**Virtual World Modeling Technology (VWMT)**

**(Russian variant)**

Authors: M. Грузман

O. Гибаев

O. Тарандевич

# Content

[1. Введение 3](#_Toc360621255)

[1.1 VWM Technology. General 3](#_Toc360621256)

[1.2 Архитектура VWMT проекта 4](#_Toc360621257)

[2 Модели игр в клиент-серверной архитектуре проекта 6](#_Toc360621258)

[2.1 Клиент-приложение 6](#_Toc360621259)

[2.2 Сервер-приложение 7](#_Toc360621260)

[3 Процесс разработки 9](#_Toc360621261)

[3.1 Описание игры на естественном языке (phase 1) 9](#_Toc360621262)

[3.2 Описание игры на языке VWML (phase 2) 10](#_Toc360621263)

[3.3 Интеграция и финальная отладка (phase 3) 10](#_Toc360621264)

# Введение

## VWM Technology. General

Нами предложен язык VWML (Virtual World Modeling Language). Этот язык и базирующаяся на нем технология VWM technology представляют собой инструментарий, ориентированный на «создание» виртуальных миров; в частности – на разработку игр различных типов.

На языке VWML описываются сущности мира и отношения между сущностями. Кроме того, виртуальные миры «населяют» различные существа, которыми могут быть как люди (в частности, игроки), так и виртуальные персонажи. События, происходящие в виртуальном мире, продуцируются его природой (законами) и поведением различных существ в нем.

Можно сказать, что VWML ориентирован на описание виртуальных миров, их эволюцию и на поведение «живущих» в нем существ.

Сказанное выше позволяет нам назвать программирование на VWML антропоморфным программированием.

Инструментарий, предложений нами, базируется на идеях математической логики, теории конечных автоматов и функционального подхода к программированию.

Отметим, что проектируемый мир является моделью некоторой формальной теории, реализацией некоторой общей схемы.

Рассматривая различные реализации одной и той же схемы, разнообразные ее сюжетные наполнения и интерфейсные вариации, мы получаем возможность создания «семейств игр».

Суммируя сказанное выше, сформулируем концептуальные положения VWM technology.

Парадигма 1

Виртуальный мир - суть совокупность сущностей, индивидов, которые находятся между собой в определенных отношениях.

Парадигма 2

Виртуальный мир - суть модель некоторой формальной теории. Аксиомы этой теории – законы виртуального мира.

Парадигма 3 (эволюционная парадигма)

Всякое изменение в виртуальном мире – это изменение совокупности индивидов (сущностей) «населяющих» этот мир либо изменение отношений между сущностями.

Парадигма 4 (парадигма неизменных законов)

Эволюция мира не изменяет его законов.

Виртуальный мир – суть «переменная» модель некоторой формальной теории.

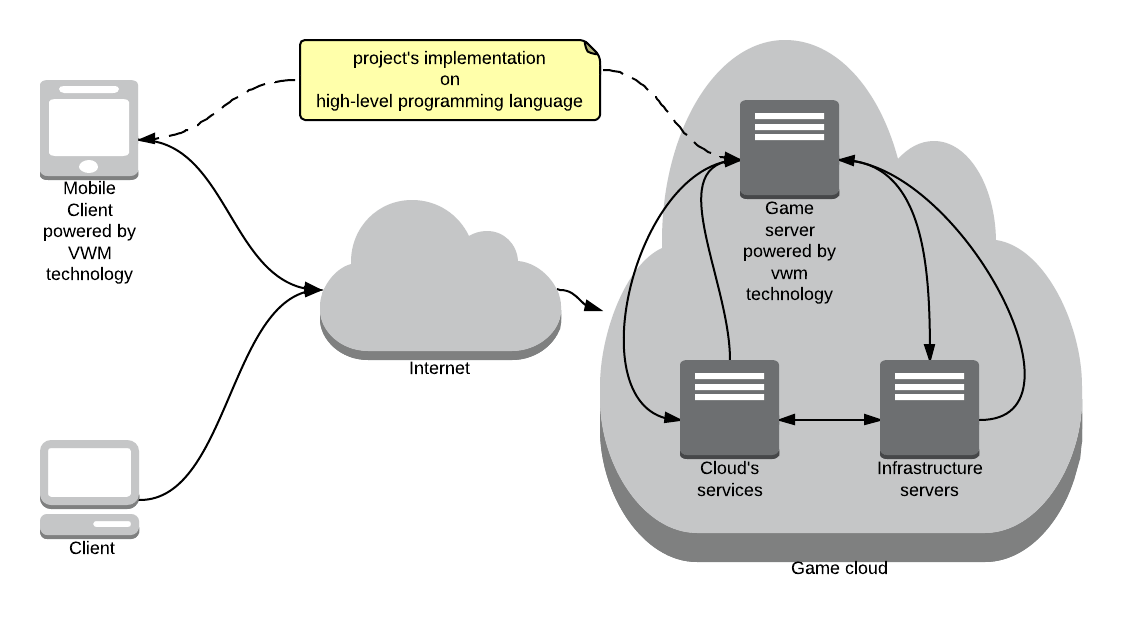
Парадигма 5 (парадигма происходящих событий)

