

Domain Specific Languages

Теория и Практика

CUSTIS

Всеволод Коньков

Обо мне любимом...

Ключевые слова:

- Всеволод
- Коньков
- Мехмат
- Microsoft .Net
- C#
- SCRUM-мастер
- Быдлокодер
- Команда VIEW
- Чистый код
- Паттерны поведения проектирования
- Рефакторинг
- TDD
- Люблю женщин и выпить математику
- Отец
- Холостяк
- Чувство юмора наиверное
- Принцип максимума
- Исследование операций





ПОЕХАЛИ

Что такое DSL?



DOMAIN
SPECIFIC
LANGUAGE

Примеры DSL

- [Excel](#)
- [SQL](#)
- [Graphviz](#)
- [FIT](#)
- [Make](#)
- [CSS](#)
- [SVG](#)
- XAML
- [HTML](#)



Признаки DSL

- Язык программирования
- Ограниченная выразительность
- Фокус на предметной области
- Похож на естественный язык



Domain Specific Language

Специализированный язык предметной области – это язык программирования с ограниченной выразительностью конкретной предметной области



Язык программирования



По Кауфману

- Инструмент для планирования поведения исполнителя
- Знаковая система для планирования поведения компьютеров

Предметная область (domain)

область объектов, универсум рассуждения, универсум рассмотрения, или просто универсум, класс (множество) объектов, рассматриваемых в пределах данного контекста. Под контекстом здесь может пониматься отдельное рассуждение или выражающая его фраза, или совокупность фраз, фрагмент научной теории или теория в целом.



БСЭ

DSL – А зачем?



