



Таблица 4 – Результаты корреляционного анализа

№	Признаки	Коэффициент корреляции Спирмена (основная группа)	Коэффициент корреляции Спирмена (группа сравнения)	Коэффициент корреляции Спирмена (основная группа и группа сравнения)
1	1, 2	0,06	0,14	0,83
2	1, 3	0,29	-0,13	0,78
3	1, 4	0,07	0,07	0,79
4	1, 5	0,20		0,61
5	2, 3	0,01	0,22	0,75
6	2, 4	-0,39	0,08	0,64
7	2, 5	0,20		0,58
8	3, 4	-0,06	-0,01	0,62
9	3, 5	0,01		0,51
10	4, 5	-0,15		0,44
11	6, 10	-0,41	-0,57	-0,61
12	9, 10	-0,38	0,48	0,18

В ходе многоэтапной эвристической процедуры отбора признаков, использующей критерий дискриминантного анализа и результаты корреляционного анализа, было установлено, что признаками, наилучшим образом разделяющими участников основной группы и группы сравнения, являются вопросы анкеты: «Антикоагулянтная терапия» (J = 9,48), «Как вы оцениваете свой уровень знаний о ФП?» (J = 5,28), «Насколько важно, по вашему мнению, регулярно принимать препарат для профилактики инсульта в соответствии с назначениями?» (J = 3,11), «Как вы оцениваете свои знания о риске инсульта как об основном осложнении ФП?» (J = 2,62), «Удовлетворенность антикоагулянтной терапией» (J = 1,45).

На основе отобранных признаков можно сделать вывод о том, что пациенты, прошедшие обучение, имеют более высокий уровень знаний о ФП и риске инсульта как об основном осложнении ФП, тщательнее следуют рекомендациям врача по приёму препарата и, как следствие, имеют большую удовлетворённость антикоагулянтной терапией.

Литература

1. Гайдель, А.В. Исследование текстурных признаков для диагностики заболеваний костной ткани по рентгеновским изображениям [Текст] / С.С. Первушкин // Компьютерная оптика. – 2013. – Т.37. – №1. – С.113–119.
2. Fukunaga, K. Introduction to statistical pattern recognition [Текст] / K. Fukunaga. – San Diego: Academic Press, 1990. – 592p.
3. Каримов, Р.Н. Статистика для врачей в понятном изложении [Текст]: руководство / Каримов Р.Н., Шварц Ю.Г. – Саратов: Изд-во Саратовского мед. ун-та, 2014. – 460 с.



А.В. Колсанов¹, А.К. Назарян¹,
А.В. Иващенко², Н.А. Горбаченко², А.С. Черепанов²

АРХИТЕКТУРА СРЕДСТВ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ХИРУРГИЧЕСКИХ ТРЕНАЖЕРОВ

(Самарский государственный медицинский университет,
Самарский национальный исследовательский университет имени академика
С.П. Королёва)

В настоящее время симуляционные технологии широко применяются в высшем медицинском образовании. В частности, широко распространены лапароскопические и эндоваскулярные тренажеры [1 – 3], используемые на различных этапах додипломного и последипломного образования. Однако, необходимость постоянного совершенствования существующих хирургических тренажеров и разработки новых, расширение перечня учебных кейсов и применение новых трехмерных моделей органов человеческого тела, с учетом различия методик обучения в разных университетах, обуславливает новые требования к программному обеспечению хирургических тренажеров, такие как интероперабельность, открытая архитектура и возможность функционирования в едином информационном пространстве.

В результате обобщения этих требований, а также опыта по внедрению симуляционных технологий, была сформулирована идея создания средств разработки программного обеспечения (СРПО или SDK – software development kit), предназначенных для адаптации существующих хирургических тренажеров при их внедрении в учебный процесс и создании качественно новых симуляционных решений для высшего медицинского образования. Назначение разработки – предоставить возможность быстрого и удобного создания новых хирургических тренажеров и трехмерных атласов на базе существующих разработок и аппаратно-программных компонентов. В данной статье представлены основные принципы проектирования расширяемой программной архитектуры СРПО в рамках решения поставленной задачи.

В предлагаемом подходе к построению распределенной архитектуры платформы для моделирования операционных случаев будем использовать шаблон проектирования «Сущность-компонент» [4]. Все объекты, описываемые в операционном случае (кейсе) (инструменты, органы, элементы окружения) являются контейнерами, не имеющими собственного поведения. На этапе моделирования с ними ассоциируются аспекты поведения, реализуемые в виде компонентов. Каждый компонент характеризуется именем и типом и реализует один или несколько интерфейсов для взаимодействия, которые могут быть получены путем отправки соответствующего запроса каждому компоненту.

Ответственность за создание и контроль жизненного цикла компонента возлагается на подсистемы, которые обслуживают поведение всех компонентов определенной группы. Пример подсистемы: физическая сцена, обеспечиваю-



щая создание, уничтожение и контроль компонентов, отвечающих за моделирование динамики твердых и мягких тел.

