Архитектура мобильного компьютерного тренажера погрузочно-разгрузочного устройства

Е.В. Долгова, Р.А. Файзрахманов, Д.С. Курушин, А.Б. Федоров, А.Ф. Хабибулин, А.А Шаронов

Мировой опыт тренажеростроения показывает высокую эффективность программ подготовки, обучения и контроля знаний, закладываемых на основе применения компьютерных моделей, объектов трехмерной графики, методов и алгоритмов имитации условий работы оператора и т. п. [1-4]. Однако проблемы вопросы построения тренажеров на основе этих технологий актуальны, поскольку существующие решения не в полной мере удовлетворяют потребностям исследуемой предметной области. И не всегда позволяют имитировать действия команды и отдельных операторов в реальном времени с высокой степенью достоверности проблематика данной работы связана с исследованием форм и методов организации обучения и построение эффективных компонентов обучающих систем на основе компьютеризированных технологий обработки информации, проработка основных элементов математического, информационного, алгоритмического, программного, технического, сетевого обеспечения.

Имитационно-тренажерный комплекс системы управления представляет собой сложный аппаратно-программный комплекс, включающий центр управления, контрольно-управляющую систему (моделирующий комплекс), систему имитации визуальной обстановки (систему визуализации). Система имитации акустических шумов воспроизводит звуки, происходящие в процессе работы системы. Система визуализации имитационно-тренажерного комплекса является одной из основных. Она производит синтез изображения трехмерного пространства на экране ноутбука. Для создания изображения система использует виртуальную трехмерную модель пространства (виртуальную 3D сцену), которая описывает параметры объектов (положение, ориентацию, форму и т.д.), источников освещения, камер, анимационных треков и других составляющих этого пространства. Для полноценной работы тренажера визуализация виртуальной сцены должна происходить в режиме реального времени и обеспечивать эффект непрерывного движения динамических объектов, что соответствует выводу изображения на экраны с частотой, не менее 30 раз в секунду.

Данная архитектура представлена на рис. 1↓.