Модифицируя интерпретацию формальной теории мы говорим о событиях, происходящих в виртуальном мире.

События, происходящие в виртуальном мире, инициируются разумными существами, «населяющими этот мир».

## Архитектура VWMT проекта

Система транслирует мир описанный языком VWML в языки высокого уровня, такие как: java, java-script, c++; тем самым мы получаем возможность реализации VWML проектов на любой платформе (сохранить фактор кросс-платформенности).



[Рис 1 Клиент серверная архитектура VWMT проекта](file:///C:\Users\vin16085\AppData\Local\Temp\VWM.docx)

Так как технология позволяет генерировать VWML в код на языке высокого уровня, мы можем получить код ядра проекта, который может «жить» в произвольной операционной среде: как на стороне клиента, так и на стороне сервера (см Рис 1.)

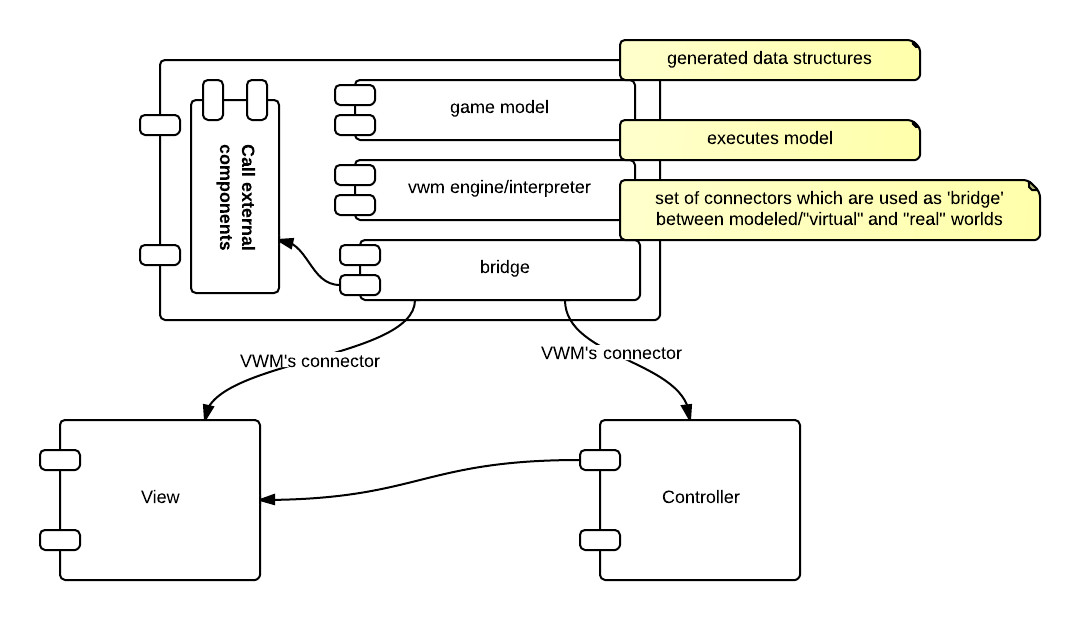
Система легко интегрируется с существующими платформами для разработки игр, такими как ODOBO.