Пример отношения подсистем и компонентов представлен на рисунке 2. Подсистемы регистрируются в ядре платформы при подключении программных модулей, содержащих эти подсистемы. В результате этого в системе появляются новые аспекты поведения объектов.

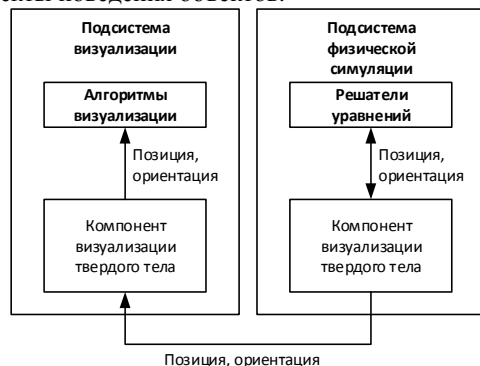


Рис. 2. Отношение подсистем и компонентов

Программная логика тренажера разделяется на базовую (управление памятью, ресурсами, созданием и удалением компонентов, объектов сцены, ведение журналов работы приложения и т.д.) и специфичную (алгоритмы визуализации, физического поведения, контроля сценария операции, поведения инструментов на сцене и т.д.). Базовая логика реализуется в ядре системы и обеспечивает взаимодействие между объектами. Специфичная логика полностью реализуется в подключаемых модулях, что обеспечивает возможность разработки модулей, работающих с наиболее подходящими для моделирования конкретного операционного случая внешними зависимостями (например, с физической библиотекой PhysX или Bullet).

Использование этого подхода имеет ряд преимуществ:

- отсутствие необходимости в глубокой иерархии классов, что уменьшает дублирование логики при большом количестве различных аспектов поведения объектов;
- возможность определения поведения объекта без перекомпиляции исходного кода, что уменьшает накладные затраты на разработку операционного случая;
- возможность распространения данных и алгоритмов в виде подключаемых модулей, легко интегрируемых в систему для расширения ее функциональности без масштабных изменений уже существующих частей системы.

Недостатками данного подхода являются:

- взаимодействие между компонентами сопряжено с накладными расходами времени выполнения из-за отсутствия знания о потребителях и поставщиках



данных (разрешение имен, контроль типов данных), поэтому требует от программиста минимизации этого взаимодействия;

- многообразие различных по своей природе компонентов делает необходимой разработку механизма хранения и получения метаданных о свойствах компонента для идентификации типа и контроля платформой допустимых сценариев взаимодействия компонентов между собой.

Ядро системы предоставляет низкоуровневые базовые службы, не привязанные к моделируемой предметной области и обеспечивающие среду для взаимодействия объектов друг с другом. Базовый каркас системы предоставляет стандартные службы, общие для моделирования широкого круга операционных случаев. К таким службам относятся различные подсистемы визуализации (OpenGL, OGRE), подсистемы физической симуляции (PhysX, Bullet, SOFA), подсистемы управления сценариями (Lua, SquitrelScript), библиотеки графического интерфейса пользователя (SDL, Qt) и т.д.

Каркас операционного блока представляет собой набор аспектов поведения, специфичных для конкретного типа операций. Например, каркас эндоскопического блока может содержать алгоритмы взаимодействия органов и инструментов (захват, разрез, пункция), алгоритмы контроля состояния пациента и качества выполнения оперативных действий оператора, модули взаимодействия с аппаратной частью тренажера. Специфичные алгоритмы конкретного операционного случая могут включать в себя детекторы успешности завершения операции (например, контроль видимости объекта при отработке навыка позиционирования камеры), контроллеры параметров конкретной операции (например, корректность позиционирования эндоскопических клипс на сосудах) и т.д.

Для каждого из уровней определяются свои интерфейсы и реализации этих интерфейсов, что позволяет иметь возможность как использования, например, физической подсистемы через ее обобщенный интерфейс, при этом имея возможность выбора конкретной реализации (PhysX, Bullet, SOFA) под нужды моделируемого случая, так и использования расширений, специфичных для какой-то конкретной подсистемы.

Оценка действий пользователей с использованием методов интервального корреляционного анализа [5] позволяет адаптировать сценарии к текущему уровню владения методиками хирургического вмешательства.

Результаты практической реализации предложенного подхода показывают, что он может быть использован для описания различных по природе аспектов поведения объектов сцены в рамках единой архитектуры, обеспечивающей хорошую гибкость настройки поведения и активное повторное использование готовых компонентов. При этом подход обладает относительно небольшими накладными расходами по сравнению с описанием аспектов поведения непосредственно в объектах сцены без использования динамического связывания.

Предложенные принципы проектирования расширяемой программной архитектуры средств разработки программного обеспечения для самостоятельного формирования медицинским сообществом решений в среде симуляционных технологий в медицине позволяют сократить временные затраты и трудо-



емкость создания новых хирургических тренажеров, а также адаптации существующих симуляционных решений при их внедрении и практическом использовании.

Данная работа поддержана грантом Минобрнауки России 2014-14-579-0003, соглашение 14.607.21.0007.