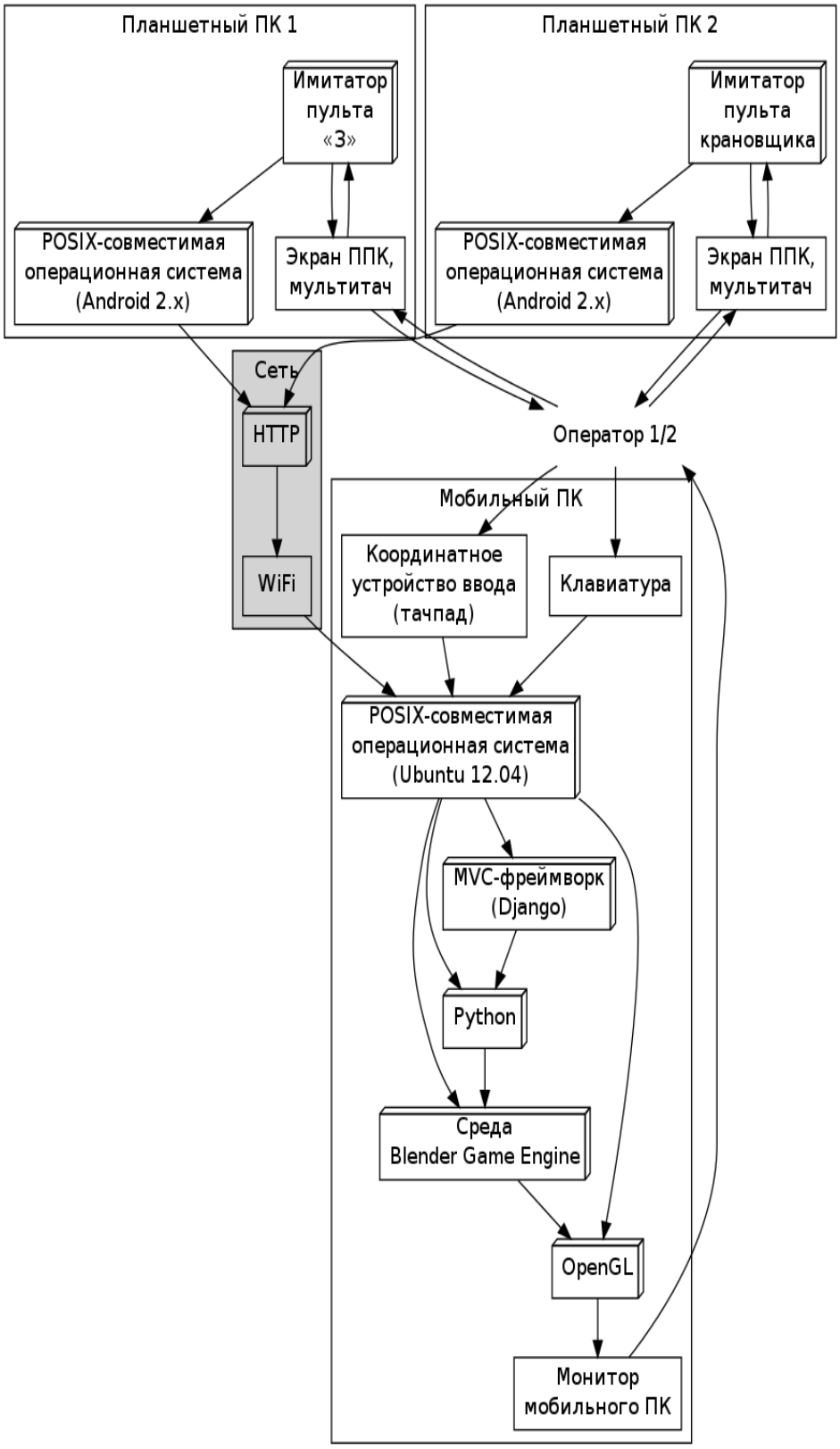

Рис. 1 – Общая архитектура тренажерного комплекса

Структурная модель тренажерного комплекса разработана на основе архитектуры тренажерного комплекса. Данная модель представлена на рисунке 2↓. Структура тренажерного комплекса состоит из: имитатора пульта крановщика; имитатора пульта «З»; средств подготовки данных; средства передачи данных; портативного персонального компьютера; беспроводной сети передачи данных.

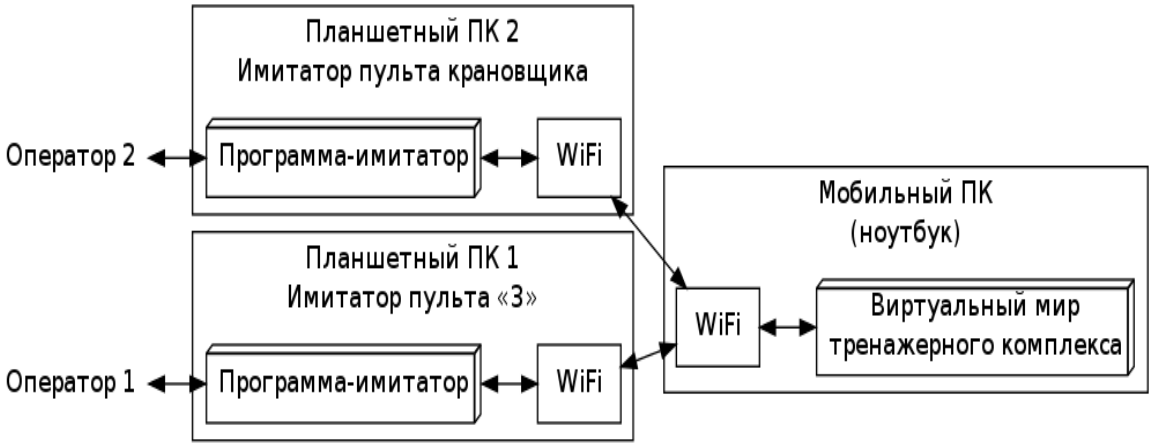

Рис. 2 – Структура тренажерного комплекса

Беспроводные сети позволяют объединить в единую систему локальные сети или отдельные компьютеры, что обеспечивает доступ большого количества (возможно десятков или сотен) удаленных абонентов к общим ресурсам сети. В нашем тренажерном комплексе они позволит соединить между собой имитаторы пультов крановщика и «З» с мобильным персональным компьютером.

