

УДК 004.921
DOI: 10.15827/0236-235X.123.439-443

Дата подачи статьи: 25.06.18
2018. Т. 31. № 3. С. 439–443

Высокоуровневая архитектура тренажерно-обучающих систем сложных технических комплексов

*А.В. Родителев*¹, ведущий программист, *avrod_94@mail.ru*
*А.М. Гиацинтов*¹, старший научный сотрудник, *alghts@inbox.ru*

¹ *Центр визуализации и спутниковых информационных технологий ФНЦ НИИСИ РАН, отдел разработки имитационных и спутниковых технологий, г. Москва, 117218, Россия*

В статье рассмотрена архитектура тренажерно-обучающей системы на примере прототипа авиационного тренажера. Подсистема визуализации тренажерно-обучающей системы обеспечивает отображение результатов моделирования внешней среды и объекта управления с помощью устройств отображения информации. Она должна обеспечивать воспроизведение созданной виртуальной сцены с достаточно подробным содержанием, позволяющим операторам успешно выполнять поставленные задачи.

Приведены требования к тренажерно-обучающей системе сложных технических комплексов, в том числе предъявляемые к ее подсистеме визуализации. Разработанная архитектура обеспечивает унификацию способов сопряжения с оборудованием, использование тренажерно-обучающей системы с различными устройствами ввода информации, а также позволяет избежать высокой связанности компонентов. Каждое устройство, как правило, имеет свои особенности: специализированное ПО, закрытые протоколы обмена информацией, разъемы различных стандартов.

Разработанная система расширений (плагинов) позволяет учитывать особенности устройств без модификации основного модуля и других подсистем. Для работы с плагинами был создан управляющий интерфейс. Плагины не зависят друг от друга и могут быть добавлены или удалены без нарушения целостности работы системы.

В зависимости от вычислительной нагрузки обработку данных тренажерно-обучающей системы можно реализовать на одном аппаратном комплексе или для каждой подсистемы использовать отдельный. Любая подсистема может быть создана независимыми разработчиками и представлена отдельными программными комплексами.

Предусмотрена возможность работы проектируемых подсистем на аппаратных комплексах с различными архитектурами процессоров, способами хранения байтов в памяти (big-endian или little-endian) и на различных операционных системах. Описан алгоритм преобразования координат, полученных от математической модели, в систему координат подсистемы визуализации.

Ключевые слова: ТОС, тренажерно-обучающие системы, система визуализации, трехмерное моделирование, тренажер, имитационные системы.

Под *тренажерно-обучающей системой* (ТОС) оператора *сложной технической системы* (СТС) будем понимать техническое средство для подготовки операторов в едином информационном окружении, отвечающее требованиям методик подготовки, создающее условия для получения знаний, навыков и умений, реализующее модель таких систем и обеспечивающее контроль над действиями обучаемого, а также для исследований. Кроме формирования индивидуальных профессиональных навыков и умений, ТОС могут применяться для отработки групповых операций.

В зависимости от объема решаемых задач ТОС можно разделить на комплексные, специализированные, процедурные. Комплексные тренажеры предназначены для решения задач практического обучения и подготовки операторов, в полном объеме охватывающих их профессиональную деятельность, и должны обеспечивать отработку и формирование всех профессиональных навыков. Специализированные тренажеры рассчитаны на подготовку операторов к выполнению определенных проблемно-ориентированных задач. Процедурные тренажеры предназначены для выработки устойчивых навыков в конкретных видах деятельности, отработки отдельных важных процедур управления определенным видом оборудования,

выполняемых на конкретных этапах эксплуатации и в определенных условиях [1].

В данной статье рассмотрена система на примере прототипа авиационной ТОС, к которой применяются следующие требования:

- имитация процедур в реальном масштабе времени (латентность тренажера не более 20 мс, для реализации необходимо выполнять вычисления в течение 15 мс, 5 мс запас);
- отсутствие разрывов изображения при многоканальной визуализации;
- исключение привития ложных навыков;
- модульная архитектура построения, позволяющая без глубокой переработки системы удалять, заменять и создавать модули.

Применение модульной архитектуры

Использование модульной архитектуры построения позволит избежать высокой связанности компонентов, которая может наложить ряд ограничений на функционирование ТОС (например, под каждую новую версию модуля от разных разработчиков необходимо для совместимости изменять функционал всей системы). Была разработана система расширений (плагинов), взаимодействующих с основным модулем. Плагины не зависят друг

от друга и могут быть добавлены или удалены без нарушения целостности работы системы [2].