DSL: Пример из недалекого будущего



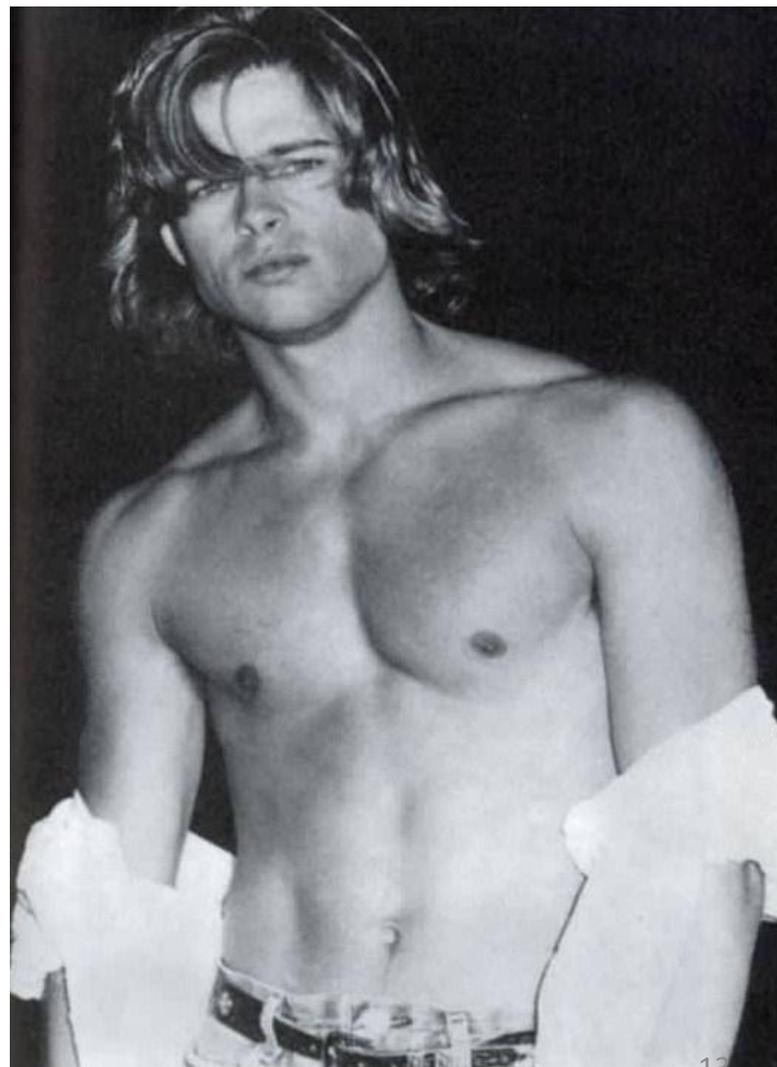
Кухонный робот



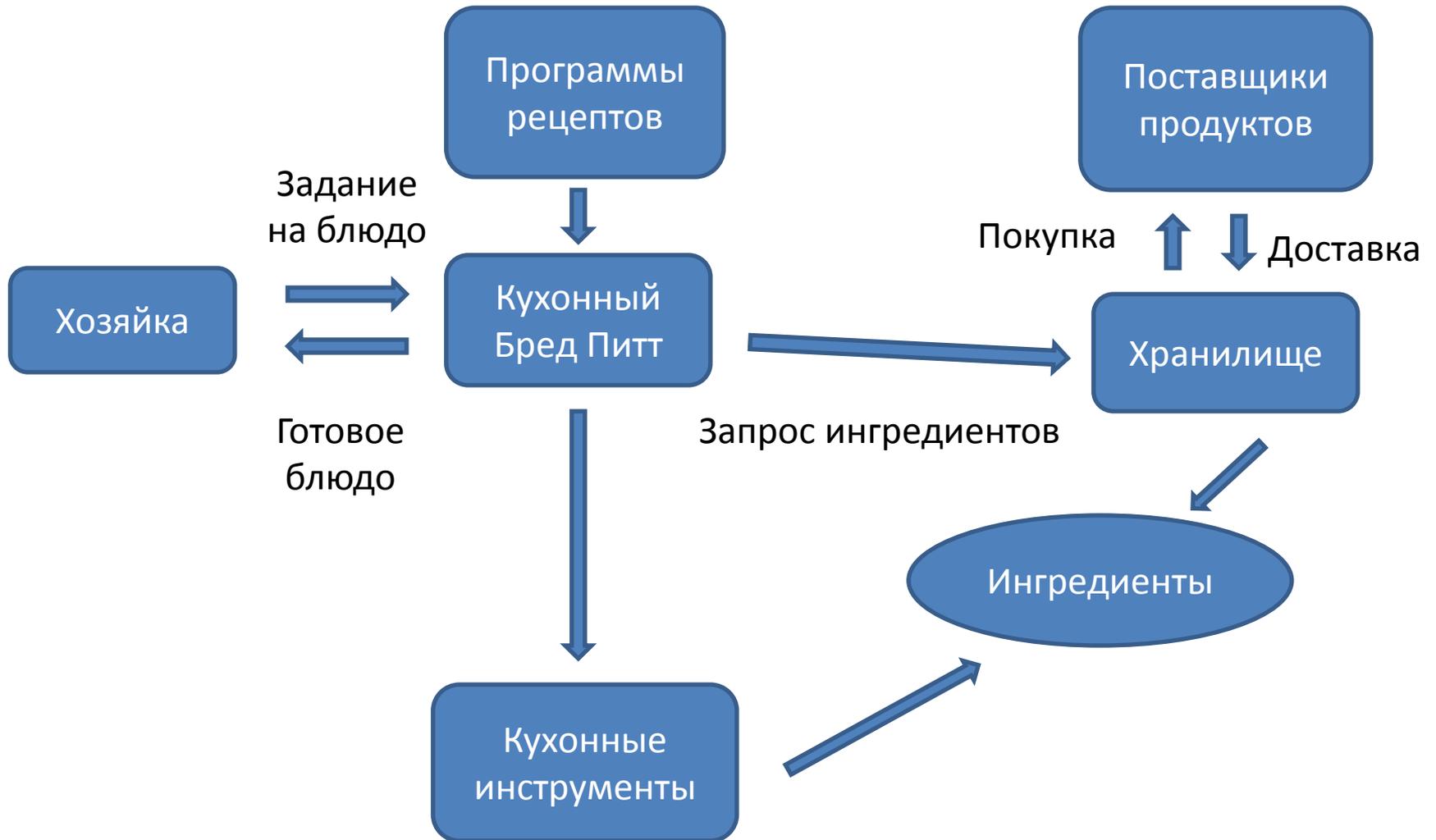
Кухонный Бред Питт



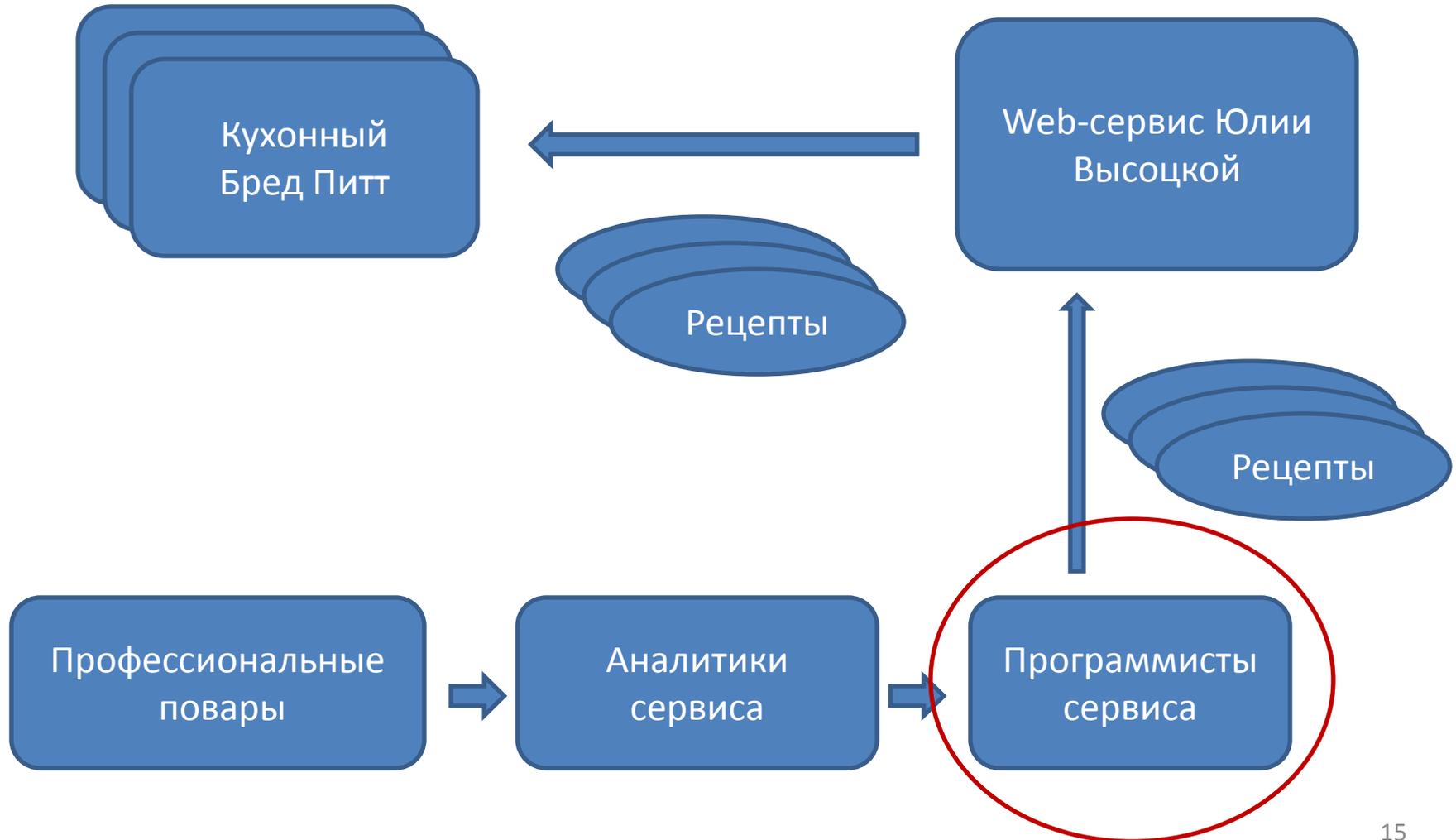
+



Как он работает?



Обновление БД рецептов



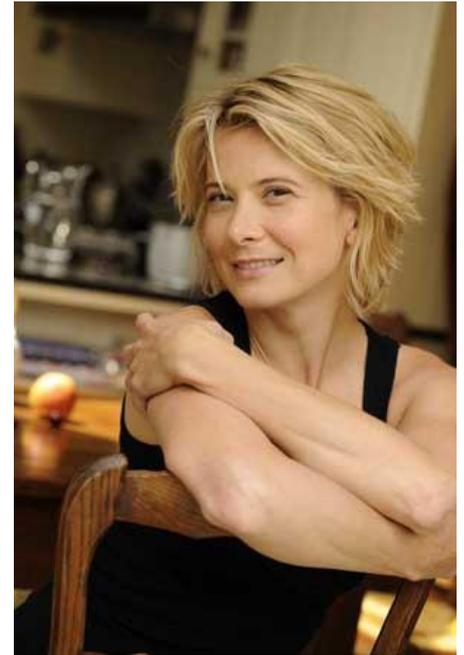
Пример рецепта: Тыквенный суп с карри

Ингредиенты:

- 1 кг тыквы
- 1 большая луковица
- 3 столовые ложки сливочного масла
- 1 столовая ложка оливкового масла
- 1 ½ литра овощного бульона
- 100 г. Сливки
- 1 ч.л. Карри
- горсть мелкорубленной петрушки
- морская соль

Приготовление:

1. Разогреть большую кастрюлю, добавить туда сливочное и оливковое масло
2. Лук крупно нарезать и обжарить до прозрачности
3. Тыкву нарезать крупными кусками и снять кожуру
4. Отправить тыкву к луку, тушить около минуты, помешивая, затем залить кипятком или овощным бульоном и варить до готовности тыквы, 25-30 минут
5. Готовую тыкву взбить в блендере, добавив немного бульона
6. Добавить сливки, посолить по вкусу, прогреть слегка



