Проектирование и разработка графов знаний

обзор технологий семантического Web

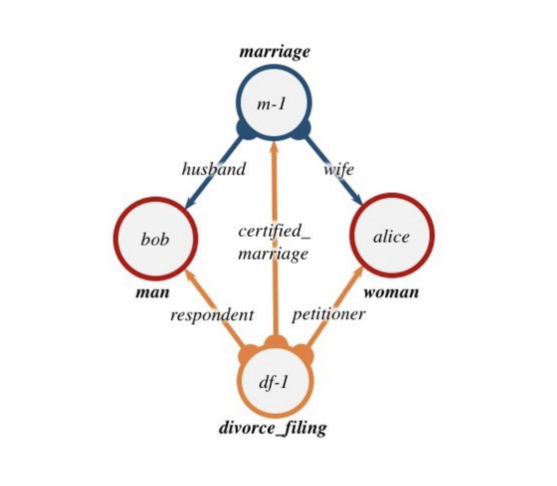
План лекции

I. Общие понятия и определения из области представления знаний
   A. Гиперграфы, графы свойств и графы на основе троек
II. Технологии Семантического Веб (обзор-напоминание)
   A. Стандарты языков для построения графов знаний: RDF, RDFS, SKOS, OWL2
   B. TBox множество теорем как разделяемое понимание предметной области
   C. ABox множество фактов, соответствующих теории
III. Специальные вопросы представления знаний
   A. Моделирование времени (темпоральность)
   B. Знаний о знаниях (мета-проектирование, OWL2 punning)
   C. Знаний о свойствах (reification, RDF*)
   D. Знаний о происхождении данных (data provenance, data lineage)
   E. Моделирование отношений часть-целое

Гиперграфы

Обобщение графа, где ребро может соединять более чем две вершины.

Гиперграф можно использовать как основу для представления знаний, при этом необходимо разделять вершины внутри гиперребра. Grakn для этого вводит понятие роли. Кроме того в Grakn гипереброо является вершиной.



M-1 и Df-1 – гиперребра. M-1 описывает отношение бинарное брака в котором определены роли husband(bob) и wife(alice).
Df-1 описывает тернарное отношение развода в котором три роли: respondent(bob), petitioner(alice), certified_marriage(m-1) .

Пример взят из https://medium.com/vaticle/modelling-data-with-hypergraphs-edff1e12edf0

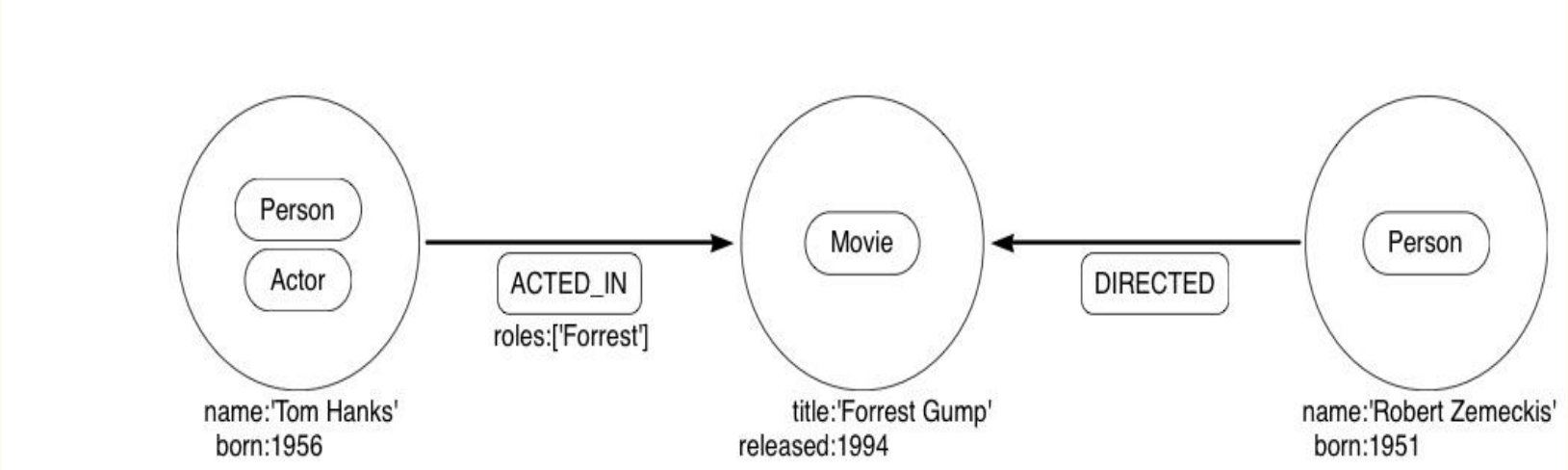
Граф свойств

Linear Property Graph (LPG)

Гибкая модель представления знаний. У вершин и ребер допускается наличие свойств. Как правило свойства представляют собой пару ключ - значение, где ключ является строкой, а значение – произвольный тип данных. LPG удобны и более интуитивны для решения прикладных задач, хотя и на уровне хранения данных часто представляют собой набор структур (JSON-документов). Наиболее популярная БД использующая этот подход: Neo4J.

