**Virtual World Modeling Technology (VWMT)**

Authors: M. Groozman

O. Gibaev

O. Tarandevich

Contents

[1. Introduction](#_Toc360621255) 3

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

[1.2 The architecture of the VWMT project](#_Toc360621257) 4

[2. Models of games in a client-server project architecture 6](#_Toc360621258)

[2.1 Client application](#_Toc360621259) 6

[2.2 Server application 7](#_Toc360621260)

[3. Development process](#_Toc360621261) 9

[3.1 Game description in original language (phase 1) 10](#_Toc360621262)

[3.2 Game description in VWML language (phase 2) 10](#_Toc360621263)

[3.3 Integration and final debugging (phase 3) 11](#_Toc360621264)

# Introduction

## VWM Technology. General description

We have proposed a language VWML (Virtual World Modeling Language). This language and the technology (VWM technology) based on it are tools oriented on the “creation” of virtual worlds, in particular - for the development of different types of games.

The language of VWML describes the entities of the world and the relationships between them. Besides, virtual worlds are “inhabited” by various entities, which can be both people (players, in particular) and virtual characters. The events taking place in the virtual world, are produced by its nature (the laws) and the behavior of the various entities in it.

We can say that VWML is focused on the description of virtual worlds, their evolution and the behavior of the entities, “inhabiting” it.

Foresaid allows us to call the programming on VWML the anthropomorphic programming.

The offered tool is based on the ideas of math logic, the theory of finite automata and functional approach to programming.

It’s worth mentioning that the projected world is a model of a certain formal theory and the realization of a common scheme.

By analyzing different implementations of the same scheme, various plot fillings and interface variations we are able to create “game families”.

Summarizing the foresaid, let’s define the conceptual provisions of VWM technology.

Paradigm 1

The virtual world is a set of entities or individuals which are in certain relationships with each other.

Paradigm 2

The virtual world is a model of a certain formal theory.

The axioms of this theory are the laws of the virtual world.

Paradigm 3 (evolutionary paradigm)

Any change in the virtual world is a change of the complex of individuals (entities) that “inhabit” this world or a change of the relationship between entities.

Paradigm 4 (paradigm of the [permanent](http://slovari.yandex.ua/permanent/en-ru/LingvoComputer/#lingvo/) laws)

World evolution does not change its laws.

The virtual world is a “variable” model of a certain formal theory.

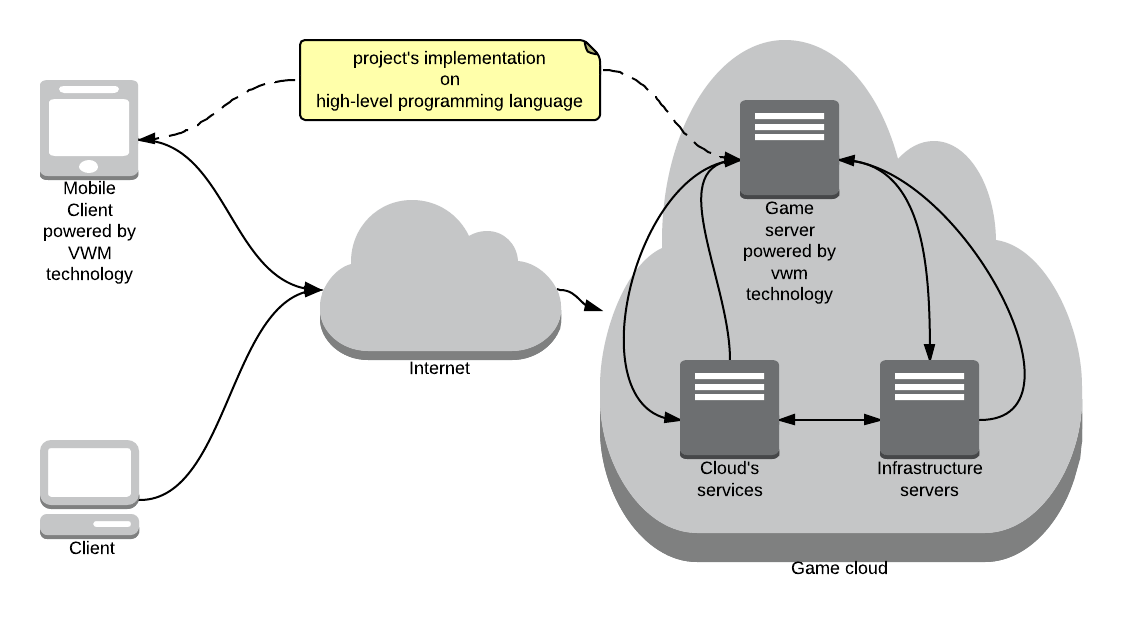
Paradigm 5 (paradigm of occurring events)

By modifying interpretation of the formal theory we are talking about the events taking place in the virtual world.