Хардкодинг на C#

```
public interface IRecipe
```

```
{
```

```
    void Cook(IKitchenBotController bot);
```

```
}
```

```
public interface IKitchenBotController
```

```
{
```

```
    void ExecuteAction(int actionCode, params object[] args);
```

```
    int StateCode { get; }
```

```
    event EventHandler<KitchenBotEventArgs> Events;
```

```
}
```



Пример в студию



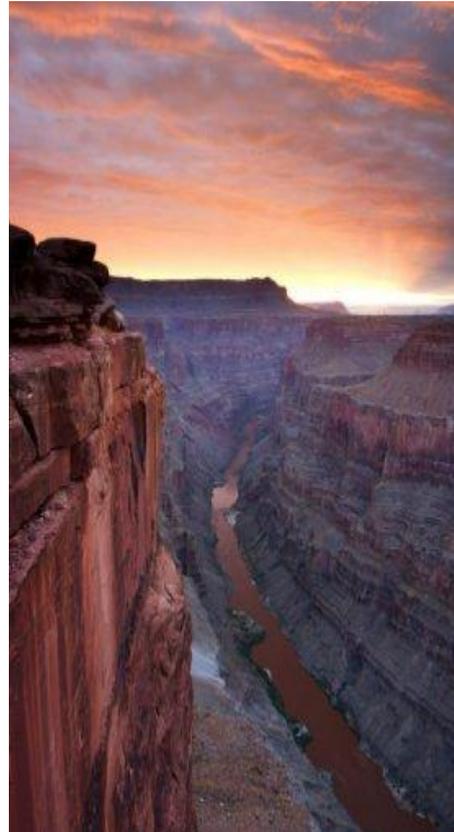
Семантический разрыв

Ингредиенты:

- 1 кг тыквы
- 1 большая луковица
- 3 столовые ложки сливочного масла
- 1 столовая ложка оливкового масла
- 1 ½ литра овощного бульона
- 100 г. Сливки
- 1 ч.л. Карри
- горсть мелкорубленной петрушки
- морская соль

Приготовление:

1. Разогреть большую кастрюлю, добавить туда сливочное и оливковое масло
2. Лук крупно нарезать и обжарить до прозрачности
3. Тыкву нарезать крупными кусками и снять кожуру
4. Отправить тыкву к луку, тушить около минуты, помешивая, затем залить кипятком или овощным бульоном и варить до готовности тыквы, 25-30 минут
5. Готовую тыкву взбить в блендере, добавив немного бульона
6. Добавить сливки, посолить по вкусу, прогреть слегка



```
bot.ExecuteAction(2025);

bot.ExecuteAction(28, "Морская соль", 1100, 1);
if (bot.StateCode == -1)
{
    bot.ExecuteAction(1028, "Морская соль", 1100, 1);
    _waitEvent.Wait();
}
bot.ExecuteAction(1012, seaSoltCode);

bot.ExecuteAction(2087);
bot.ExecuteAction(2088, butterCode);
bot.ExecuteAction(2088, oliveOilCode);
bot.ExecuteAction(2089, 60, 1);
_state = 1;
_waitEvent.Wait();

bot.ExecuteAction(2075, onionCode);
bot.ExecuteAction(2089, onionCode);
_state = 3;
_waitEvent.Wait();

bot.ExecuteAction(2075, pumkinCode);
bot.ExecuteAction(2088, pumkinCode);
bot.ExecuteAction(2088, creamCode);
bot.ExecuteAction(2091, "Бульон");
_state = 4;
bot.ExecuteAction(2089, 35, 1);

bot.ExecuteAction(2093);
bot.ExecuteAction(2088, creamCode, 1);
bot.ExecuteAction(2087, seaSoltCode, 2);
bot.ExecuteAction(2089, 1, 1);
```

Что можно сделать с этим разрывом?



- Высокоуровневое API
- Модель данных (рецептов)

Пример на C# в студию



Что это было?