# Модели игр в клиент-серверной архитектуре проекта

Рассмотрим процесс разработки и интеграции для клиент-приложений и сервер-приложений.

## Клиент-приложение

Классическим паттерном для разработки клиент приложения является составной паттерн MVC (Model-View-Controller); где Model – определяет модель (в нашем случае модель виртуального мира, в частности игры) , View – пользовательский интерфейс (UI) и Controller отвечает за процесс коммуникации и диспетчерезации событий.



[Рис 2. Архитектура клиент приложения](file:///C:\Users\vin16085\AppData\Local\Temp\VWM.docx)

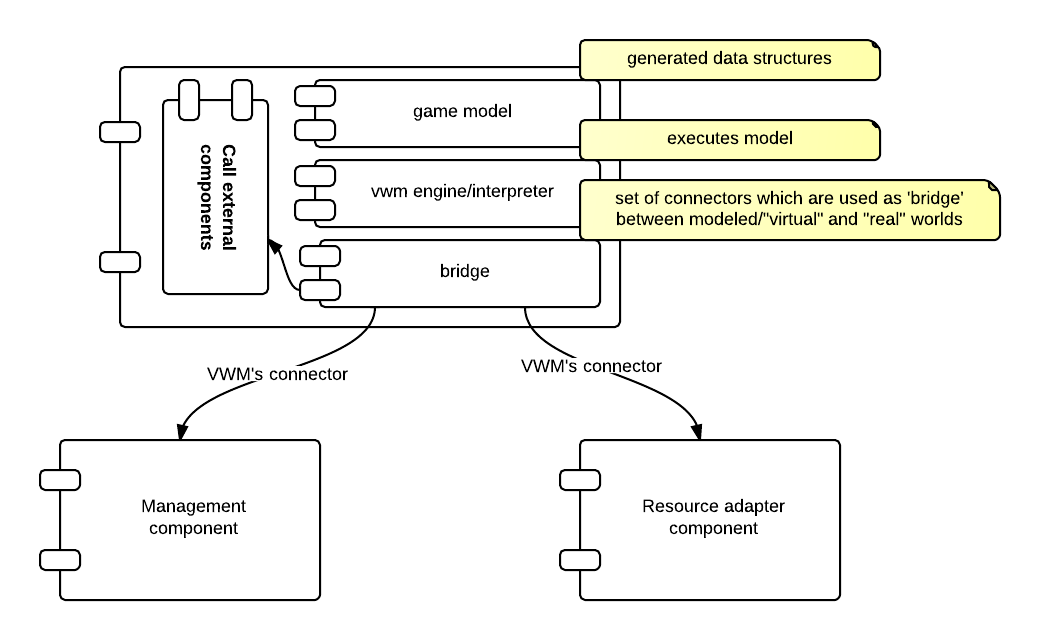
*На схеме рассмотрен случай ‘толстого клиента’. Логическое ядро игры (модель) расположено на клиенте.*

Модель описанная выше состоит в свою очередь из:

1. Game model - сгенерированные структуры данных
2. Vwm engine/interpreter – интерпретирует описание виртуального мира (изменяет виртуальный мир в соответствии с его VW моделью)
3. Bridge – коннекторы которые служат для интеграции с другими частями системы.
4. User Components – внедряемый код, написанный на языке высокого уровня в который компилируется VWML код.
5. View – интерфейсное отображение логических сущностей виртуального мира и происходящих в нем событий.
6. Controller – отвечает за процесс коммуникации и диспетчерезации событий в системе.

## Сервер-приложение

Основой серверного приложения является модель виртуального мира и связанные с ней инфраструктурные компоненты. Приведенная ниже диаграмма описывает связь модели, созданной с помощью VWM технологии, со стандартной серверной инфраструктурой.