При разработке тренажера рассматривались два вида топологии: «звезда» и «шина».

Топология «звезда» — базовая топология компьютерной сети, в которой все компьютеры сети присоединены к центральному узлу (сервер), образуя физический сегмент сети. Топология типа общая шина, представляет собой общий кабель (называемый шина или магистраль), к которому подсоединены все рабочие станции. При разработке тренажерного комплекса использовалась топология «Звезда», так как она более подходит для реализации обмена информацией между персональными компьютерами операторов, виртуальными пультами с сервером.

На рис. 3↓ представлена структура сети мобильного тренажера устройства. Мобильный тренажер состоит из центрального сервера, реализованного на мобильном ПК и имитаторов пультов. К серверу с использованием технологии WiFi, подразумевающей использование сетевой топологии «звезда», подключено два пульта управления.

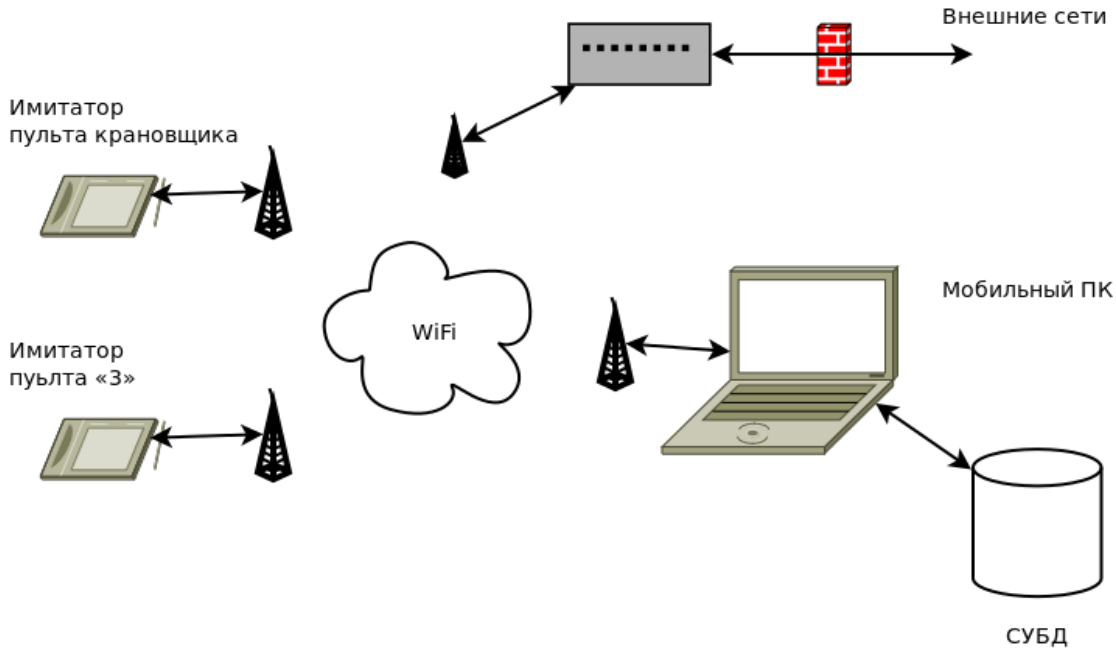

Рис. 3 – Структура сети мобильного тренажера

Кран с гидроаппаратурой служит для подъема и опускания груза, погрузки и разгрузки его с грунта (стеллажа), и машин. Управление краном производится с помощью рукояток управления, расположенных на гидрораспределителе, и лимба дросселя. Задача заключается в частичном повторении внешнего вида рукоятей управления краном.