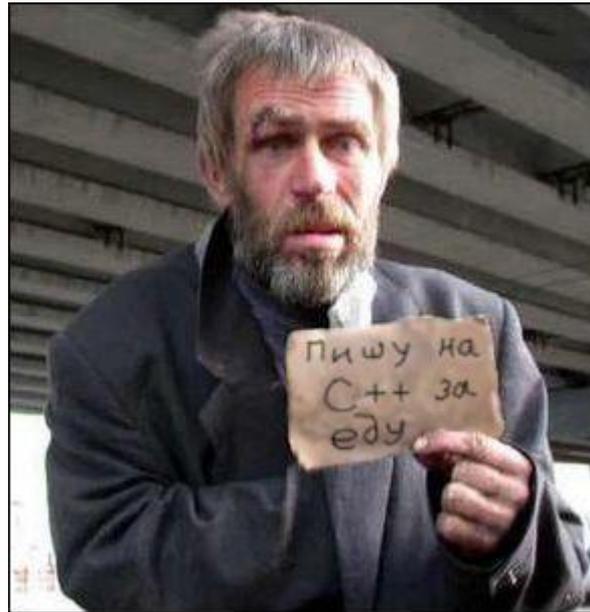


Это был внутренний DSL на C#

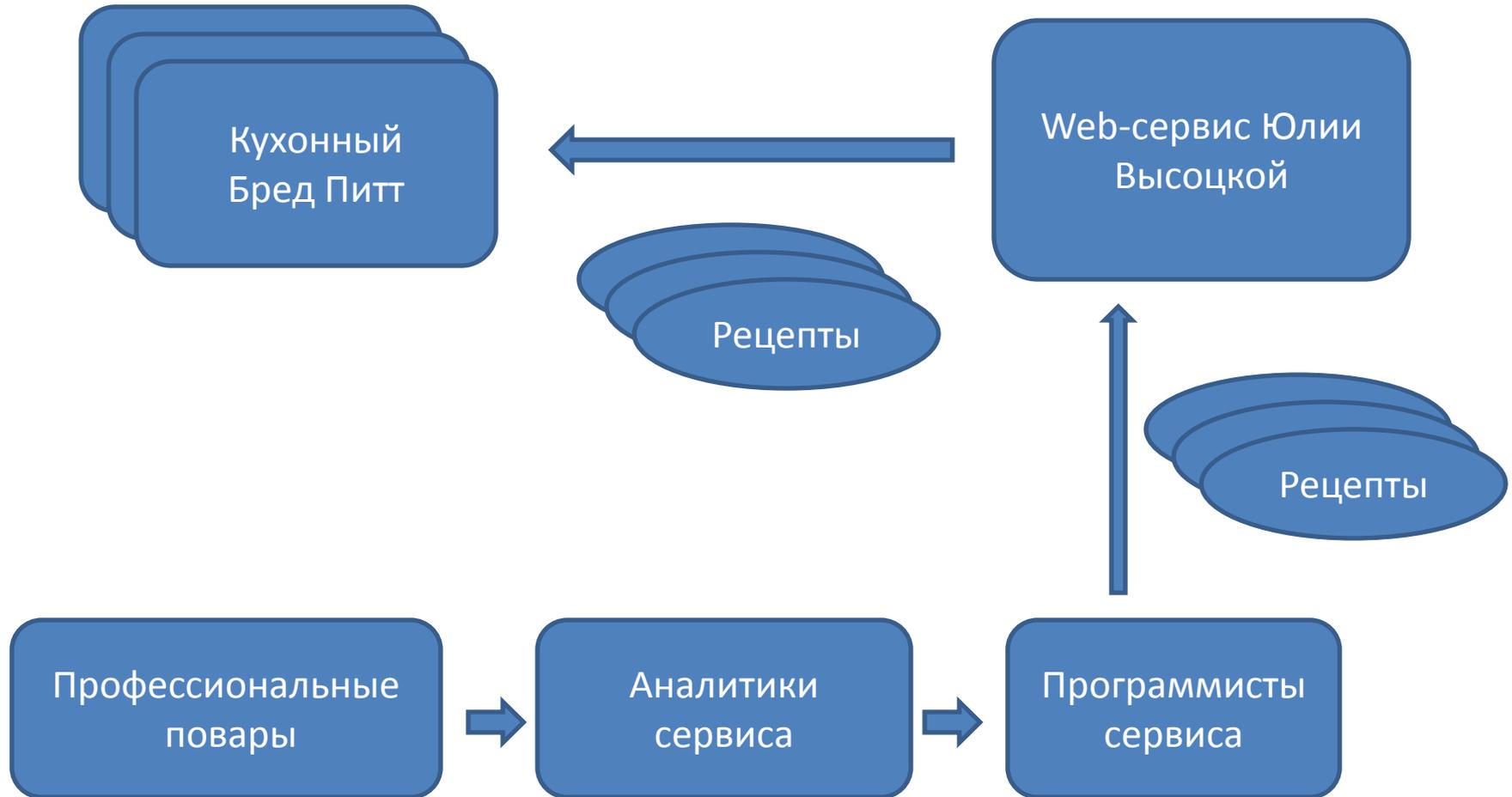


Пользователи DSL

- Программисты



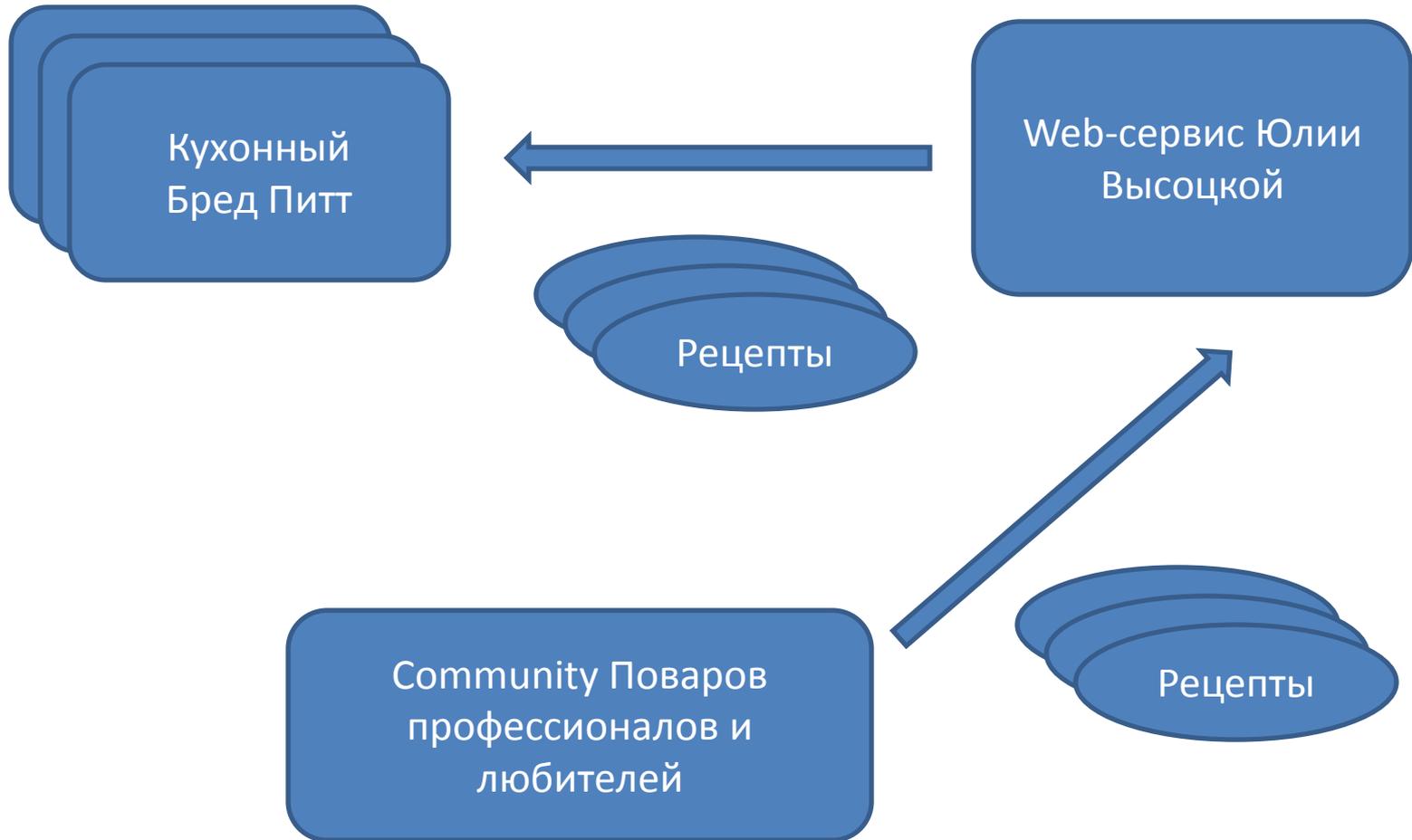
Обновление БД рецептов



Кто должен программировать рецепты?



Обновление БД рецептов (в идеале)

