



OSTIS-2016

(Open Semantic Technologies for Intelligent Systems)

УДК 004.92

РЕАЛИЗАЦИЯ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ВИРТУАЛЬНЫХ СРЕД ПО ДЕКЛАРАТИВНОЙ МОДЕЛИ

Грибова В.В., Федорищев Л.А.

*Институт автоматики и процессов управления Дальневосточного отделения
Российской академии наук, г. Владивосток, Россия*

gribova@iacp.dvo.ru

fleo1987@mail.ru

В работе приводится описание подхода к разработке пользовательских интерфейсов виртуальных сред по декларативной модели. Данный подход позволяет заменить кодирование в разработке и сопровождении не только самих виртуальных сред, но и интерфейсов их естественным декларативным описанием, что открывает дополнительные возможности по гибкой настройке и управлению пользовательских интерфейсов.

Ключевые слова: интерфейсы, виртуальные среды, онтологии, облачные технологии.

Введение

В настоящее время при создании программного обеспечения большое внимание уделяется его жизнеспособности или максимально возможному увеличению жизненного цикла. Известны данные [Ogheneovo E.E., 2013], что в процессе жизненного цикла программного средства около 30% всех усилий уходит на стадию разработки и более 70% на стадию сопровождения в связи с изменениями требований пользователей, условий эксплуатации, предметной области, ошибками и дефектами в программном обеспечении

Разработка профессиональных виртуальных сред (ПВС) также связана с указанными трудностями [Трухин, 2008]. Существующие средства ориентированы на программистов, а не на экспертов предметных областей. При этом основу виртуальных сред составляют предметные знания, которые используются и реализуются в рассмотренных средствах неэффективно (встроены в код программного средства).

В лаборатории интеллектуальных систем ИАПУ ДВО РАН реализован комплекс Интернет Разработки Виртуальных Интерактивных Сред – ИРВИС [Gribova et al., 2014], в котором авторами предлагается принципиально другой подход к созданию и сопровождению виртуальных сред: разработка декларативных моделей (с привлечением экспертов предметной области) вместо прямого кодирования. Кроме того, данный комплекс работает на облачной платформе IASaaS [Клещев

и др., 2011], что обеспечивает значительное упрощение коллективной разработки и сопровождения виртуальных сред, т.к. все сервисы и инструменты находятся «в облаках».

Одна из проблем, возникших при разработке комплекса ИРВИС, заключается в сложности формирования произвольных пользовательских интерфейсов для различных сервисов виртуальной реальности. Предметные области, для которых создаются виртуальные среды, имеют различные задачи, а, следовательно, и различающийся интерфейс. В качестве возможного решения этой проблемы можно было бы предусмотреть множество заранее настроенных компонентов интерфейса для различных задач. Однако такой подход в создании интерфейса является достаточно трудоемким, т.к., например, включение-выключение нужных компонентов пришлось бы выполнять каждому пользователю. Другая проблема, возникающая при работе с интерфейсами, заключается в том, что в некоторых задачах необходимо добавлять некоторую специфическую программную логику для компонентов интерфейса.

Целью работы является описание метода создания интерфейсов виртуальных сред на основе декларативной модели и реализация данного подхода.

1. Онтологический подход к созданию и управлению виртуальных сред

Основная идея создания интерфейсов в комплексе ИРВИС заключается в использовании той

же технологии, что и для создания самих виртуальных сред: технологии декларативных моделей на основе семантических представлений.

Основные принципы подхода к генерации виртуальных сред на основе семантических представлений были рассмотрены в [Грибова и др., 2012; Грибова и др. 2015 (OSTIS)]: Логическое представление ПВС является ключевым описанием, по которому, с одной стороны, формируется визуальное представление виртуальной среды, с другой стороны, оно определяет логику ее работы. Логическое представление включает описание объектов виртуальной среды и их свойств (атрибутов), возможные действия пользователя в виртуальной среде и результаты, к которым могут привести эти действия. Вся информация, составляющая логическое представление модели ПВС, является знаниями экспертов предметной области, соответственно эксперты должны не только формировать их, но также и сопровождать в процессе жизненного цикла ПВС.

2. Онтология интерфейса

Принципиально новым расширением в разработке и управлении ПВС стала возможность декларативного формирования WIMP-интерфейса. На рис. 1 показано место декларативной модели интерфейса в общей системе программного комплекса, построенного на основе онтологического подхода.

Анализ виртуальных сред показал, что независимо от предметной области и решаемых задач, в которых используются виртуальные среды, их все можно разделить на несколько классов, каждый из которых имеет фиксированный основной функционал, и, возможно, некоторый

дополнительный. Набор фиксированных функций определен в интерпретаторе, поэтому не требует дополнительного программирования и обработки. Однако, в случае включения дополнительных функций, программист может их добавить и, если необходимо, описать соответствующий WIMP-интерфейс.