The events taking place in the virtual world are initiated by intelligent beings that are “inhabiting this world”.

## The architecture of the VWMT project

The system translates the world described by VWML in high-level languages, such as: java, java-script, c++; thus we are able to implement VWML projects on any platform (to keep the cross-platform factor).



[Image 1. Client-server architecture of VWMT project](file:///C:\Users\vin16085\AppData\Local\Temp\VWM.docx)

As the technology allows to generate VWML into the code in a high-level language, we can get the code for the core of the project, which can “run” in any operating environment, both on the client side and the server side (See Image 1.)

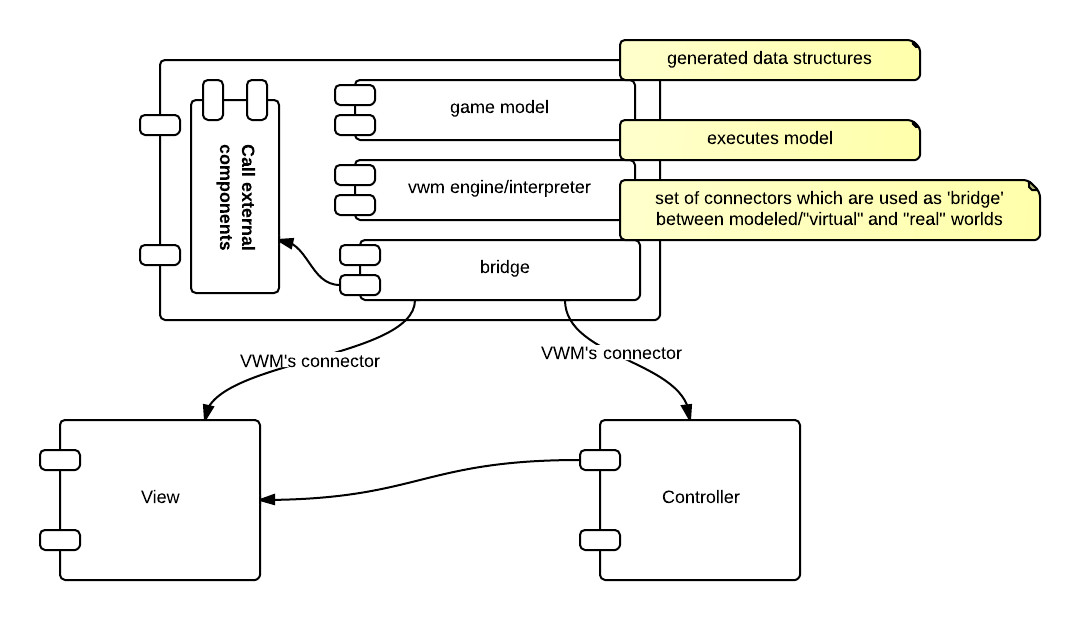
The system can be easily integrated with the existing platforms for developing games such as ODOBO.

# Models of games in a client-server architecture project

Let’s consider the process of development and integration for client applications and server applications.

## Client application

The composite pattern MVC (Model-View-Controller) is a classic pattern for the development of a client application; where Model defines a model (in this case the model of the virtual world, game, in particular), View is an user interface (UI) and Controller is responsible for the communication process and event dispatching.



[Image 2. Client application architecture](file:///C:\Users\vin16085\AppData\Local\Temp\VWM.docx)