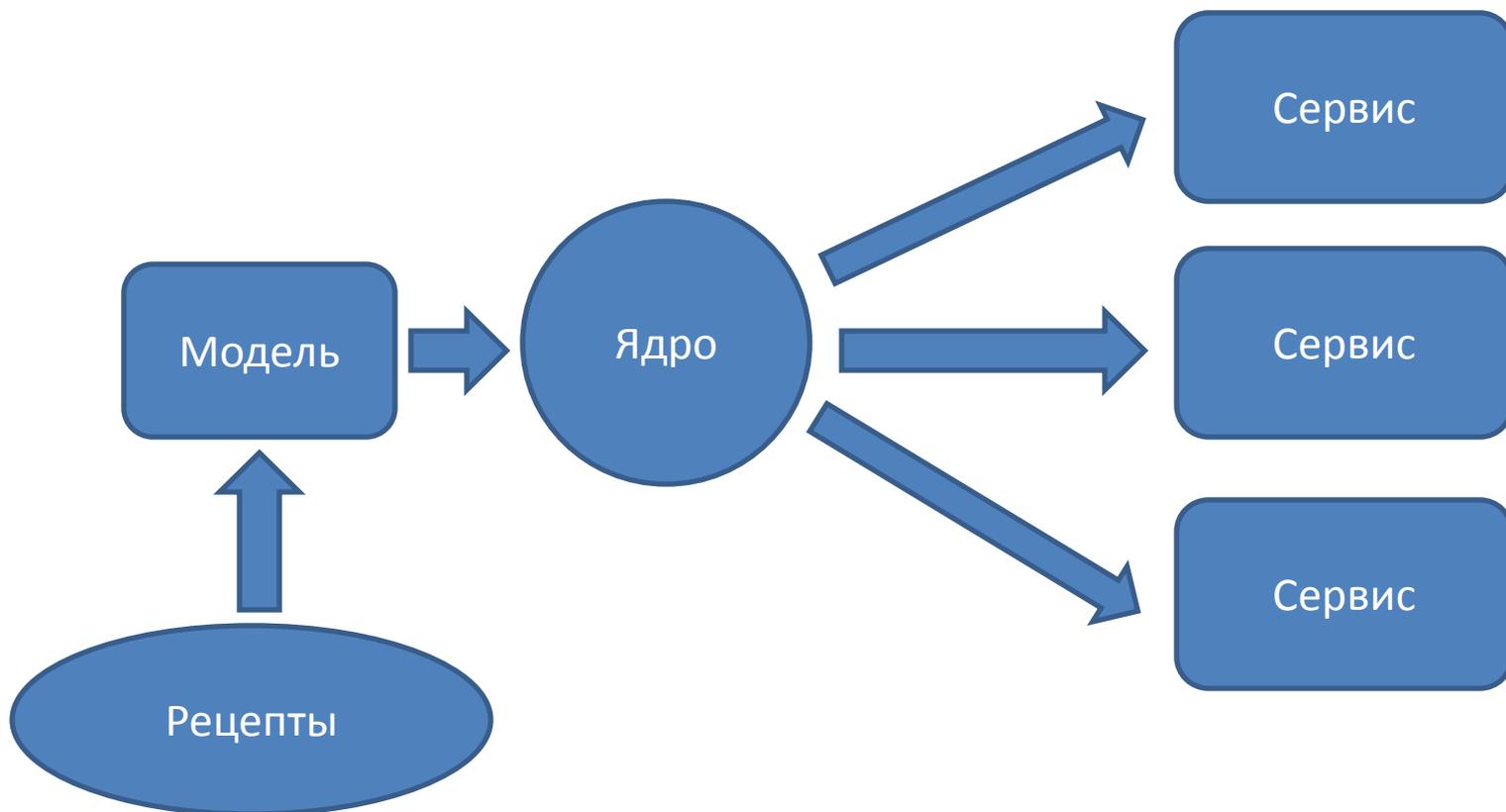


Почему обновление Бреда Питта ВОЗМОЖНО?

У процесса приготовления есть:

- Возможность отделения общих инфраструктурных сервисов.
- Возможность создания общей модели процесса приготовления, следовательно обобщенного алгоритма приготовления блюда (ядра).
- Возможность параметризации ингредиентов и действий приготовления значениями атрибутов, которые будут интерпретироваться ядром

Составляющие процесса приготовления





Рецепт – это данные
модели приготовления

Какой формат должны иметь данные рецепта?

- Бинарный
- XML
- Текстовый

Как читать и редактировать рецепт?

- С помощью визуального средства
- С помощью текстового редактора



Язык описания рецептов – это DSL

Вне зависимости от форматов и средств редактирования

Пользователи DSL

- Программисты
- Системные аналитики (инженеры)
- Бизнес-аналитики
- Конечные пользователи

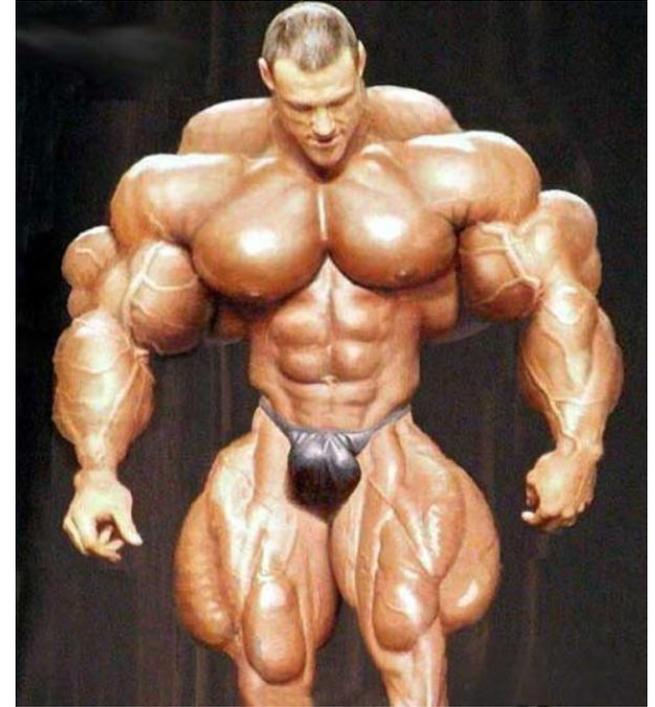


Классификация DSL

- Внутренний DSL (C#, Ruby, Groovy, Python, Boo)
- Внешний DSL (sql, excel, html)
- Языковые инструменты (Language Workbenches)

Language Workbenches

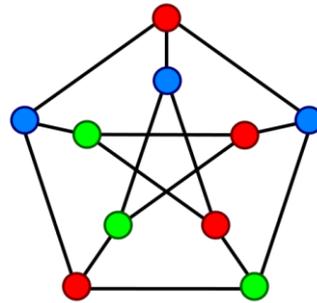
- Intentional Workbench
- Meta Edit
- Meta-Programming System
- Xtext
- Microsoft Visualization & Modeling SDK
- Microsoft SQL Modeling Server (Oslo)



Другие классификации DSL

- Текстовые vs Графические vs ...
- Компилируемые vs Интерпретируемые
- Императивные vs Декларативные

Альтернативная вычислительная модель



- Адаптивная модель (Adaptive Model)
- Таблица решений (Decision Table)
- Сеть зависимостей (Dependency Network)
- Системы вывода правил (Production Rule System)
- Конечный автомат (State Machine)

Новые парадигмы программирования

- Инженерия предметной области
- Model Driven Development (Architecture)
- Domain Driven Design
- Языково-ориентированное
программирование

Инструменты для создания DSL

- Средства языков
- Среды выполнения
- Библиотеки
- Генераторы парсеров
- Language Workbenches