Для работы с внешними плагинами был создан управляющий интерфейс. При загрузке системы происходит поиск плагинов по расширению в указанной директории, после чего производится инициализация плагина. Каждый плагин работает в отдельном потоке, так как использование одного потока для всех плагинов может привести к зависанию системы моделирования, если какой-то плагин будет ожидать выполнения определенного условия. Для функционирования загруженного плагина в системе необходимо наличие реализации функции плагина, в которой описываются основные этапы работы плагина. Например, в данной функции могут определяться заранее заданные в конфигурационном файле входящие и исходящие порты и IP-адреса, осуществляться подготовка данных для отправки в соответствующую подсистему и обработка обновленных данных, приходящих из подсистемы с последующей записью в хранилище, а также задаваться частота отправки пакетов.

Если инициализация прошла успешно, плагин добавляется в список загруженных. При неуспешной инициализации он выгружается из памяти. При запросе остановки основного модуля осуществляется выгрузка всех плагинов.

Описание подсистем, входящих в состав ТОС

В зависимости от вычислительной нагрузки обработку данных ТОС возможно реализовать на одном аппаратном комплексе или для каждой подсистемы использовать отдельный [3]. Любая подсистема может быть создана независимыми разработчиками и представлена отдельными программными комплексами. Необходимо учитывать, что разрабатываемые подсистемы могут работать на аппаратных комплексах с различными архитектурами процессоров, способами хранения байтов в памяти (big-endian или little-endian) и на различных операционных системах. Например, при отправке пакета данных к подсистеме, которая использует иной порядок байт, необходимо производить перестановку байт в плагине после получения пакета от подсистемы и перед отправкой данных. Также следует учесть, что для структур, в которых содержатся вложенные структуры, необходимо производить перестановку порядка байт отдельно для каждого значения. Обмен данными между подсистемой и плагином может осуществляться по технологии Ethernet с использованием протокола UDP либо посредством подключения библиотеки к плагину.

Для указания расположения подсистем в рассматриваемом прототипе ТОС используется конфигурационный файл, в котором указываются IP-адреса, входящие и исходящие порты, MAC-адрес

сетевой карты аппаратного комплекса основного модуля. Использование MAC-адреса обусловлено возможным наличием нескольких сетевых карт на рабочей станции.

Для подсистем возможен следующий подход к работе с сетью. При получении пакета подсистемой определяется входящий IP-адрес. После расчета новых параметров сформированный пакет будет отправлен по данному IP-адресу на указанный исходящий порт. Входящие и исходящие порты могут задаваться как в конфигурационном файле для конкретной подсистемы, так и в виде входных аргументов приложения. В случае отсутствия заданного IP-адреса или описания входящих или исходящих портов будут использоваться значения, заданные по умолчанию (localhost).

В основной модуль поступают параметры из всех подсистем, значения которых хранятся в едином хранилище (структуре) [4]. Значения параметров могут иметь различный тип, доступ возможен из любого плагина. В данной структуре могут храниться числа, строки, массивы и бинарные данные. Хранение данных реализовано в виде набора пар ключ–значение. Если данные из структуры необходимо извлечь и отправить в нужную подсистему, их преобразуют с помощью соответствующих функций из формата, используемого в этой структуре, в формат, требуемый для нижележащих компонентов.

В ТОС входят подсистемы

- рабочего места инструктора;
- формирования управляющих воздействий;
- математического моделирования объекта, в том числе БД описаний объекта моделирования;
- моделирования электронных и механических средств отображения состояния объекта;
- математического моделирования окружающей обстановки, в том числе физического моделирования, моделирования подстилающей поверхности и БД объектов реального окружения;
- визуализации;
- навигации.

Использование такой архитектуры также может обеспечить унификацию способов сопряжения с оборудованием и позволит использовать ТОС с различными устройствами ввода информации. Каждое аппаратное средство, как правило, имеет свои особенности: специализированное ПО, закрытые протоколы обмена информацией, разъемы различных стандартов [5]. Разработанная система плагинов позволяет учитывать особенности аппаратных средств без модификации основного модуля и других подсистем.

Во время инициализации подсистемы формирования управляющих воздействий производится определение всех подключенных устройств. Этап инициализации состоит в получении индексов устройств в системе, их имен и записи полученных данных в общий массив. Значения параметров

штурвалов и рычагов управления представляют собой углы отклонения от соответствующих начальных значений осей.

Подсистема математического моделирования объекта в том числе выполняет следующие функции:

- обработка введенных инструктором начальных условий;
- расчет системы автоматического управления (при наличии);
- расчет параметров объекта с учетом управляющих воздействий, в том числе определение пространственного положения объекта в виртуальном окружении и расчет параметров для приборов и рычагов управления.