Пользовательский интерфейс может состоять, из набора стандартных средств, входящих в состав Android SDK или же отрисовываться самостоятельно. Примером набора стандартных средств могут послужить стандартные кнопки, трэк-бары и т.п., входящие в состав Android SDK. Преимущества этого метода очевидны. Это сокращение время разработки приложения и упрощение разработки. Но также это в свою очередь приводит и к недостаткам, таким как не возможность изменения формы и функционала в широких пределах. Поэтому были созданы сторонние библиотеки позволяющие создавать пользовательский интерфейс из шаблонов, например входящих в дополнительную библиотеку AndEngine. Данная библиотека позволяет с помощью шаблонов, создать 2D, 3D пользовательский интерфейс. Этот подход сокращает время разработки 2D, 3D интерфейса и позволяет создать на основе шаблона то что необходимо, не задумываясь об ограничениях стандартных шаблонов, за счет довольно сильного усложнения программной части. Если нам не нужен большой функционал или 3D графика, намного проще создать свой собственный пользовательский интерфейс. Преимущества данного подхода — это создание именно того что нам нужно, за короткое время и усложнения программного кода, в разумных пределах. В нашем случае необходимо создать пользовательский интерфейс максимально похожий на оригинальные пульты в довольно коротко время, поэтому нам идеально подходит разработка пользовательского интерфейса самостоятельно. Пусть необходимо создать пользовательский интерфейс максимально похожий на оригинальные пульты. Из этого можно выделить основные требования к пользовательскому интерфейсу:

1. Минимальное время оклика интерфейса на действие пользователя -Обработка одновременного качания (до 5 точек)

2. Кнопки, тумблеры и индикаторы должны максимально походить на оригинальные

3. Программно имитировать действия соответствующие кнопкам, тумблерам и индикаторам

Логическую структуру пульта крановщика мы можем представить в виде схемы (рис. 4).

Интерфейс: • onDraw() • onTouchEvent()	⇔	Логика • onClick() • getSettings()

Рис. 4 – Логическая структура пульта крановщика

В методах onClick и GetSettings происходят основные процессы по взаимодействию пульта крановщика с сервером. Эти методы могут быть вызваны системой, происходить по расписанию или вызваны действиями пользователя. Основные процессы происходящие в логике системы:

1. Передача информации между интерфейсом и сервером, этот процесс заключается в проверке и форматированию исходящих запросов на сервер.

2. Отображение интерфейса по критериям пользователя системы

3. Формирование статуса работы системы. В системе предусмотрено 6 статусов:

1) «Не введен адрес сервера» — этот статус формируется при отсутствии адреса сервера

2) «Нет ping» — данный статус формируется если по какой то причине доступ к серверу невозможен (например не найден DNS сервер).

3) «Ожидает обработки» — после получения доступа к серверу, система ждет от него настроек системы (варианты настроек варьируются в зависимости от типа интерфейса)

4) «Загрузка» — на данном этапе происходит загрузка с сервера так называемого скина (если его нет в системе), который будет отображаться в интерфейсе.

5) «Готов» — данный статус присваивается системе, которая успешно прошла настройки,

и служит для обозначения того, что пользователь может приступить к работе.

6) «Ошибка» — этот статус, служит сигналом для администратора системы, говоря о том, что на каком то из этапов произошла ошибка и необходимо ручное устранение ошибки.

4. Модульная подсистема, подсистема подключения модулей для расширения возможности системы (модуль проверки индексации).

Логическая часть интерфейса обрабатывает полученные от метода onTouchEvent запросы и относительно них перерисовывает интерфейс и отсылает данные в метод onClick. В части оформления, содержаться шаблоны элементов интерфейса. Данная схема очень удобна в виду того, что мы отделяем визуальную часть от содержимого. Это влияет на понятность кода, а так же на возможности, расширяемости, редактирования и тестирования кода пульта крановщика.

Приложение состоит из трех классов (см. рис.5).

