Ноотропы RDF для BigData

PETER-SERVICE

Леонид Юрьев
Петер-Сервис R&D, Сколково

Sk Skolkovo

Sk IT

DevConf

http://www.devconf.ru

СОДЕРЖАНИЕ ДОКЛАДА

≈ 30 слайдов

1. Иллюстрация проблем на трех примерах.
2. Анализ: в чем причины и что делать?
3. Пара слов о RDF и OWL. Где здесь решение?
4. Критика, мифы и реальные "грабли".
5. Чего не хватает. Нечеткие знания и нечеткая логика.

PETER-SERVICE

Леонид Юрьев

- 20-25 лет программирую
- системный архитектор в Петер-Сервис R&D
- текущие проекты: TopGun DPI, 1Hippeus

leonid.yuriev@billing.ru
leo@yuriev.ru

PETER-SERVICE

- решения для крупных операторов связи: BSS, Telco protocols, BigData, HA & HighLoad
- ≈ 20 лет полного цикла: разработка, внедрение и сопровождение
- более 100 миллионов абонентов обслуживается при участи наших систем
- ≈ 900 сотрудников, из них 400 разработчиков, 250 во внедрении и поддержке
- R&D подразделение в Сколково

BigData ≈ на ближайшие 40 минут, это когда…

1. данных **много**
   = тяжело унести физически ;)

2. и они **разные**
   = долго описывать структуру формально…

Пример №1: ОЦЕНКА РИСКОВ ПО (1)

По контрагенту, поставщику, выявление «однодневок», аналитика:

- **Источники** = ИФНС, ГМЦ Росстата, ЕГРЮЛ, ЕГРИП, Арбитраж, OpenData, LinkedData, связи из соцсетей...
- **Объекты** = адреса, юрлица, учредители, судебные решения, госконтракты, отчетность, упоминания в прессе...

Sk Skolkovo Sk IT PETER-SERVICE

Пример №1: **ОЦЕНКА РИСКОВ** (2)

Структура некоторых источников в цифрах:

- **Юрлицо** *≈50 полей, ≈5 таблиц*
- **Баланс субъекта** в Роскомстате
  ≈ перечень полей на 7 страниц
- Реестр **ЕМИСС** *≈ 4500 результирующих таблиц*

Sk Skolkovo

Sk IT

PETER-SERVICE

Пример №1: ОЦЕНКА РИСКОВ (3)

- **Автономность** ≈ *источники данных независимо спроектированы и* ***продолжают развиваться***
- **Разнородность** ≈ *данные гетерогенны, схемы построены разными методиками*
- Зоопарк ≈ *разные технологии, инструментальные средства, форматы и синтаксисы*

Sk Skolkovo Sk IT

PETER-SERVICE

Пример №2: ПРЕДСКАЗАНИЕ ПОКУПОК

Таргетирование рекламы по предпочтениям и ***персональному*** прогнозу покупки:

- **Источники** = Социальные сети, Ритейлеры, Мобильные приложения, OpenData...
- **Объекты** = потребители, предпочтения, покупки, лайки, открытые события...

Sk Skolkovo Sk IT

PETER-SERVICE

Пример №3: DLP / COPM-3 / PRISM

Родственные технологии, главные различия в масштабе и детализации:

- **Записи** = «Растущие» атрибутные деревья с элементами наследования
- **Индексы** = Много FK и регулярно возникает желание персонально индексировать каждый «листик»...

Sk Skolkovo Sk IT

PETER-SERVICE

ЭФФЕКТ «ГОРИЗОНТА»

- Источники данных = нефиксированный, итеративно пополняемый список
- Структура данных = нефиксированная, подвержена итеративному расширению и постоянной эволюции
- Изменения естественны и постоянны, планирование и оптимизация почти напрасны

Интеграция становится адовой задачей, чем дальше – тем сложнее. «Горизонт» отдаляется…

ТРАДИЦИОННЫЕ ПУТИ РЕШЕНИЯ И ИХ ПРОБЛЕМЫ

- **SQL** = много таблиц и NULL-значений, всё плохо…
- **Key-Value** и **SOA** = лепим жупел из сервисов, решается часть проблем с хранилищем…
- **Doc-Oriented** = schemaless, некая локальность, нужны индексы, «ручной» привод для 1-2-3, можно выжить…
- **Graph-Oriented** & **RDF-storage** = ага, чуть позже…

ПОЧЕМУ?
ПРИЧИНЫ (ПАФОСНО)