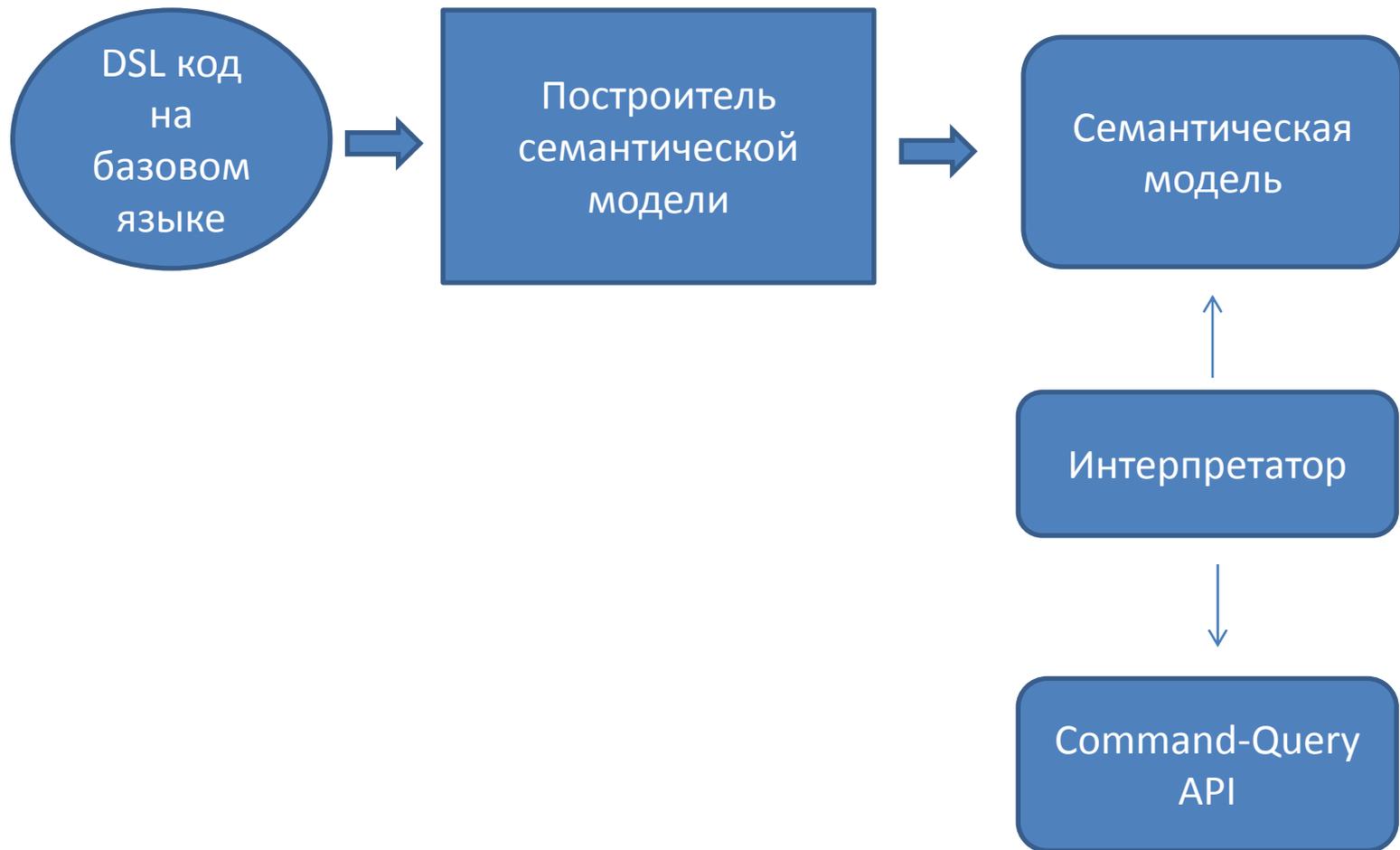


Внутренние DSL

Составляющие Внутренних DSL

- Command-query API
- Семантическая модель
- Построитель семантической модели

Внутренний DSL : Реализация



Внутренние DSL : Инструменты

- Языковые средства
- Среды выполнения
- Библиотеки



Приемы по созданию внутреннего DSL

- Expression Builder
- Function Sequence
 - Nested Function
- Method Chaining
- Object Scoping
 - Closure
 - Nested Closure
 - Literal List
 - Literal Map
- Dynamic Reception
 - Annotation
 - Parse Tree Manipulation
- Class Symbol Table
 - Textual Polishing
- Literal Extension



Пример на Ruby в студию

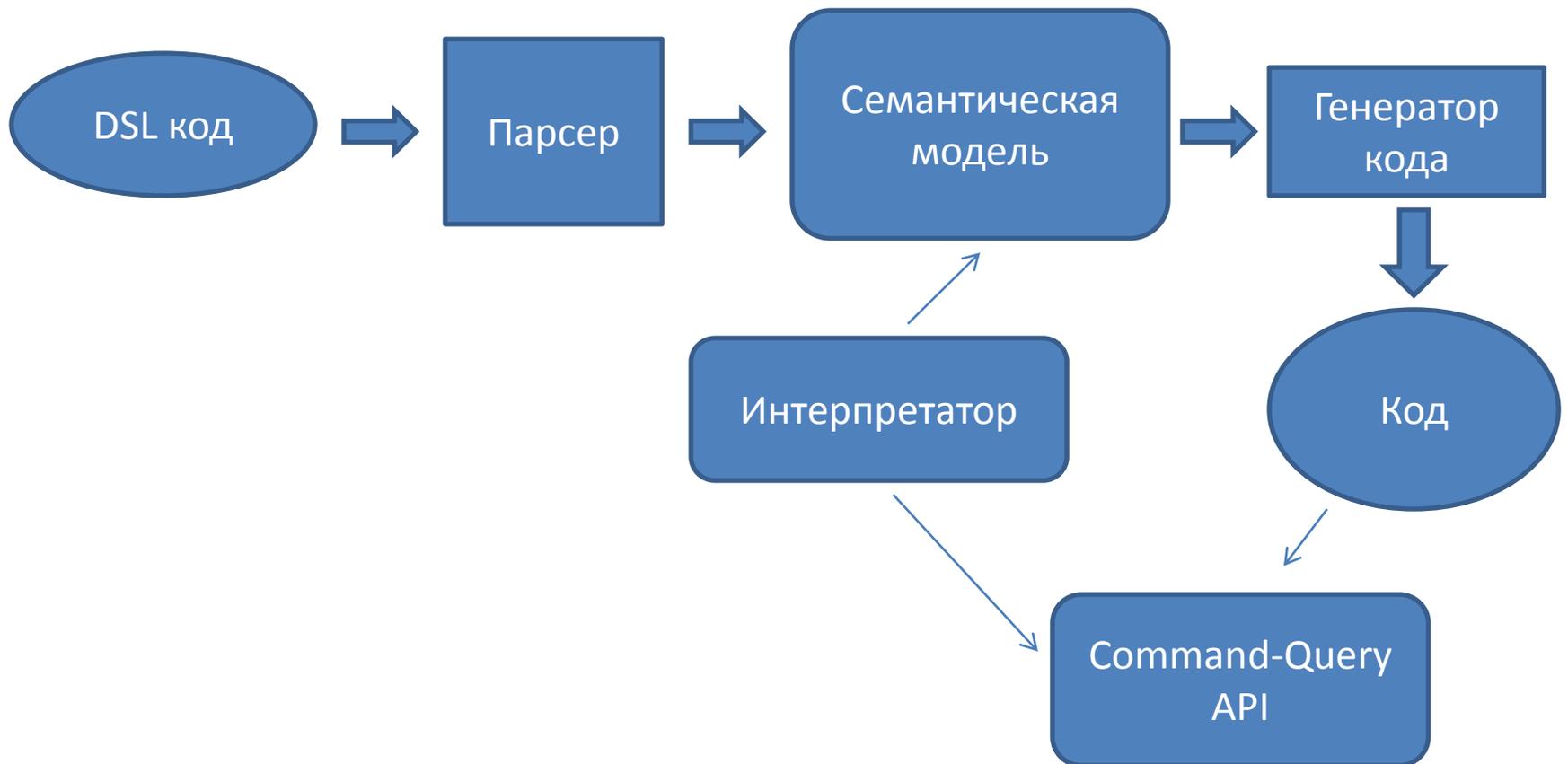


Внешние DSL

Составляющие Внешних DSL

- Command-query API
- Семантическая модель
- Построитель модели или генератор кода
- Парсер языка