Для ряда систем поведение в нештатных ситуациях недостаточно изучено. Это может быть связано с тем, что существует недостаток экспериментальных данных или же получение таких данных очень затратно или вовсе опасно.

Подсистема рабочего места инструктора – инструмент для контроля и изменения всех параметров моделирующей системы [6]. С ее помощью можно изменять параметры работы всех систем, входящих в состав ТОС. Инструктор на протяжении всей тренировки может задавать нештатные ситуации, в том числе отказы различных систем и оборудования, задавать параметры окружения, выполнять аварийное отключение тренажера. Система инструкторского контроля обеспечивает сбор, обработку, документирование, анализ и выдачу статистических данных.

Требования, предъявляемые к системе визуализации ТОС

Рассмотрим требования, которые предъявляются к системе визуализации ТОС. Информация, поступающая путем визуального наблюдения, воспринимается различно в зависимости от удаленности наблюдаемых объектов. Существует разделение на три зоны: до 1 метра, от 1 до 30 метров и свыше 30 метров от наблюдателя. В зоне более 30 метров, когда бинокулярное зрение ослабевает (оба глаза получают одинаковое изображение), окклюзия и размеры приобретают важность для адекватного восприятия. Если экран размещен на расстоянии порядка 3 метров от обучаемого, а объект по сценарию тренировки удален от наблюдателя менее чем на 3 метра, при использовании проекционных систем отображения информации может происходить рассогласование.

В тренажере передача зрительных сигналов о перемещении основана на том, что мозг человека с трудом различает изменения ситуаций (используя только зрительный канал), когда наблюдатель неподвижен, а окружающие объекты находятся в движении и когда движется наблюдатель, а объекты неподвижны. Поэтому генерируется движу-

щаяся окружающая обстановка, создавая иллюзию перемещения наблюдателя (подвижная платформа способствует усилению восприятия движения) [7].

Детализация воспроизведения окружающей обстановки напрямую связана с производительностью имитационной системы. Для повышения эффективности использования ресурсов тренажера необходимо определить наиболее важные для достижения целей тренировки группы объектов и, следовательно, детально воспроизводить именно их.

Для воссоздания окружающей обстановки используются реальные снимки подстилающей поверхности. Формирование подстилающей поверхности может осуществляться в системе для создания и редактирования трехмерной графики или при помощи процедурной генерации [8]. При воссоздании взлетно-посадочной полосы необходимо отобразить места касания шасси воздушного судна при посадке. После создания трехмерной сцены выполняется ее сохранение в промежуточный формат, например COLLADA, который позже преобразуется конвертером графического движка во внутренний формат трехмерных сцен.

Метод преобразования координат математической модели ТОС в систему координат подсистемы визуализации

В подсистему визуализации через плагин из основного модуля передаются значения идентификации пакета, долготы, широты и высоты (WGS 84), углов (в градусах) крена, рысканья, тангажа. Эти значения используются для изменения позиции и углов наклона виртуальной камеры в трехмерной сцене. Так как подсистема визуализации имеет свою систему координат, необходимо преобразовать полученные значения в систему координат подсистемы визуализации. Расчет расстояния между двумя точками на поверхности Земли происходит по формуле

$$d = R\sqrt{\Delta\lambda^2 \cos(\lambda_1)\cos(\lambda_2)\Delta\varphi^2},$$

$$\Delta\lambda = \lambda_2 - \lambda_1, \Delta\varphi = \varphi_2 - \varphi_1, \quad (1)$$

где $\lambda_2 - \lambda_1$ – широты двух точек; $\varphi_2 - \varphi_1$ – долгота двух точек; R – радиус Земли; $\Delta\lambda, \Delta\varphi$ – расстояние между точками широты и долготы.

Далее необходимо рассчитать значение азимута, который представляет собой угол между плоскостью меридиана точки наблюдения и вертикальной плоскостью, проходящей через эту точку и наблюдаемый объект [9]. Значение азимута необходимо для расчета итогового положения воздушного судна:

$$\psi = \tan^{-1}\left(\frac{\cos(\lambda_2)\sin(\Delta\varphi)\cos(\lambda_1)}{\sin(\lambda_2) - \sin(\lambda_1)\cos(\Delta\varphi)}\right). \quad (2)$$

Абсолютные значения числителя и знаменателя формулы (2), используемые для вычисления отно-

сительного азимута, ограничены диапазоном $\pm \pi/2$. Поэтому, если значение знаменателя формулы (2) меньше нуля, то $\psi = \pi - \psi$, если же значение числителя формулы (2) меньше нуля, то $\psi = -\psi$. После этого вычисляется итоговое значение направления магнитного поля в пределах ВПП в радианах:

$$q = RQDM + MagVar, \quad (3)$$

где $RQDM$ – направление магнитного поля; $MagVar$ – магнитная вариация.