[Рис 3. Архитектура сервер-приложений](file:///C:\Users\vin16085\AppData\Local\Temp\Рис%203.%20Архитектура%20сервер-приложений)

На диаграмме (см. рис 3) представлены основные компоненты сервер-приложения и связанные с ним компоненты инфраструктуры:

1. Game model - сгенерированные структуры данных
2. Vwm engine/interpreter – интерпретирует описание виртуального мира (изменяет виртуальный мир в соответствии с его VW моделью)
3. Bridge – коннекторы которые служат для интеграции с другими частями системы.
4. Call External Components – внедряемый код, написанный на языке высокого уровня в который компилируется VWML код.
5. Management component – обеспечивает конфигурирование и управление сервером
6. Resource adapter component – обеспечивает взаимодействие с клиент-приложением

Как видно из приведенных диаграмм, структуры клиентских и серверных приложений аналогичны. Это упрощает процесс отладки, как на стороне сервера, так и на стороне клиента.

И, кроме того, описания виртуального мира в случаях клиент-приложений и сервер-приложений эквивалентны.

# Процесс разработки

Разработка UI

Инфраструктурная интеграция

Моделирование на языке VWML, трансляция на язык высокого уровня

UI developing;

Net connections;

(Infrastructure integration)

Final debugging

Phase 3

**<< JAVA + HTML5 >>**

UI developing;

Net connections;

(Infrastructure integration)

Final debugging

Debugging Balancing

Compiling

VW modeling

Phase 2

Phase 3

**<< JAVA + HTML5 >>**

**<< JAVA ‘ready ’>>** >>

**<< VW ML project’>>** >>

**<< Project Requirement**

**Document (PRD) >>**

**<< JAVA >>**

Описание на естественном языке

Not formal describing

Phase 1

[Рис 4. Поэтапная разработка проекта](file:///C:\Users\ogibayev\projects\vw\doc\overview\Рис%204.%20Поэтапная%20разработка%20проекта)

Процесс создания игры разделяется на 3 фазы, смотри Рис 4.

## Описание игры на естественном языке (phase 1)

Речь идет об описании игры на естественном языке. Описание игры происходит по некоторой не жесткой схеме. Указывается тип игры, аналоги, сюжет, в чем состоит “психологическая интрига” игры, ее особенности, возможные событийные вариации, интерфейсные решения и т.п.

Результатом первой фазы является документ который мы называем PRD (Project Requirement Document).

## Описание игры на языке VWML (phase 2)

На этом этапе описывается игра на языке (VWML – Virtual World Modeling Language) . Отметим что данное описание не содержит описания граничного интерфейса, каких бы то ни было рисунков, анимаций. Можно сказать, что речь идет об описании модели игры “в чистом виде”. При этом поддерживается парадигма “двойного мира” - формального и воспринимаемого. Последнее означает что любое событие происходящее в виртуальном мире (игре) описывается как формально, так и с точки зрения восприятия (т.е. с точки зрения User Interface). Таким образом, имея формальное описание, мы знаем “точки подстановок” в коде обеспечивающего восприятия происходящих в виртуальном мире (в игре) событий.

Далее полученное описание на VWML игры конвертируется в язык высокого уровня (Java/ Java script). Следующим шагом происходит отладка кода игры и балансировка параметров модели игры.

*Дальнейшее развитие VWMT технологии предполагает проводить отладку VWML кода в специальной среде проектирования VWME (Virtual World Modeling Environment).*

После этой фазы мы имеем логическое ядро проекта в виде кода на языке высокого уровня готового к инфраструктурной интеграции.

## Интеграция и финальная отладка (phase 3)

На данном этапе происходит разработка графического интерфейса пользователя (GUI), а также разработка других интерфейсов.

Как упоминалось ранее, полученный код на языке высокого уровня (Java/Java script/C++) позволяет проводить инфраструктурную интеграцию в соответсвии с парадигмой “двойного мира”.

Кроме того, на этом этапе происходит интеграция с имеющейся инфраструктурой.

Заключительным этапом разработки является финальная отладка системы.

*Замечание:*

*К разработке проекта рекомендуется привлечь 2 команды: одна для Phase 1 и Phase 2, вторая - для Phase 3*