Внешний DSL : реализация



Внешние DSL

- Среды выполнения
- Библиотеки
- Генераторы парсеров
- Language Workbenches



Внешние DSL как языки разметки

- HTML
- SVG
- XSL-FO
- RML
- Огромное количество конфигурационных файлов

```
<business-operation code="opt_ship_ware" name="Отгрузка товара (ОПТ)">
  <account-operation module-code="OPT" operation-code="opt_op_ware_shipped" categories="inout" />
  <account-operation module-code="OPT" operation-code="opt_op_sell_cancel" categories="unrequired"/>
  <account-operation module-code="OPT" operation-code="opt_op_ware_shipped_underload"
categories="unrequired"/>
  <account-operation module-code="OPT" operation-code="opt_op_change_ship_str"
categories="unrequired"/>
  <account-operation module-code="OPT" operation-code="opt_op_sell_reject" categories="unrequired"/>
</business-operation>

<business-operation code="opt_op_sell_noreserv" name="Продажа без резерва (ОПТ)">
  <account-operation module-code="OPT" operation-code="opt_op_sell_noreserv" categories="inout"/>
  <account-operation module-code="OPT" operation-code="opt_op_sell_cancel" categories="unrequired"/>
  <account-operation module-code="OPT" operation-code="opt_op_ware_shipped_underload"
categories="unrequired"/>
  <account-operation module-code="OPT" operation-code="opt_op_sell_reject" categories="unrequired"/>
</business-operation>
```

Причины популярности XML

- Наличие стандартных средств разбора XML
- Бесплатные редакторы XML с валидацией по схеме
- Наличие средств XSLT – преобразование XML в другие форматы

Пример на XML в студию



Пример XSLT-генерации в студию



Внешние DSL: Генераторы парсеров

- Yacc\Lexx
- Bison
- Coco\R
- ANTLR



ANTLR - ANother Tool for Language Recognition

- Язык описания грамматики
- Собственная среда выполнения
- Средство разработки грамматики - ANTLRWorks
- Поддержка нескольких целевых языков: Java, JavaScript, C#, C\C++, Ruby, Python

Пример на ANTLR в студию



Language Workbenches



Language Workbenches: Составляющие

- Схема семантической модели
- Среда редактирования DSL (схемы, грамматики и т.д.)
- Поведение семантической модели



Language Workbenches: Примеры

- Jet Brains Meta-Programming System (MPS)
- Microsoft Visualization & Modeling SDK
- Microsoft SQL Modeling Server (Oslo)

Jet Brains MPS : вехи

- Проект стартовал в 2003
- В 2004 вышла статья Сергея Дмитриева «Языково-ориентированное программирование»
- С 2005 в open source под Apache 2.0
- С 2006 часть продуктов Jet Brains перешло на MPS

Jet Brains MPS : факты

- Язык – Java
- Языки определения языков:
 - Язык структуры
 - Язык редактора
 - Язык ограничений
 - Язык системы типов
 - Язык генератора
- Базовый язык для DSL

Пример на MPS в студию



Расчет потребления пива



Microsoft Visualization & Modeling SDK : вехи

- Известен с Visual Studio 2005 как DSL Tools, в составе Visual Studio SDK
- С Visual Studio 2008 выделен в отдельный пакет DSL SDK
- С Visual Studio 2010 переименован

Microsoft Visualization & Modeling SDK : факты

- Средство создания визуального DSL
- Противопоставлялся UML
- Составляющие:
 - Описание модели предметной области
 - Описание презентации
 - Сериализация модели
 - Генерация с помощью T4

Пример на VMSDK в студию



Что такое проект Oslo?

Проект Oslo – это набор инструментов для поддержки разработки программного обеспечения управляемой моделями (model-driven development).

Проект Oslo : вехи

- 2007 Анонс
- Сентябрь 2008. Исключение из проекта части, связанной с обновлением технологий обмена сообщений и потоков работ в продуктах Microsoft таких как BizTalk Server.
- Октябрь 2008. Первый СТР.
- Ноябрь 2009. Последний СТР. Присвоение проекту товарного имени **SQL Modeling Server**
- Апрель 2010. Обновление СТР под Visual Studio 2010.
- Июль 2010. Закрытие Quadrant.
- Август 2010. Уход Дугласа Парди (одного из идеологов Oslo и ведущих разработчиков проекта) в Facebook.
- Сентябрь 2010. Заявление Дона Бокса о закрытии существенном изменении SQL Modeling Server.

Что такое MDD?



Модель

Система абстракций, которая описывает избранные аспекты предметной области и может использоваться для решения задач, относящихся к этой области.



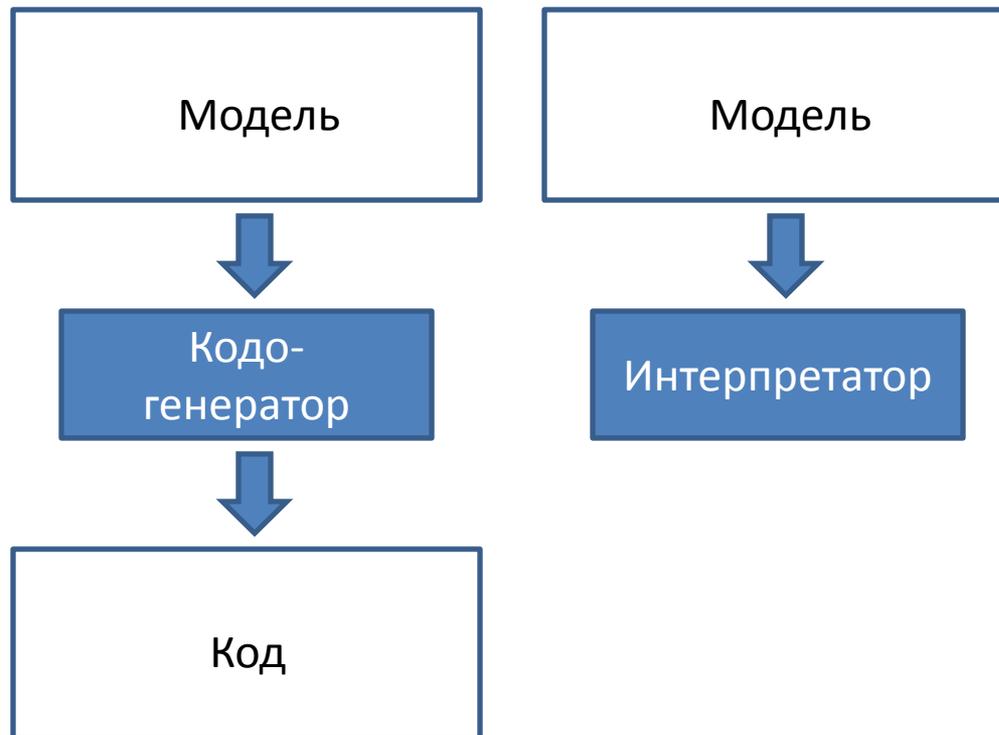
Что такое MDD?

