать пользователям все возможные варианты инсталляции и использования компьютеризированных систем [1].

Однако все имеющиеся средства ориентированы на использование программистами. Часто для разработки виртуальных сред требуется использовать несколько различных библиотек и инструментальных средств.

Современный подход в области информационных технологий требует минимизации участия программистов в разработке прикладных программных систем; осуществление доступа через Интернет, как к средствам разработки программных систем, так и средствам их использования.

Целью данной работы является описание методики интерпретации виртуальной интерактивной среды для использования в морской отрасли.

Реализация указанных выше требований может быть найдена в рамках процедурно-декларативного подхода к автоматизации морской профессиональной деятельности. В соответствии с ключевыми идеями данного подхода предлагаются следующие основные принципы проектирования системы [2]:

1. Разработка процедурно-декларативной модели метаинформации для формирования проекта виртуальной интерактивной среды.
2. Разработка процедурно-декларативного проекта виртуальной интерактивной среды в терминах метаинформации.
3. Обеспечение функционирования виртуальной интерактивной среды через интерпретацию его процедурно-декларативного проекта и генерацию по нему виртуальной системы.
4. Селектор проекта и генератор виртуальной системы необходимы для уменьшения трудоемкости разработки и сопровождения виртуальной интерактивной среды.
5. Реализация и использование всего программного комплекса как облачного сервиса. Использование технологии облачных вычислений дает новый уровень гибкости для управления и сопровождения программных средств, а также значительно расширяет возможности пользователей [3].

Общая архитектура системы состоит из информационных и программных компонентов.

Информационными компонентами системы являются: процедурно-декларативные модели метаинформации, процедурно-декларативные проекты, различные ресурсы (3d-модели, текстуры и др.), внешние функции.

Программными компонентами системы являются: редактор проекта; редактор виртуальной системы; редактор внешних функций; селектор.

На основе представленных информационных составляющих с помощью приведенных редакторов строится процедурно-декларативный проект виртуальной среды, который интерпретируется в трехмерную виртуальную среду, отображаемую пользователю в браузере. Процесс интерпретации процедурно-

декларативного описания проекта в работающую программную систему выполняет селектор.

Интерпретация процедурно-декларативной модели проекта заключается в создании на основе этой модели программной виртуальной среды, обеспечении интерактивного взаимодействия с пользователем в соответствии с заданными в модели интерактивными возможностями (рис. 1).

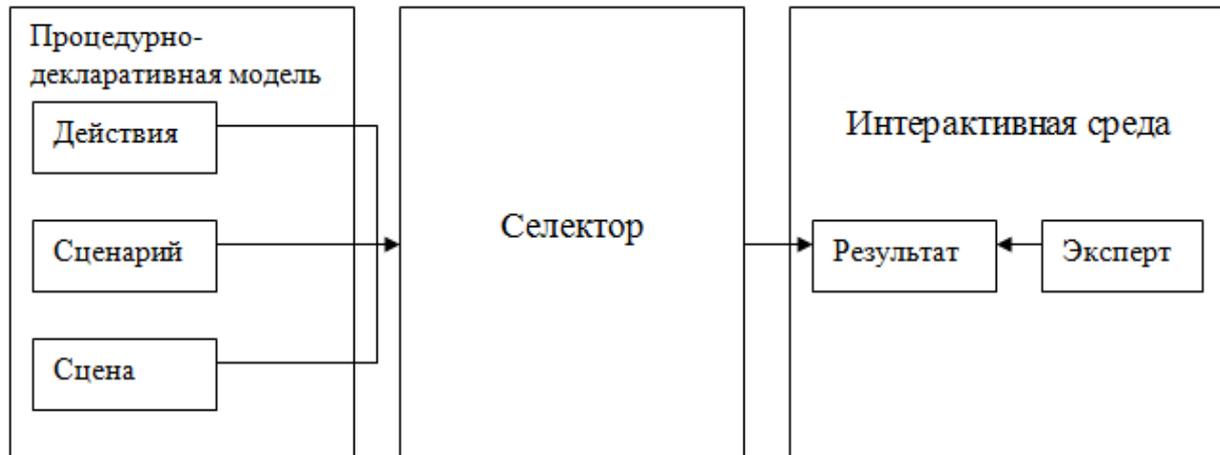


Рис. 1. Интерпретация процедурно-декларативной модели

Селектор выполняет следующие задачи:

1. Инициализация виртуальной сцены по ее проекту.
2. Инициализация интерактивного взаимодействия пользователя с созданной виртуальной сценой на основе описания действий, представленных в проекте.
3. Динамическая обработка событий виртуальной среды (действий пользователя) с применением и изменением процедурно-декларативного проекта. Изменение виртуальной среды.
4. Обработка и оценка (интерпретация) выполненных действий в соответствии с описанием сценария (если сценарий задан в проекте).

На рис. 2 представлена упрощенная архитектура селектора

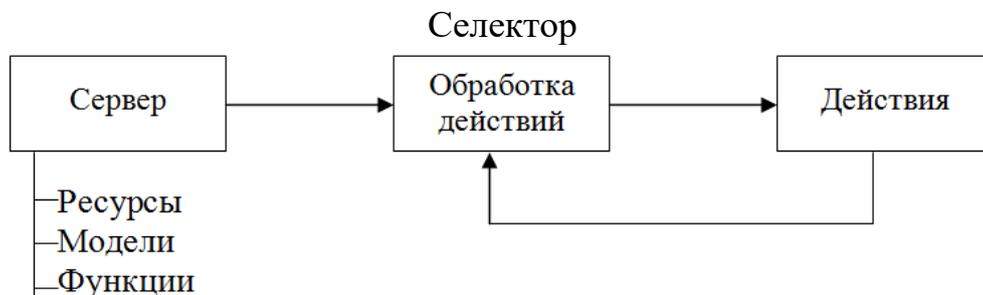


Рис. 2. Архитектура селектора

Архитектура селектора состоит из двух основных частей: клиентской и серверной.