Структура описаний стандартных и расширенных задач в целом совпадают, принципиальная разница заключается в том, что решение стандартных задач поддерживаются интерпретатором модели ПВС; для включения в ПВС дополнительных или расширенных задач необходимо описать структуру интерфейса этой задачи и добавить агент, который обеспечивает ее решение (обработку).

Каждая задача может быть разделена на подзадачи. Подзадачи предназначены для группировки элементов интерфейса в семантически связанные группы, которые имеют один агент обработки для расширенных задач, либо связаны с решением стандартной подзадачи, реализованной в интерпретаторе. С каждой задачей может быть связано произвольное множество возможных фрагментов интерфейса (различных представлений), из которых разработчик может выбрать представление, удобное для конкретной реализации, либо описать это представление самостоятельно.

Каждый тип ПВС имеет свой обязательный или номинальный набор функций, необходимый для формирования данного типа ПВС. Онтология WIMP-интерфейса ПВС описывает декларативное определение компонентов разных типов ПВС, которые могут быть с успехом применены к различным виртуальным средам одновременно.

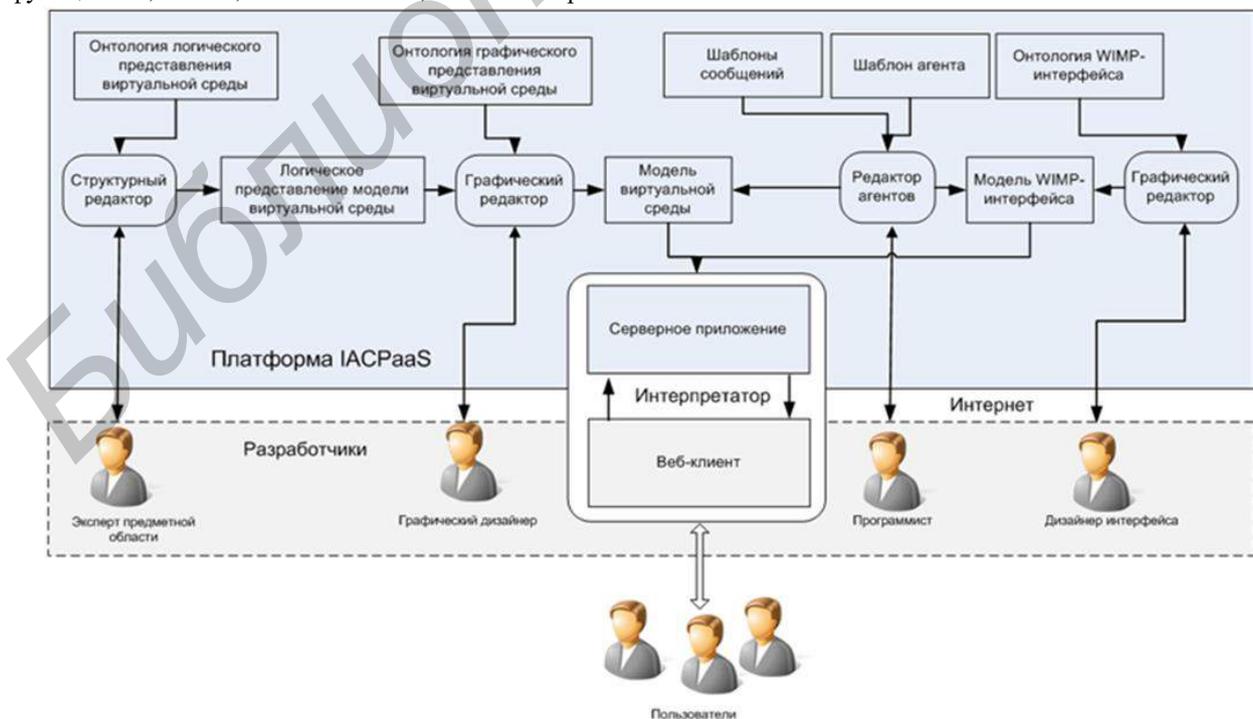


Рисунок 1 -- Модель программного комплекса

Онтология WIMP-интерфейса имеет следующий вид (см. рис. 2):

WIMP-интерфейс = *<Настройки, Стандартные компоненты, Пользовательские компоненты>*

Общие настройки = *<Настройки сервиса, Настройки виртуальной среды>*.

Настройки сервиса ∈ *<Ширина сервиса, Высота сервиса>*.

Настройки виртуальной среды = *<Ширина окна виртуальной среды, Высота окна виртуальной среды>*.