Магнитная вариация представляет собой различные изменения во времени геомагнитного поля, обусловленные существованием как внутренних, так и внешних по отношению к поверхности Земли источников магнитного поля [6].

Далее необходимо численное выражение направления магнитного поля преобразовать из градусов относительно магнитного меридиана в градусы относительно истинного меридиана. После этого результат приводится к диапазону ± 180 градусов по следующему алгоритму.

Если $q > \pi$, значение примет следующий вид: $q = q - 2\pi$.

Если $q < -\pi$, значение примет следующий вид: $q = q + 2\pi$.

Вычисление итогового положения воздушного судна внутри виртуальной сцены будет происходить по следующим формулам:

$$\begin{aligned} x &= -(-d\sin(\psi)\cos(2\pi - q) + d\cos(\psi)\sin(2\pi - q)); \\ z &= -(-d\sin(\psi)\cos(2\pi - q) - d\cos(\psi)\sin(2\pi - q)); \\ y &= altitude, \end{aligned} \quad (4)$$

где x, y, z – преобразованные координаты модели; $altitude$ – высота модели над уровнем моря.

Значения крена (*roll*), рысканья (*yaw*), тангажа (*pitch*) будут преобразованы из радиана в градусы следующим образом:

$$\begin{aligned} rx &= \frac{180^\circ}{\pi} pitch; \\ ry &= -\frac{180^\circ}{\pi} yaw; \\ rz &= -\frac{180^\circ}{\pi} roll, \end{aligned} \quad (5)$$

где rx, ry, rz – значения отклонения модели в градусах от плоскостей OX, OY, OZ соответственно.

Интервалы во времени между получением новых данных могут накладывать ограничения на плавность движения виртуальной камеры [10]. Это связано с тем, что математическая модель работает с постоянной частотой, а система визуализации на протяжении всего рабочего цикла может изменять частоту обновления кадров. Для сохранения плавности движения необходимо использовать промежуточные значения координат, которые будут рассчитываться с помощью интерполяции.

Заключение

Для воссоздания ощущений реального мира в процессе тренировочных занятий необходимо воссоздавать сигналы, которые могут быть получены оператором во время выполнения задач. Чтобы повысить эффективность использования ресурсов тренажера, необходимо определить, какие именно группы объектов наиболее важны для достижения целей тренировки, и, следовательно, детально воспроизводить именно эти группы.

Разработанная система расширений (плагинов) позволила в зависимости от вычислительной нагрузки реализовать обработку данных на одном аппаратном комплексе или использовать для каждой подсистемы отдельный комплекс. Плагины не зависят друг от друга и могут быть добавлены или удалены без нарушения целостности работы системы. Описанные в статье подсистемы ТОС могут работать на аппаратных комплексах с различными архитектурами процессоров, способами хранения байтов в памяти (*big-endian* или *little-endian*) и на различных операционных системах.

Представлен алгоритм для преобразования географических координат, полученных от математической модели, в систему координат подсистемы визуализации. На следующем этапе развития прототипа ТОС планируется разработать и внедрить систему моделирования атмосферных явлений и систему процедурной генерации земной поверхности.

Литература

1. Rolfe J.M. and Caro P.W. Determining the training effectiveness of flight simulators: some basic issues and practical development. *Appl. Ergon*, 1982, no. 13, pp. 243–250.
2. Решетников В.Н. Тригонометрические воспоминания. Сер.: Космические телекоммуникации. СПб, 2015. 138 с.
3. Гиацинтов А.М., Мамросенко К.А., Решетников В.Н. Инструментальные средства предтренажерной и тренажерной подготовки операторов сложных технических систем // Программные продукты, системы и алгоритмы. 2014. № 1. URL: <http://swsys-web.ru/simulator-training-operators.html> (дата обращения: 22.04.2018).
4. Allerton D. Principles of flight simulation. Chichester, UK, Wiley Publ., 2009, 471 p.
5. Kunc M., Morecroft J.D.W., Brailsford S. Special issue on advances in system dynamics modelling from the perspective of other simulation methods. *J. of Simulation*, 2018, pp. 87–89.
6. Miloš Ljumović. C++ Multithreading Cookbook. Packt Publ. Ltd, 2014, 422 p.
7. Горкин А.П. География. Современная иллюстрированная энциклопедия; [под ред. А.П. Горкина]. М.: Росмэн, 2006. 84 с.
8. Гиацинтов А.М., Родителей А.В., Агафонов Н.А. Методы визуализации погодных явлений в имитационных системах // Вестн. кибернетики. 2018. № 1. С. 61–65.
9. Акасофу С.И., Чепмен С. Солнечно-земная физика; [пер. с англ.]. М.: Мир, 1975. 512 с.
10. Olsson O., Assarsson U. Tiled Shading. *J. Graph. GPU Game Tools*, 2011, vol. 15, no. 4, pp. 235–251.

