Virtual World Modeling

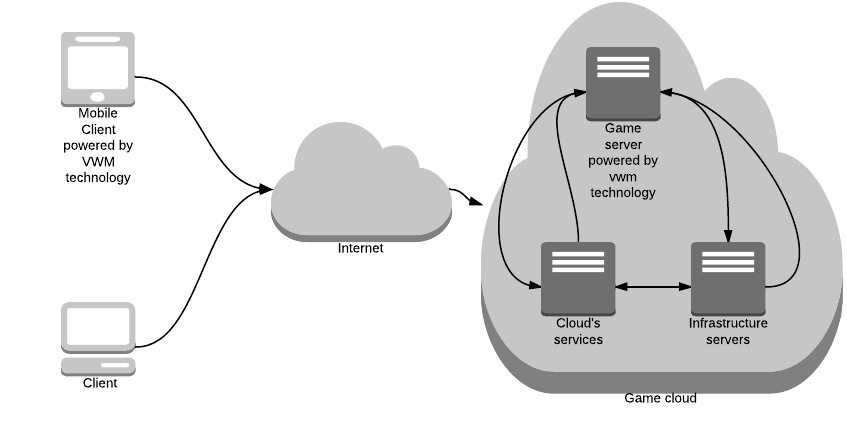
Technology

(Russian language)

# Введение

Ядром системы является язык ориентированный на описание эволюции виртуальных миров. Язык VWML описывает мир на языке сущностей этого мира, в частности поведение разумных существ связанных законами этого мира; важно заметить что мир эволюционирует в соответствии с правилами определенными в описании мира и намерениями существ. Данный язык использует идею современной математической логики и теории конечных автоматов; таким образом система сочетающая в себе интеграцию реального и виртуального мира органично подходит для проектирования и создания игровых миров, в частности для мира азартных игр.

Система транслирует мир описанный языком VWML в языки высокого уровня, такие как: java, java-script, c++; сохраняя определенный уровень абстракции, тем самым позволяя сохранить фактор кросс-платформенности. Одна из важных функциональностей системы является возможность изменения поведения мира run-time, посредством изменения законов мира или поведения сущностей населяющих этот мир. Система легко интегрируется с различными платформами для разработки игр.

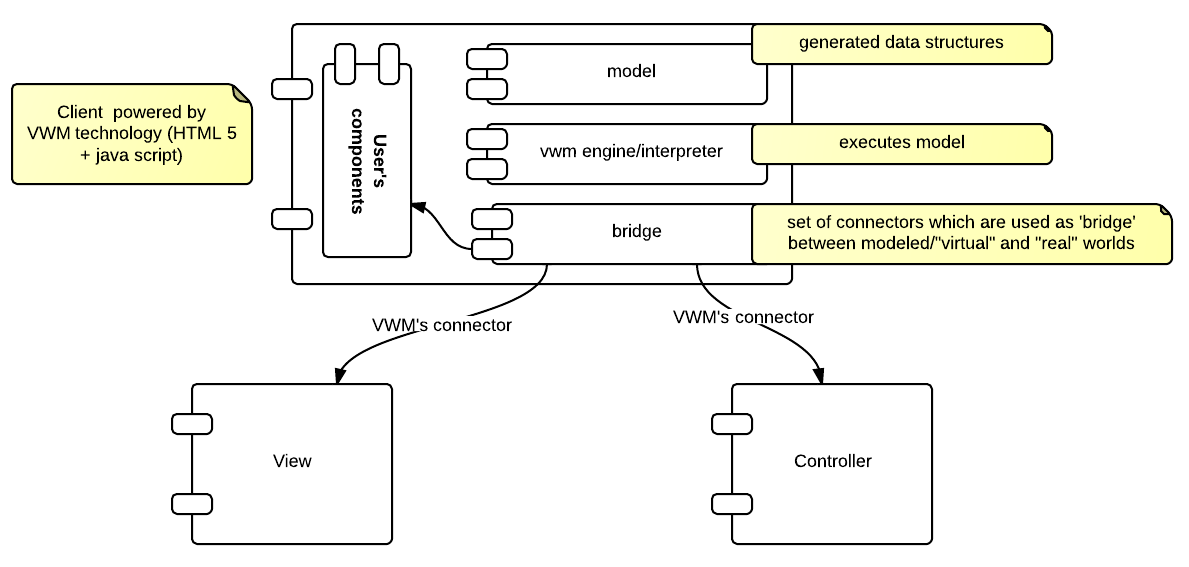


## Общая архитектура ‘клиент-сервер’ моделей сгенерированных посредством VWML.

Технология основанная на языке VWML позволяет описывать модели поведения как для клиентных так и для серверных приложений. Ниже мы рассмотрим процесс разработки и интеграции для обоих типов 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 или игры) или бизнес логику, View – пользовательский интерфейс (UI) и Controller отвечает за процесс коммуникации и диспетчерезации событий.