Информация о действительности обладает структурой, но неподъемной для формализации, поэтому:

- Упрощаем и искажаем, отображая на возможности технологий и инструментов, дробим знания между БД, сервисам и app
- Создаем внутренние знания: типы в схеме, ID записей, ID в справочниках и таксономиях
- Итог = Изменение схемы мега-дорого, итеративное развитие невозможно, интеграция трудна…

НАПРИМЕР, **В ИДЕАЛЕ…**

1. **не отказываться** от схемы/модели данных
2. но и **не фиксировать**
3. позволить постоянно **развиваться**…

Но как итеративно уточнять формальную модель, одновременно «не ломая» систему хранения и код сервисов/приложений**?**

Объектно-ориентированные БД ≈ было,
нет удобных формальных моделей эффективных для реализации,
программисты увлекаются…

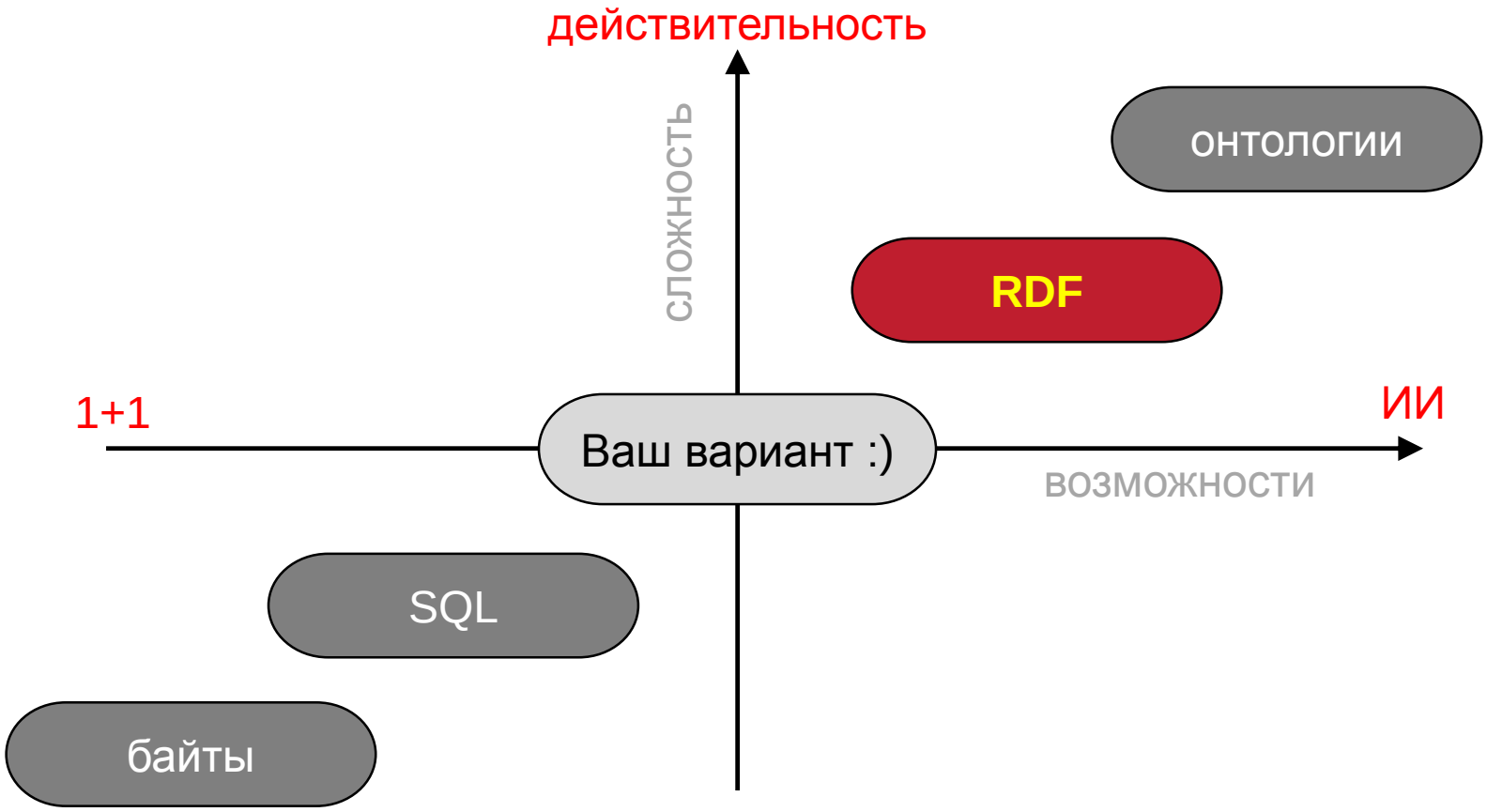
ТОГДА…

1. **Упорядоченность усилий** ≈ видение пути к целям
2. **Agile** ≈ лояльность к изменениям
3. **Итеративность** ≈ возможность выдавать value на каждом шаге