На судовом компьютере выполняется задача генерации изображения трехмерной сцены виртуальной интерактивной среды, осуществление взаимодействия между пользователем и средой, отображение интерфейса. Ставятся

задачи, независимые от внешних обработок, т. е. такие, которые могут быть выполнены на локальном компьютере пользователя.

На сервере происходит вся логическая и дополнительная внешняя обработка действий, и затем эта информация пересылается для отображения в браузер пользователя. Серверная часть не зависит от параметров локального компьютера пользователя и структуры клиента [2].

Командные действия на судовом компьютере интерпретируются в соответствующие элементы интерфейса с предоставлением выбора нужных команд для пользователя. Интерактивные действия на этапе инициализации и последующей интерпретации динамически связываются с необходимыми объектами в соответствии с их текущими состояниями.

В процессе работы интерактивного приложения селектор отслеживает и изменяет свойства и связи объектов, событий и действий в зависимости от момента времени и состояний объектов. Селектор в процессе работы программы устанавливает новые и убирает ненужные события, связанные с выполнением соответствующих действий.

В результате выполнения действия для его обработки могут быть получены и потребованы различные параметры, например, координаты объекта. По процедурно-декларативному проекту селектор определяет требуемые параметры для обработки действия, получает их из имеющейся информации в виртуальной среде и затем передает для обработки на сервер. На сервере реализуется либо стандартная обработка параметров действий, либо специфическая, которая определена в проекте. Для изменения состояний объектов селектор должен пройти по всем описанным в модели атрибутам нового состояния объекта и выполнить:

1. Изменение визуального отображения объекта сцены (связанное с такими презентационными атрибутами объекта, как: трехмерная модель объекта, текстура, координаты, повороты, коэффициенты масштабирования) на клиентском компьютере.

2. Изменить логические и презентационные атрибуты объектов в процедурно-декларативном проекте в состоянии и на сервере.

Обработка действий селектором может зависеть от множества различных параметров, включающих атрибуты объектов, параметры сцены в целом. Алгоритм получения результата может быть реализован во внешнем модуле. Поэтому основная обработка информационных результатов находится на серверной части, на которую с клиента отправляется соответствующее сообщение. На судовом компьютере выделяются следующие подзадачи:

1. Сформировать сообщение серверу о необходимости выполнения некоторого алгоритма для получения результата.

2. Получить ответное сообщение от сервера с результатом выполнения.

3. Отобразить пользователю результат обработки [4].

Задача селектора – найти результат, соответствующий всем полученным параметрам, отобразить его пользователю в виртуальной среде, получить оценку этого результата, выполнить объяснение этого результата в соответствии со сценарием, заданным в процедурно-декларативной модели.

Данный программный комплекс можно использовать при создании интерактивных виртуальных сред для решения различных задач в морской отрасли.

Список использованных источников:

1. Горб С.И. Тенденции развития технического обслуживания судов // Автоматизация судовых технических средств. Вып. 12. – 2007. – С. 26-34.
2. Грибова В.В., Федорищев Л.А. «Виртуальная реальность в образовании: система разработки интернет-проектов» // НИТО, 2012. – С. 116-118.
3. Соснин П.И. Создание и использование онтологий проектов в разработке автоматизированных систем // КИИ 2010, Тверь. – 2010. Том 2. – С. 187-195.
4. Гаркуша Г.Г. Концепция внедрения е-навигации в практику судовождения // Всеукраїнська конференція «Сучасні методи і моделі розвитку морського транспорту та безпека судноплавства» 26 травня 2017 р. АМІ НУ «ОМА» – С. 84-90.

Шокотько Д.Д.

студентка,

Одеський державний університет

імені І.І. Мечникова

ВИКОРИСТАННЯ ІНДЕКСАТОРІВ ПРИ РЕАЛІЗАЦІЇ АЛГОРИТМІВ ЧИСЕЛЬНИХ МЕТОДІВ ЛІНІЙНОЇ АЛГЕБРИ

Сучасна мова програмування C# та середовище розробки Windows-додатків Microsoft Visual Studio є однією з найпотужніших, процвітаючих та затребуваних систем в IT-галузі [1; 2].

В цьому середовищі розробляються найрізноманітніші програми: від невеликих студентських додатків до масштабних Web-порталів та Web-сервісів.

Але у мови програмування C# також є і специфічні незручності, які вона отримала «у спадок» від своїх іменитих попередників C та C++.

Ми зупинимося на одній з них. Проблема полягає в індексації елементів масивів різної форми. У мові програмування C# вона починається з нуля.

Однак, для позначень, які прийняті в класичній математиці матричних обчислень, де нумерація рядків і стовпців починається з одиниці, це породжує деякі незручності, і саме через це у програмістів часто виникають прикрі помилки.

Цю проблему для задач алгебри матриць, на наш погляд, можна частково вирішити за допомогою індексаторів.

За правилом, в C# індексація масивів різної розмірності реалізується з використанням оператора [].

Ми можемо створити свій тип даних – клас `Class_Matrix`, в якому виконаємо перевантаження цього оператора, не вдаючись до «послуг» методу `operator()`, а за допомогою створення індексатора (`indexer`).