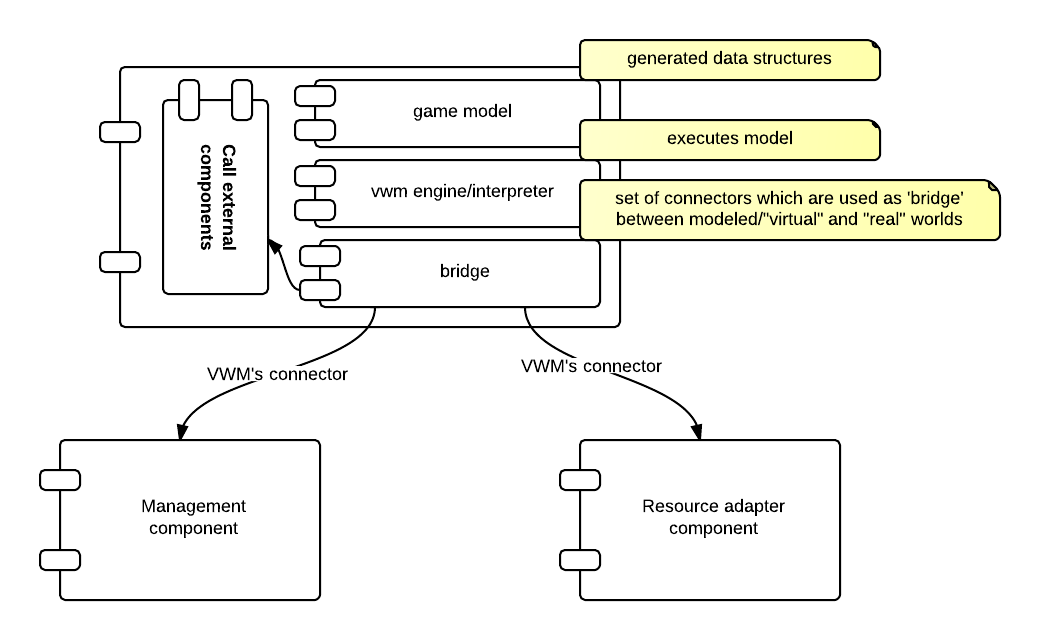
*The case of “a thick client” is shown on the scheme. The logical core of the game (model) is located on the client.*

The aforementioned model consists of:

1. Game model, which is a generated data structure.
2. VWM engine/interpreter interprets the description of the virtual world (changes the virtual world in accordance with its VW model).
3. Bridge, that consists of connectors which are used to integrate with other parts of the system.
4. User Components, which is an implemented code written in a high-level language in which the VWML code is compiled.
5. View, which is an interface mapping of logical entities of the virtual world and events occurring in it.
6. Controller, responsible for the process of communication and dispatching events in the system.

## Server application

A model of a virtual world and related infrastructure components are the basis of the server application. The scheme below describes the connection of the model generated with the help of VWM technology with the standard server infrastructure.



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

The main components of the server application and related infrastructure components are shown on the diagram (See Image 3):

1. Game model, which is a generated data structure.
2. Vwm engine/interpreter interprets the description of the virtual world (change the virtual world in accordance with its VW model).
3. Bridge, that consists of connectors which are used to integrate with other parts of the system.
4. Call External Components which is an implemented code written in a high-level language in which the VWML code is compiled.
5. Management component provides configuring and managing of the server
6. Resource adapter component provides interaction with the client application

It can be seen from the provided diagrams, that structures of client and server applications are similar. This simplifies the debugging process both on the server side and the client side.

And, moreover, the description of the virtual world in terms of client applications and server applications is equivalent.

# Development process

UI development

Infrastructure integration

Modeling in the VWML, translation on a high-level language

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 >>**

Description in the original language

Not formal describing

Phase 1

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

The creation process of the game is divided into 3 phases (See Image 4).

## Game description in natural language (phase 1)

This case is about game description in the original language. Game description proceeds according to the particular flexible scheme. The type of the game, its analogues, the plot, "psychological intrigue" of the game, its features and possible event variations, interface solutions, etc. will be also specified.

The result of the first phase is a document which is called PRD (Project Requirement Document).

## Game description in VWML language (phase 2)

On this stage the game is described in VWML (VWML – Virtual World Modeling Language). Let’s point out that this description contains neither the description of the boundary interface, nor images or animations. The task lies in describing the game model “in the raw”. Herewith the paradigm of “dual world” is supported, both formal and perceived. This means that any event occurring in the virtual world (game) is determined both formally and from the point of perception (i.e., from the viewpoint of User Interface). Thus, having a formal description, we know the “substitution points” in the code that provides the perception of occurring events in a virtual world (the game).

Later the obtained description of VWML game is converted into a high-level language (Java / Java script). The next step is to debug the game code and to balance the parameters of the game model.

*Further development of the VWMT technology involves debugging the VWML code in a special environment of engineering VWME (Virtual World Modeling Environment).*

After this phase, we get a logic core of the project in the form of a code in the high-level language ready for infrastructure integration.

## Integration and final debugging (phase 3)

At this stage, the development of graphic user interface (GUI) occurs, as well as the development of other interfaces.

As it was mentioned earlier, the received code in a high level language (Java / Java script / C ++) allows to perform the infrastructure integration in compliance with the paradigm of the "dual world".

Moreover, integration with existing infrastructure is taking place at this stage.

The final stage of development is final debugging of the system.

*Remark:*

*It is recommended to engage 2 teams into the development of the project: one for Phase 1 and Phase 2, the second - for Phase 3.*