Литература

1. Колсанов А.В., Яремин Б.И., Воронин А.С., Черепанов А.С., Иващенко А.В., Сапцин Н.В. Программное обеспечение тренажера эндоваскулярной хирургии // Программные продукты и системы, 2013. – № 2. – с. 262 – 267
2. Колсанов А.В., Чаплыгин С.С., Иващенко А.В., Кузьмин А.В., Горбаченко Н.А., Милюткин М.Г. Программное обеспечение тренажера лапароскопической хирургии // Программные продукты и системы, 2013. – № 2. – с. 267 – 270
3. Колсанов А.В., Иващенко А.В., Кузьмин А.В., Черепанов А.С. Комплекс «Виртуальный хирург» для симуляционного обучения хирургии // Медицинская техника, 2013. – № 6. – с. 7 – 10
4. Nystrom R. Component [Электронный ресурс] // Game Programming Patterns. URL: <http://gameprogrammingpatterns.com/component.html> (дата обращения: 12.11.2015).
5. Иващенко А.В. Интервально-корреляционный анализ ритмичности взаимодействия в интегрированной информационной среде предприятия / Системы управления и информационные технологии, 2010, № 1(39) – с. 32 – 36

Н.И. Лиманова, С.Г. Атаев

АЛГОРИТМ ПРОГРАММНОГО АНАЛИЗА И МЕТОД ПАРАМЕТРИЗАЦИИ ОБЪЕКТОВ КОМПЬЮТЕРНОЙ ТОМОГРАФИИ

(Поволжский государственный университет
телекоммуникаций и информатики)

В настоящее время компьютерная томография (КТ) является неотъемлемой частью амбулаторной и стационарной медицинской помощи. Показания к её проведению определяет широкий круг врачей многих специальностей. Предложенный авторами метод разработан для рентгеновской компьютерной томографии, однако он не привязан к способу получения томографического изображения и может быть адаптирован для других видов томографии.

Послойная структура томографических снимков дает возможность специалистам получать исчерпывающую информацию об объектах, изображённых на снимках. Толщина каждого среза может варьироваться и выбирается в зависимости от специфики исследуемых органов, однако имеют место и ограничения, налагаемые томографом. Особенностью томографических снимков является размещение изучаемых объектов сразу на нескольких слоях. В настоящее



время диагностика с применением послойных томографических снимков проводится медицинским персоналом визуально. Послойная структура томограмм обуславливает сложность восприятия специалистом информации об исследуемых объектах, что затрудняет процесс проведения диагностики.

В работе предложены метод и алгоритм, позволяющие проводить анализ объектов, изображённых на снимках компьютерной томографии, а также выполнять их параметризацию.

Каждый слой проекции томограммы даёт точное значение плотности любого из пикселей на снимке, которые потом отображаются как светлые и тёмные оттенки серого. Чем светлее оттенок серого, тем плотнее ткань в пределах пикселя [1]. Таким образом, наличие информации о множестве вокселей (аналог пикселя для трёхмерного пространства: область, проецирующаяся на пиксел томографического среза), относящихся к целевому объекту снимка, даёт исчерпывающие сведения о свойствах объекта в каждой точке пространства, и делает возможным его дальнейший анализ и исследование.

Задачу параметризации объекта, изображённого на снимке, можно разделить на подзадачи определения границ объекта (представляет из себя частный случай задачи сегментации изображения [2]) и последующей параметризации этого объекта путём обращения к определённому ранее множеству вокселей. В качестве примера задачи параметризации объекта рассмотрим задачу определения объёма объекта.

В связи с тем, что разрешающая способность мониторов не способна отразить весь диапазон градаций плотности (по шкале Хаунсфилда), каждый томографический снимок имеет свой собственный диапазон плотностей, отображаемых с помощью оттенков, находящихся между чёрным и белым цветами. Данный диапазон зависит от специфики снимаемых объектов, и точная информация о плотности тканей за пределами данного диапазона теряется [3]. Границы диапазона задаются двумя значениями: центром диапазона и его шириной. Чисто белому цвету пикселя соответствует ткань, превосходящая по плотности диапазон отображения, чисто чёрному цвету — ткань, менее плотная, чем диапазон отображения. Промежуточные оттенки серого цвета пикселя проецируют значения плотности из интервала отображения и позволяют делать вывод о точной величине плотности ткани, находящейся в конкретной точке пространства.

Рассмотрим метод определения границ исследуемого объекта. Его работа начинается с загрузки пользователем упорядоченных слоёв томографического снимка в программу и указания начальной точки определения объекта. Также пользователь должен ввести показатель толщины для каждого слоя и указать граничное значение яркости пикселя (в интервале между 0 и 1, т.е. чёрным и белым цветами, соответственно) или плотности в единицах Хаунсфилда, которое послужит критерием проверки принадлежности каждого пикселя к выбранному объекту в процессе дальнейшей работы алгоритма. В список пикселей на данном слое, относящихся к целевому объекту, добавляется стартовый пиксел. Одновременно ведётся список пикселей, в который изначально поме-