- Разработка, управляемая моделями – это разработка программного обеспечения, в центр которого ставится разработка модели предметной области, с которой работает создаваемое программное обеспечение.
- Построенная модель является главным артефактом разрабатываемого программного обеспечения, на основании которой создается исполняемый код.



Что происходит с кодом в MDD?

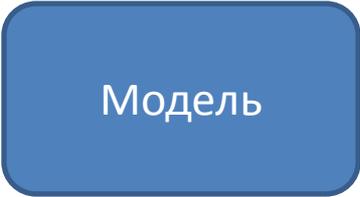
- Код может генерироваться
- Код может интерпретироваться



Роль инструментов Oslo в MDD

Упрощение процессов:

- Разработки модели
- Редактирования модели
- Развертывания модели
- Хранения модели



Модель

Составляющие Oslo

Язык M

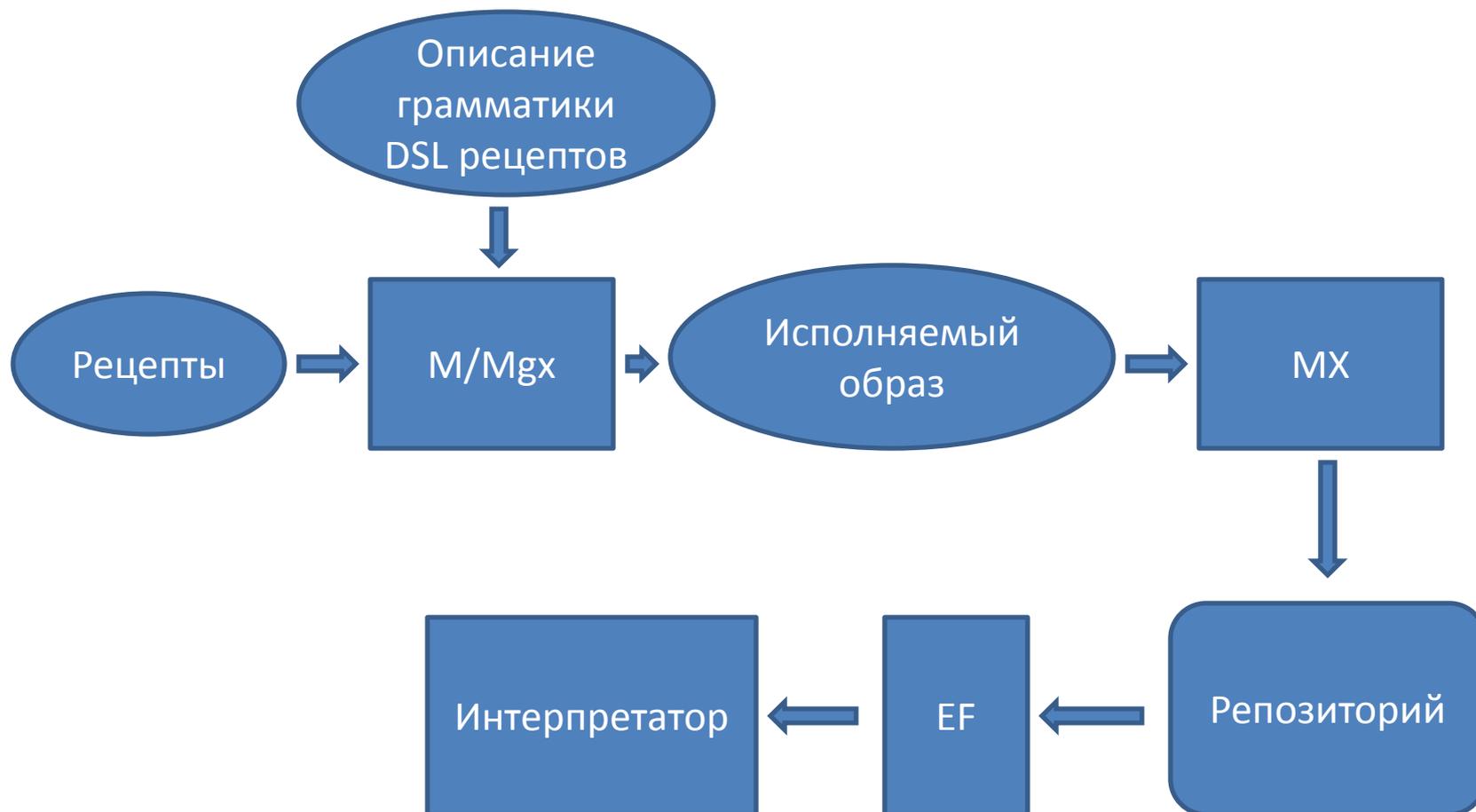
Quadrant

Репозиторий

+

Всякая полезная
хрень

Общая картина работы



Пример на Oslo в студию



*Лично для меня Oslo –
первый шаг к реализации
моего стремления сделать
всех программистами
(даже если они не будут об
этом знать)*

Дуглас Парди



Преимущества использования DSL

- Высокая скорость разработки
- Коммуникация с экспертами предметной области
- Легкое изменение контекста выполнения
- Альтернативная вычислительная модель
- Поддержка DDD

Проблемы с DSL

- Языковая какофония
- Высокая цена разработки
- Языковое гетто
- Ограниченная абстракция
- Версионность

Domain Driven Design & DSL

Единый язык (ubiquitous language)

Разговорный язык,
структурированный в
соответствии с моделью
предметной области и
используемый всеми
разработчиками для
координации работ группы
над программным проектом.



DSL vs UL

- **UL** – язык общения между бизнесом и разработчиками
- Программы на **DSL** чрезвычайно выразительны и находятся в прямой связи с **UL** (единым языком).
- **DSL** – формальный язык общения бизнеса и разработчиков

Жизненные примеры

- Вычислительные аналитические таблицы (XML + VBScript)
- Топология установок на нефтеперерабатывающем производстве (Внешний DSL)
- Логика генерации ресурсов web-сайта (XML)
- План счетов учетной машины (XML, XSLT)
- [Кубы данных \(XML + SQL\)](#)
- [Шаблоны операций товародвижения \(XML, XSLT\)](#)
- Шаблоны учетных регистров (Внешний DSL для конечного пользователя)
- Аналитические отчеты (Внешний DSL)



Спасибо за Внимание