Стандартные компоненты = { *Объект VR_p* ∈ *Стандартный элемент* }ⁿ, n = количество стандартных элементов;

Пользовательские компоненты = { *Компонент_p* ∈ *Пользовательский компонент* }ⁿ, n = количество пользовательских компонентов;

Стандартный элемент ∈ { *Меню, Управление объектами среды, Библиотека, Свойства объекта среды, Координаты объекта, Тест* }.

Пользовательский компонент ∈ { *Текст, Кнопка, Список, Выпадающий список* }.

Кнопка = *<Текст, Координаты, Внешнее действие, Внутреннее действие, Ответное внутреннее действие>*.

Внешнее действие – указывает на внешний агент и описанную в нем функцию реализации необходимого действия, которое должно выполняться в результате нажатия на кнопку.

Внутреннее действие ∈ { *Функция АПИ* }_i где i < количество функций виртуальной среды, доступной для использования в интерфейсе пользователя.

Ответное внутреннее действие отличается от обычного внутреннего действия тем, что выполняется не после нажатия на интерфейсный элемент (кнопку), а после завершения внешнего действия и возвращения соответствующего результата от внешнего агента на клиент виртуальной среды.

Ответное внутреннее действие ∈ { *Функция АПИ* }_i где i < количество функций виртуальной среды, доступной для использования в интерфейсе пользователя

Список = *<Текст, Координаты, Внешнее действие, Внутреннее действие>*

Для описания интерфейса дизайнер интерфейса, используя графический редактор, формирует необходимый интерфейс. В основном проектирование интерфейса сводится к выбору представления интерфейса из множества предлагаемых вариантов. При этом библиотека представлений интерфейса является расширяемой.

Если разработчику необходимо свое уникальное представление, то, задавая тип задачи, решаемой с помощью виртуальной среды, он может спроектировать свое уникальное представление, выбрав интерфейсные элементы и определив их атрибуты.

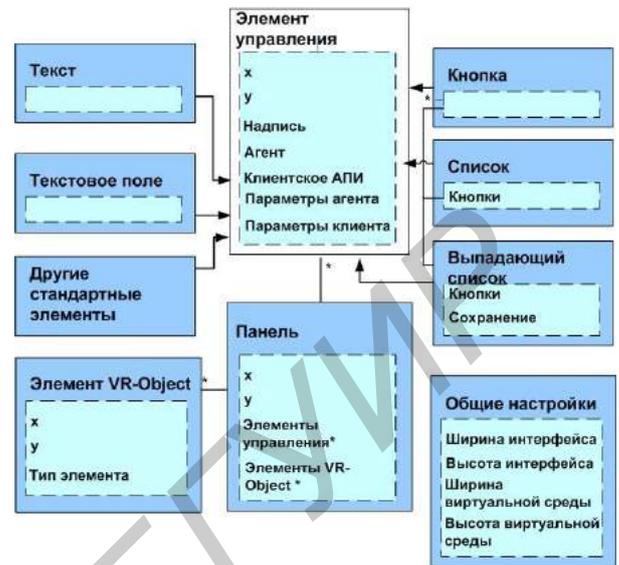


Рисунок 2 - Онтология интерфейса

3. Метод реализации

Реализация программного средства разработки интерфейсов виртуальных сред с использованием онтологического подхода выполняется в виде отдельного программного сервиса-редактора на платформе IACPaaS. Данный редактор имеет графический WIMP-интерфейс с визуальным представлением всех элементов интерфейса, заданных в онтологии.

Реализовывается данный редактор как клиент-серверное приложение (аналогично самим виртуальным средам и другим сервисам платформы IACPaaS) на основе мультиагентной технологии на серверной стороне и с веб-клиентом на клиентской стороне.

Использование мультиагентного подхода совместно с семантическим представлением информации на всей платформе IACPaaS позволяет легко формировать, менять, загружать, использовать любую декларативную информацию, хранимую в семантическом представлении, в том числе, декларативные модели интерфейсов ПВС. Это означает, что единовременно созданная декларативная модель интерфейса может с успехом использоваться, обрабатываться и меняться во множестве других сервисов платформы, включая сервисы самих виртуальных сред (редактор ПВС и интерпретатор ПВС).

На клиентской стороне (в веб-клиенте сервиса) интерфейс создается визуально по технологии WYSIWYG (what you see is what you get - что видишь, то и получишь) (см. рис. 3). При этом все данные об изменениях интерфейса отправляются на

серверный скрипт (агент-контроллер), который формирует декларативное представление модели данного графического интерфейса. Механизм Drag&Drop позволяет перетаскивать и размещать элементы в любое место интерфейса.