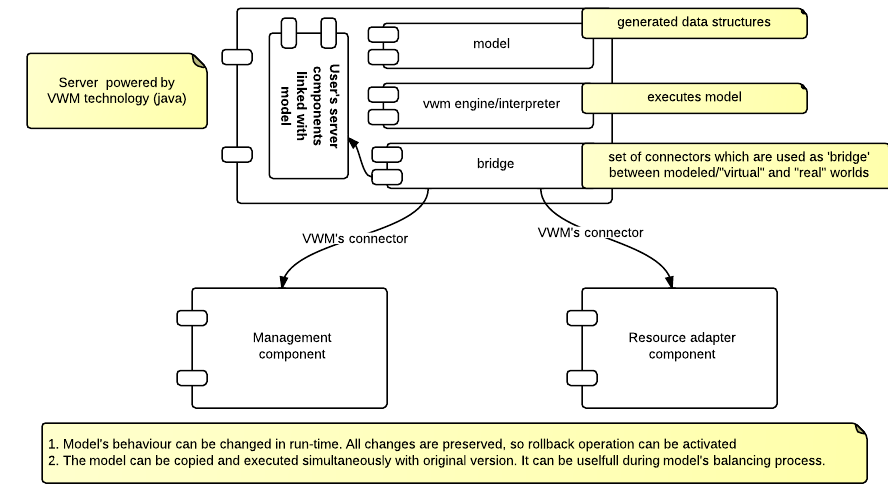


Model описанная выше состоит в свою очередь из:

1. Game model - сгенерированные структуры данных (динамически создаются VWML процессором)
2. Vwm engine/interpreter - выполняет операции над Game model согласно правилам которые определяются в процессе описания модели; точкой входа являются так называемые ‘entity life term’.
3. Bridge – коннекторы которые служат для интеграции с другими частями системы.

## Сервер приложение

Основой серверного приложения является его бизнес логика или модель и связанные с ней инфраструктурные компоненты. Приведенная ниже диаграмма описывает связь модели, созданной с помощью VWM инструментов и стандартной серверной инфраструктуры.



Model описанная выше состоит в свою очередь из:

1. Game model - сгенерированные структуры данных (динамически создаются VWML процессором)
2. Vwm engine/interpreter - выполняет операции над Game model согласно правилам которые определяются в процессе описания модели; точкой входа являются так называемые ‘entity life term’.