High-level architecture of training simulation systems of complex technical systems

A.V. Roditelev¹, Programmer of ORIST department, avrod_94@mail.ru
A.M. Giatsintov¹, Senior Researcher, giatsintov@niisi.ras.ru

¹ Center of Visualization and Satellite Information Technologies SRISA, Moscow, 117218, Russian Federation

Abstract. The paper provides a detailed description of the training simulation system (TSS) architecture using the example of an air simulator prototype. A TSS visualization subsystem provides visualization of external environment and a control object using display devices. It should provide reproduction of the created virtual scene with a sufficiently detailed content that allows TSS operators to perform the assigned tasks successfully.

The authors give the requirements for TSS subsystems, including those for the TSS visualization subsystem. The developed architecture avoids high coupling of components and provides a unified approach to managing hardware, such as various input devices.

Usually, a device has some peculiar properties: specific control software, closed information exchange protocols, different connector types. The developed plugin management systems allows taking into account various hardware features without modifying the main module and other subsystems. The created control interface works with pluggable modules. Plugins are self-sufficient and can be added or removed without violating the integrity of the system.

Depending on the workload, data processing can be organized on one machine or each subsystem can operate on a separate machine. Each subsystem is a standalone software complex that may be developed by a third-party developer.

The main module and its subsystems can operate on hardware complexes with different processor architectures, endianness (little or big) and operating systems. The paper also describes an algorithm that transforms geographic coordinates received from the modeling subsystem to the coordinate system used by the visualization subsystem.

Keywords: TSS, Training simulation systems, visualization system, 3D modeling, trainer, simulation systems.

References

1. Rolfe J.M., Caro P.W. Determining the training effectiveness of flight simulators: some basic issues and practical development. *Appl. Ergon.* 1982, no. 13, pp. 243–250.
2. Reshetnikov V.N. *Trigonometric Memories. Ser. Space Telecommunications.* St. Petersburg, RIP SPB Publ., 2015, 138 p.
3. Giatsintov A.M., Mamrosenko K.A., Reshetnikov V.N. Tools for pre-training and simulator training for operators of complex technical systems. *Software, Systems and Algorithms.* 2014, no. 1. Available at: <http://swsys-web.ru/simulator-training-operators.html> (accessed April 22, 2018).
4. Allerton D. *Principles of Flight Simulation.* Chichester, UK, Wiley Publ., 2009, 471 p.
5. Kunc M., Morecroft J.D.W., Brailsford S. Special issue on advances in system dynamics modelling from the perspective of other simulation methods. *J. of Simulation.* 2018, pp. 87–89.
6. Ljumović M. *C++ Multithreading Cookbook.* Packt Publ. Ltd, 2014, pp. 422.
7. Gorkin A.P. *Geography. Contemporary Illustrated Encyclopedia.* Moscow, Rosmen Publ., 2006, 84 p.
8. Giatsintov A.M., Roditelev A.V., Agafonov N.A. Methods of visualizing weather phenomena in imitation systems. *Proc. in Cybernetics.* 2018, no. 1, pp. 61–65 (in Russ.).
9. Akasofu S.I., Chapman S. *Solar-Terrestrial Physics.* Oxford, 1972 (Russ. ed.: Moscow, Mir Publ., 1975, 512 p.).
10. Olsson O., Assarsson U. Tiled Shading. *J. Graph. GPU Game Tools.* 2011, vol. 15, no. 4, pp. 235–251.

Примеры библиографического описания статьи

1. Родителев А.В., Гиацингов А.М. Высокоуровневая архитектура тренажерно-обучающих систем сложных технических комплексов // Программные продукты и системы. 2018. Т. 31. № 3. С. 439–443. DOI: 10.15827/0236-235X.123.439-443.

2. Roditelev A.V., Giatsintov A.M. High-level architecture of training simulation systems of complex technical systems. *Software & Systems.* 2018, vol. 31, no. 3, pp. 439–443 (in Russ.). DOI: 10.15827/0236-235X.123.439-443.