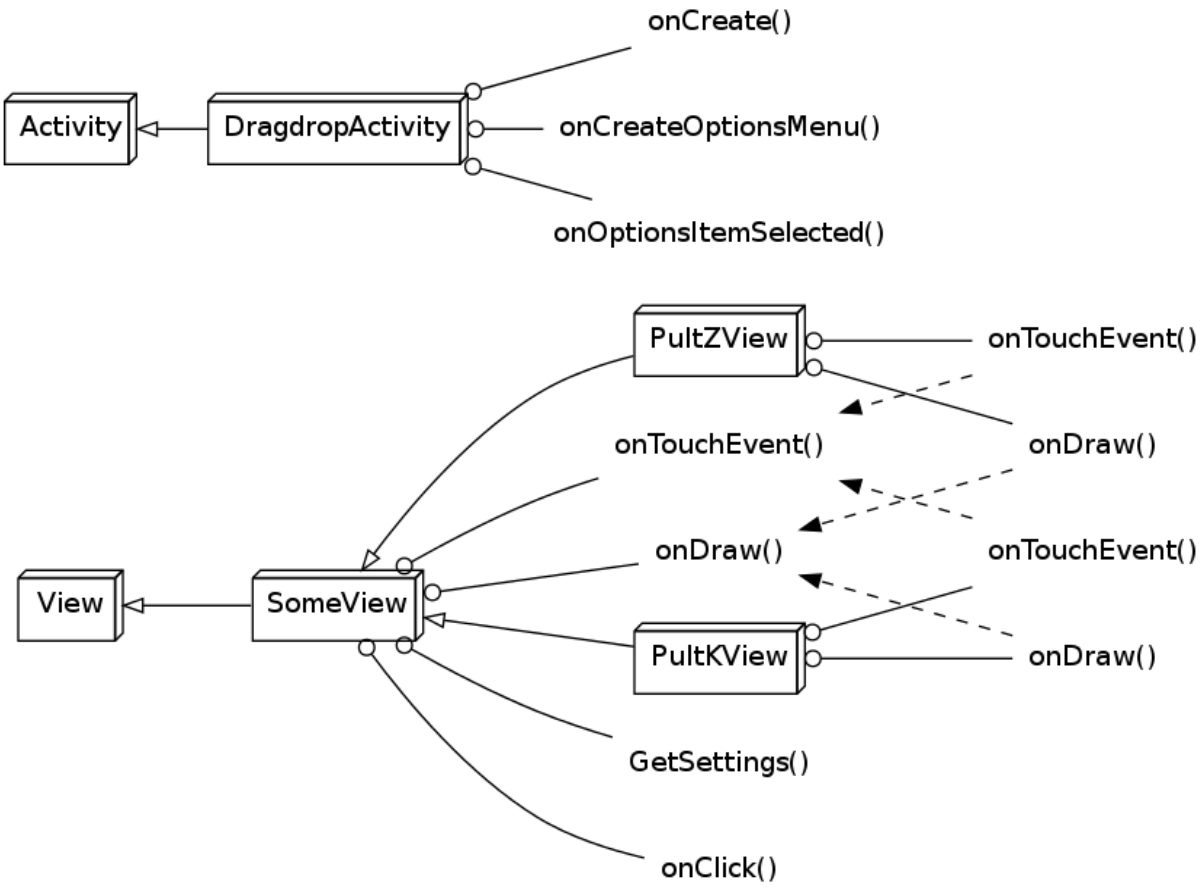

Рис. 5 – Фрагмент диаграммы классов

Основнеой класс DragdropActivity реализует «активность» приложения и служит для интеграции с платформой Android. Фрагмент листинга класса см. алг. 1.1↑. Класс DragdropActivity реализует запуск приложения-имитатора пульта и позволяет запускать диалог настройки.

Весомую часть клиентского приложения составляет графический интерфейс. Рассмотрим две самых распространенных технологии (SVG и HTML5 Canvas) для реализации графики.

Преимущества Canvas:

- высокая производительность при отрисовке любых 2D объектов;
- стабильная производительность — все есть пиксель, производительность падает только при увеличении разрешения изображения;
- можно сохранить полученное изображение в PNG или JPG файл;
- лучше всего подходит для создания растровой графики (например, в играх, фракталов и

т.п.), редактирования изображений и операций, требующих манипулирования на уровне пикселей.

Преимущества SVG:

- нет зависимости от разрешения — SVG лучше подходит для кроссплатформенных пользовательских интерфейсов, так как позволяет масштабировать изображение при различных разрешениях экрана;

- SVG очень хорошо поддерживает анимацию. Элементы могут быть анимированы с использованием описательного синтаксиса или с помощью JavaScript;

- можно получить полный контроль над каждым элементом, используя SVG DOM API в JavaScript.