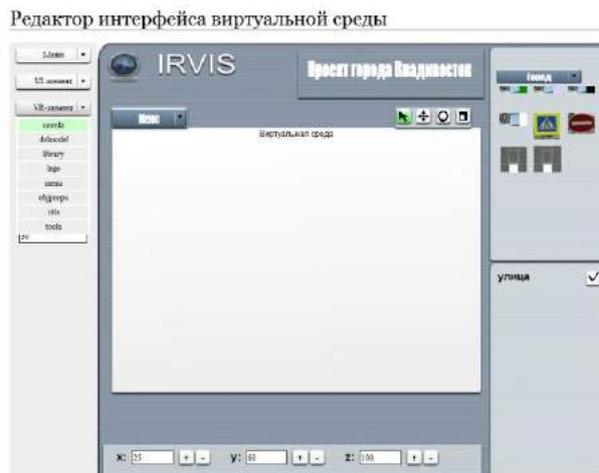


Рисунок 3 - Редактор интерфейса ПВС

4. Применение интерфейсов

На базе комплекса ИРВИС разработаны и находятся в разработке различные виртуальные среды [Грибова и др., 2014]: обучающий тренажер по классическим методам исследования зрения в офтальмологии, обучающий тренажер по неврологии, виртуальная химическая лаборатория, виртуальные среды по моделированию виртуального города, по созданию модели транспортных потоков. Приведенные примеры показывают разнообразные задачи, которые, соответственно, требуют разных интерфейсов: 1) в обучающих тренажерах требуется интерфейс тестирования и обучения, 2) в модели виртуального города требуются способы управления объектами города и изменения их с учетом информации, полученной из внешнего источника (по отношению к сервису ПВС), 3) в моделях транспортных потоков требуются интерфейсные средства управления визуализации потоков (например, в виде линий или графиков).

Заключение

В данной статье описаны принципы декларативного подхода для разработки интерфейсов такого класса интеллектуальных систем как виртуальные среды. Приводится онтология интерфейсных компонентов виртуальных сред. Описывается метод реализации пользовательских интерфейсов виртуальных сред по декларативным моделям.

Работа выполнена при финансовой поддержке РФФИ, проект № 15-07-03193 и ДВО РАН проект «Дальний Восток».

Библиографический список

[Gribova et al., 2014] Gribova V, Fedorischev L. Cloud service for development of virtual interactive environments // Proc. of 2014 2nd International Conference on Advanced ICT for Education (ICAICTE 2014), August 16-17, 2014 in Dalian, China. Atlantis Press, 2014. Vol. 1. Pp. 20-23

[Ogheneovo E.E., 2013] Ogheneovo E.E. Software Maintenance and Evolution: The Implication for Software Development // Journal of Industrial and Academic Research . – 2013. –Vol.7 No. 1

[Грибова и др., 2012] Грибова В.В., Федорищев Л.А. Обучающие виртуальные системы и средства их создания // Вестник информационных и компьютерных технологий. – 2012. – №3. – С. 48-51

[Грибова и др., 2014] Грибова В.В., Федорищев Л.А. Создание интеллектуальных сервисов с виртуальной реальностью // Материалы 14-й национальной конференции по искусственному интеллекту с международным участием (КИИ 2014). – 24-27 сентября 2014 Казань, Россия.

[Грибова и др., 2015] Разработка и генерация виртуальных сред на основе семантических представлений // Материалы V международной научно-технической конференции “Открытые семантические технологии проектирования интеллектуальных систем” (OSTIS-2015), Минск, 2015. С. 183-187

[Клещев и др., 2011] Клещев, А.С. и другие Облачная платформа для разработки и управления интеллектуальными системами // Труды конференции OSTIS-2011. С. 5-14.

[Трухин, 2008] Трухин, А.В. Анализ существующих в РФ тренажерно-обучающих систем // Открытое и дистанционное образование. - 2008. - №1. - С. 32-39

REALIZATION OF USER'S INTERFACES OF VIRTUAL ENVIRONMENTS BY DECLARATIVE MODEL

Gribova V.V. , Fedorischev L.A

Institute of Automation and Control Processes, the Far Eastern Branch of Russian Academy of Science Vladivostok, Russia

gribova@iacp.dvo.ru

fleo1987@mail.ru

The article presents an approach to development of user's interface for virtual environments by declarative model. The approach is to change coding of a program by declarative description of not only the program but it's interface too. It opens additional possibilities on management of users interfaces.

Introduction

At present virtual environments is perspective direction of development. However, development and maintenance of virtual interactive environments and it's interfaces are still a complicated process. We developed the new program complex IRVIS on the cloud platform IACPaaS. At this paper we want to show a new approach to generate user's interfaces for virtual environments that are developed by IRVIS.

Conclusion

Principles of applying of semantic nets to interfaces of virtual environments are described at the article. There are presented the method of development, formalized semantic nets and an example of interfaces of virtual environment.