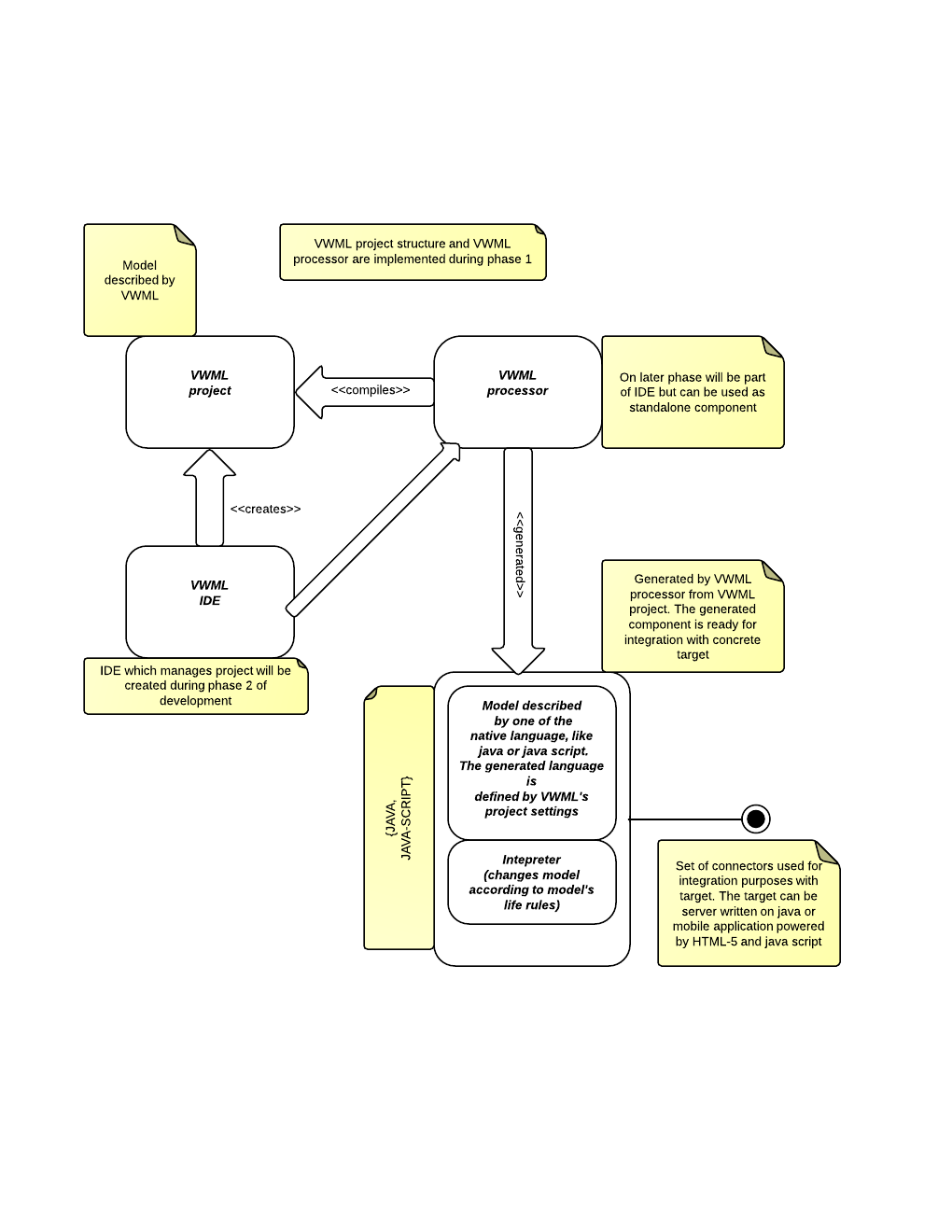
Bridge – коннекторы которые служат для интеграции с другими частями системы.

Как видно из приведенных диаграмм для ‘клиентного’ и ’серверного’ приложений структура модели остается неизменной. Это в свою очередь сильно упрощает процесс отладки, балансировки и связи между моделями. В случае использования IDE эти процессы могут быть выполнены в контексте самой IDE без использования ‘mockup’ компонент.

## Технология и процесс разработки.

## Компоненты

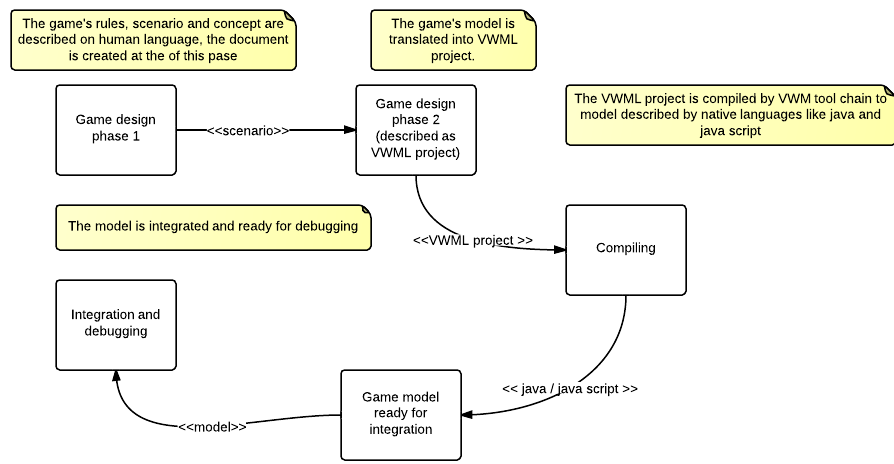
Следующая диаграмма описывает компоненты принимающие участие в процессе генерации модели.

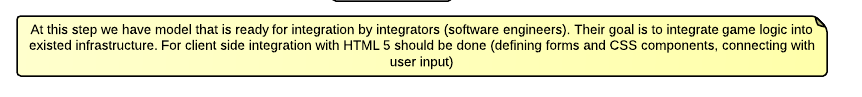


1. VWML project – множество vwml файлов описывающих модель.
2. VWML processor – компилирует VWML project в проект на одном из языков высокого уровня таких как java или java script.
3. VWML IDE – среда для разработки и моделирования, планируется разработка на второй фазе проектирования системы.

## Процесс разработки

Следующая диаграмма описывает шаги разработки игр с помощью VWM технологи.





## Группы разработчиков