RDF ≈ СРАВНИВАЯ С…

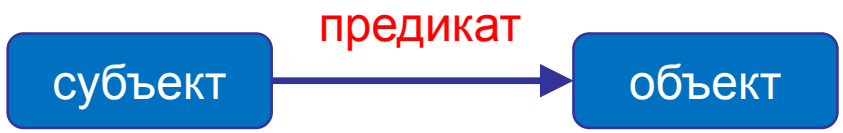
- **key-value** = это RDF с переносом «смысла связей» в логику кода
- **doc oriented** = в RDF нет документов и локальности, связи неотделимы от данных
- **SQL** = RDF похож на доменно-ключевую нормальную форму (DKNF, 5+)
- **graph database** = очень близко, но в RDF у связей нет атрибутов

ПРАВИЛЬНАЯ ПАФОСНАЯ КАРТИНКА



RDF/OWL – WTF (1)

Яндекс://**RDF primer**

1. 

2. маша *является* персоной
   рама *является* предметом
   маша *мыла* раму

3. **Triplestore** или **RDF storage**
   ≈ утрированно одна таблица, на деле сложнее…

RDF/OWL – WTF (2)

1. **RDFS** = схема, может врожденно храниться вместе с данными

2. персона *type* class. предмет *type* class. мыть *type* property. мыть *domain* персона. мыть *range* предмет.

3. **OWL** ≈ расширяет RDFS для описания онтологий, добавляет: отрицания, cardinality, множества классов, метаданные схемы…

RDF/OWL – WTF (3)

1. **SPARQL** = язык запросов **>** SQL, CRUD, графы, XPath, hash, …

2.
```
PREFIX foaf: http://xmlns.com/foaf/0.1/
SELECT ?name (COUNT(?friend) AS ?count)
WHERE {
    ?person foaf:name ?name .
    ?person foaf:knows ?friend .
} GROUP BY ?person ?name
```

3. **SPIN** ≈ «хранимые процедуры» SPARQL внутри RDF, включая constraints и rules

RDF ≈ **ФИШКИ**

1. **Аскетизм в базисе**, только триплеты
2. **Онтология и схема вместе с данными**
3. **Интеграция через URI** как схем, так и данных
4. **Объектная модель** классов с наследованием и декларацией эквивалентности
5. **Рефлексия**, встроенный ИИ для оптимизации
6. Подходит для краудсорсинга онтологий

RDF ≈ ГДЕ ТУТ РЕШЕНИЕ ?

Препятствия сохраняются, но их легче преодолевать:

1. Эволюция схемы без ALTER TABLE
2. Внутренние знания не нужны, либо становятся открытыми
3. Интеграция легка и естественна

RDF ≈ КРИТИКА

1. **мимо** ≈ Онтология верхнего уровня сомнительна
   – пусть делают философы ;)
2. **мимо** ≈ Семантическая паутина нереализуема
   – нет необходимости
3. **OWL** ≈ Сложно, но есть теория
4. **SPARQL** ≈ Перестарались, но есть алгебра
5. **Storage** ≈ Нет образца, но есть стандарт

RDF ≈ WTF REASONING ?

- Если «Маша мыла раму»,
  то следовательно «рама помыта Машей»
- Дескрипционная логика = компромисс между выразительностью и вычислимостью
- Reasoner = механизм вывода:
  - реализуется программно, очень важен алгоритм
  - может быть независимым, а не частью хранилища
- Материализация = вывод с сохранением

МАТЕРИАЛИЗАЦИЯ ≈ ИНДЕКС ?!