SVG хранится в формате XML, что предоставляет больше возможностей браузерам по обеспечению доступности SVG документов по сравнению с элементом canvas. Таким образом, SVG выглядит лучшим решением для пользовательских интерфейсов веб-приложений.

Недостатки Canvas:

- отрисовка основана на пикселях;

- не существует API для анимации. Вам придется прибегать к использованию таймеров и других событий для обновления канвы;

- слабые возможности по рендерингу текста.

Таким образом, canvas — возможно, не самый лучший выбор, когда доступность имеет решающее значение. Канва предоставляет вам поверхность для рисования в выбранном контексте (2D и 3D). Можно указать альтернативный контент внутри элемента canvas, который будет показан браузером при невозможности отображения графики. Кроме того, вы можете выполнить проверку доступности выбранного Canvas API с помощью JavaScript. На основе этого вы можете обеспечить различную функциональность для пользователей браузеров с разной поддержкой HTML 5 Canvas;

HTML 5 Canvas не подходит для создания веб-сайтов или интерфейсов веб-приложений, так как пользовательские интерфейсы обычно должны быть динамическими и интерактивными, а Canvas требует от вас постоянной перерисовки каждого элемента в интерфейсе.

Недостатки SVG:

- низкая скорость рендеринга при увеличении сложности документа (рисунка), так как используется модель DOM;

- скорее всего, SVG не подходит для таких приложений как игры.

Каждая технология имеет свою область применения.

HTML 5 Canvas следует использовать для:

- редактирования изображений обрезки, изменения размеров, фильтров (удаления эффекта красных глаз, создания эффекта сепии, изменения цветности или яркости);

- создания растровой графики: визуализации данных, создания фракталов и графиков функций;

- анализа изображений: создания гистограмм и т.п.;

- создания игровой графики, такой как спрайты и фоны;

SVG следует использовать для:

- создания пользовательских интерфейсов веб-приложений, независимых от разрешения экрана;

- высокоинтерактивных анимированных пользовательских интерфейсов;

- графиков и диаграмм;

- редактирования векторных изображений.

Необходимый функционал нам полностью обеспечивает SVG технология. Так же в работе есть моменты, для которых оптимальными будут решения на основе HTML и CSS. Это такие вещи, как закругленные углы, переходные эффекты, тени и прозрачность, расположение на экране, отступы.

В работе также бskb использованы использовать библиотеки JavaScript — jQuery UI (например, для создания более удобного и красивого скролла).

Эти решения были успешно положены в основу создания мобильного компьютерного тренажера погрузочно-разгрузочного устройства. Таким образом, поставленная задача находит свое решение для конкретного объекта с учетом предъявленных к тренажеру требований.

Литература:

1. Долгова Е.В., Файзрахманов Р.А., Курушин Д.С., Кротов Л.Н., Федоров А.Б., Хабибуллин А.Ф., Шилов В.С., Ромин Е.А., Бакунов Р.Р., Бикметов Р.Р., Полевщиков И.С. «Моделирование динамики перемещения груза в компьютерном тренажере погрузочно-разгрузочного устройства/ Вестник МГОУ, серия "Физика-математика", N 2, 2012.

2. Орлов А.Н. Общая динамическая модель грузоподъемных кранов // В сб. «Оптимизация параметров строительных и дорожных машин». — Ярославль, изд-во Яросл. полит. ин-та, 1992. С.13-20.

3. Скрипкина М.А. Применение методологических подходов при разработке модели формирования графической компетенции курсантов военного вуза [Электронный ресурс] // «Инженерный Вестник Дона», 2010, №4. - Режим доступа: http://ivdon.ru/magazine/archive/n4y2010/257 (Доступ свободный) - Загл. с экрана. - Яз.рус

4. Файзрахманов Р.А., Бакунов Р.Р., Мехоношин А.С. Создание трехмерных моделей для системы визуализации тренажерного комплекса // Вестник ПГТУ. Электротехника, информационные технологии, системы управления. - 2011. - №5. - С. 62-69.