- добавляем триплеты и этим помечаем связанные элементы
- **не обязательно** REASONING/OWL, можно «руками» или через SPARQL
- крутое хранилище может обеспечивать **отслеживание** и авто-материализацию

RDF ≈ ГРАБЛИ №1

- Хранилище может быть в чем-то ~~наивным~~ *специализированным*:
  - без упорядоченных величин (фильтры Блюма)
  - что-то из SPARQL выполнять очень медленно
  - нет гарантий base level of performance
- Онтологии сложно создавать:
  - отдельная область знаний и навыков, **методологии**
  - легко сделать неправильно и понять это в конце
- SPARQL и REASONING:
  - слишком много возможностей
  - детали могут сильно влиять на скорость

RDF ≈ **ГРАБЛИ** №2

- Попробовали, не получилось:
  - изучали сложное
  - долго делали
  - получилась хрень…
- Почему?
  технология = маловероятно
  компоненты = возможно
  опыт и компетенции = скорее всего
- **Причины**: плохо с методологиями,
  мало опыта, некого спросить, легко ошибиться…

RDF+BigData ≈ HR FAILURE

Страшилка о требуемых компетенциях специалиста:

1. не бояться непонятно-сложных задач и нового
2. уметь что-то программировать
3. понимать в устройстве баз данных
4. знать статистику
5. разбираться в машинном обучении
6. быть экспертом в предметной области
7. **добавляется**: навыки RDF, OWL, SPARQL и SPIN

– пора учиться…

RDF ≈ ГДЕ **ПРЕДЕЛ ?**

- Нечеткие знания:
  - неполая информация, предположения
  - неточная информация, ошибки изменения
  - неоднозначная информация
  - недетерминированность процессов и выводов

  Птицы летают? А пингвины птицы? Сколько их?

- Модель закрытого мира ≈ плохо
  - знания всегда не полные
  - события вероятностны (квантовая механика)
  - ложь/истина, либо выбрасываем…

ОЦЕНКА УПОМИНАНИЙ (1)

- Требуется **обработка естественного языка** (NLP):
  - морфология, ключевые слова и сочетания
  - лексико-статический анализ (наивная семантика)
  - синтаксический анализ, подлежащее/сказуемое
  - ABBYY COMPRENO
- Хранение: RDF не обязательно, PostgreSQL + JSONb, Mongo…
- Вывод: Много правил/условий, поэтому SPARQL и REASONING по результатам NLP

ОЦЕНКА УПОМИНАНИЙ (2)

- Начало проблем:
  - модель языка упрощена, знания о нём не полны
  - много неоднозначностей
  - точный результат недостижим
- Вывод по дескрипционной логике:
  - ложь/истина или выбрасываем
  - ошибки накапливаются, результаты искажаются
  - монотонность нарушается, обучение не работает
- Чем больше данных и правил, тем хуже ≈ **жупел**

ВРЕМЕННОЕ РЕШЕНИЕ

Квантование:

100%	= истина/да
75%	≈ вполне возможно
50%	≈ может быть
25%	≈ маловероятно
0%	= ложь/нет

- Тяжелеет описание схемы и правил
- 20-80% результатов стремится к «может быть»
+ Позволяет решить 20-80% задач

UNCERTAIN FUZZY → RDF

- Сомнительная пушистость :)
  - коэффициенты уверенности
  - интервальная арифметика
  - вероятностный и модифицированный байесовский подход
  - нечеткая логика и множества (fuzzy logic)
  - теория очевидностей (Демпстера — Шафера)
- Методики и инструменты
  - Fuzzy OWL (онтология для нечеткой логики)
  - PR-OWL (онтология с байесовским подходом)
  - UMP-ST (моделирование нечетких процессов)

КОДА

- RDF = одна из форм записи графов,
  может проецироваться на любое хранилище
- Ореолы Triplestore и Graph Database почти совпадают
- RDFS и OWL = описывают знания, а не ограничивают их структуру
- Apache Jena *прекрасно* работает поверх PostgreSQL

PETER-SERVICE

Спасибо, за внимание!

Леонид Юрьев

- 20-25 лет программирую
- системный архитектор в Петер-Сервис R&D
- текущие проекты: TopGun DPI, 1Hippeus

leonid.yuriev@billing.ru
leo@yuriev.ru

ССЫЛКИ

1. **RDF Primer**
   http://yandex.ru/yandsearch?text=rdf+primer
2. Reasoning for Linked Data
   http://renaud.delbru.fr/doc/pub/rw-2013.pdf
3. Fuzzy OWL
   http://arxiv.org/pdf/1009.3391.pdf
4. Fuzzy logic reasoner
   http://gaia.isti.cnr.it/straccia/download/papers/FUZZ08/FUZZ08.pdf
5. Dempster–Shafer OWL
   http://arxiv.org/ftp/arxiv/papers/1106/1106.3876.pdf
6. Защита докторской по байесовской OWL
   https://www.youtube.com/watch?v=Zl5rmag6BqY