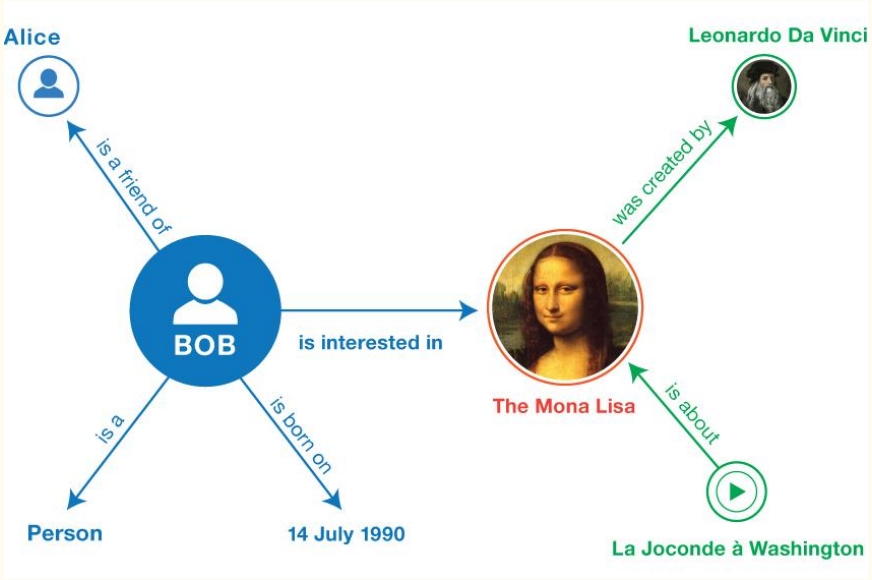
Граф свойств

Картинка взята с https://neo4j.com/docs/getting-started/current/graphdb-concepts/



Помеченный ориентированный мультиграф

Также называется RDF-граф



Между двумя вершинами может быть множество ребер, но все должны иметь разные метки.

Минимальная конструкция для представления знаний.

RDF

Основные понятия:

IRI (Internationalized Resource Identifier) – Unicode-строка соответствующая RFC3987

Literal – используется для представления строк, чисел и дат. Во всех случаях кроме одно состоит из двух частей – строкового представления значения и IRI типа значения.

Blank node – специальный идентификатор используемый в случае, когда требуется обозначить что нечто обладает определенными свойствами, при этом не обозначая что именно.

RDF-триплет – тройка (субъект;предикат;объект) , где субъект – IRI или Blank Node, предикат – IRI, объект – IRI, Literal или Blank Node .

RDF-граф – множество RDF-триплетов

RDF словарь (выжимка)

Предикаты

rdf:type – предикат указывающий на принадлежность к типу. Используется с owl:Class и rdfs:Class .

Реификация – запись информации об RDF-триплете

rdf:Statement – класс RDF-триплетов (утверждений).

rdf:subject – предикат указывающий на субъект, входящий в экземпляр rdf:Statement

rdf:predicate, rdf:object – по аналогии с rdf:subject .

RDFS

RDF Schema – базовые элементы для создания онтологий.

rdfs:Class – сущность “Класс”

rdfs:Resource – класс всего. Всё что описывает RDF является ресурсом

rdfs:subClassOf (rdfs:subPropertyOf) – предикаты устанавливающий отношение “подкласс” (“подсвойство”) между двумя классами (свойствами)

rdfs:domain – предикат устанавливающий область определения предиката к которому он применен

rdfs:range – предикат устанавливающий область значения предиката к которому он применен

rdfs:domain rdfs:domain rdf:Predicate . rdfs:range rdfs:range rdfs:Class .

rdfs:comment – комментарий к ресурсу
rdfs:label – используется для указания человеко-читаемой метки ресурса.

OWL 2

Ontology Web Language

Совместим с RDFS и более выразителен. При моделировании используется гипотеза об открытом мире (OWA) – если некий факт отсутствует в нашей базе фактов, то нельзя утверждать что он не присутствует где-то ещё, таким образом постулирование отрицания такого факта – ошибка.

OWL позволяет описывать динамические классы со сложными условиями. Например “Класс круглых отличников за последний семестр из многодетных семей”. Впрочем, на практике такое используется редко.

Виды OWL: Full, DL (EL, QL, RL) .

SKOS

Simple Knowledge Organization System – Простая система организации знаний (СОЗ).

Онтология для построения СОЗ, таких как тезаурусы, классификации, таксономии и другие.

Суть в том что система организации знаний описывается в виде RDF-триплетов и такое описание является машиночитаемым артефактом – что даёт возможность информационной системе взаимодействовать с внешней СОЗ.

Модель данных SKOS представляет собой OWL-онтологию, при этом менее SKOS менее формализована в сравнении с OWL и не предназначена для описания сложных онтологию.

SKOS

Центральный класс: skos:Concept

Рекомендации к использованию:

1. Описание иерархичных отношений (skos:broader / skos:narrrow)
2. Описание ассоциативных отношений (skos:related)
3. Описание эквивалентности понятий (skos:exactMatch)

SKOS следует использовать для классификаторов и индексов.

Следует быть осторожным с одновременным использованием OWL и SKOS. Если экземпляры SKOS станут OWL классами это сразу приведет к OWL Full !

TBOX ABOX

Два вида утверждений:

TBox (terminological box) – содержит утверждения о классах и их свойствах.

ABox (assertion box) – содержит утверждения об экземплярах классов

Сами понятия уходят корнями в дескриптивную логику (ещё есть RBox!).

В ИС предпочтительно разделять TBox и ABox . В OWL классы, предикаты и аксиомы относятся к TBox.

Темпоральность

Как хранить время?

Три подхода:

1. Разбиение время существования моделируемого объекта на интервалы, для каждого интервала указывается время начала и время завершения и все атрибуты присущие объекту на данном интервале.
2. Временному интервалу соответствует именованный граф. Состояние объекта в разные моменты времени хранятся в соответствующих этим моментам графам.
3. Временные метки задаются с помощью реификации.
4. N-арные отношения (https://www.w3.org/TR/swbp-n-aryRelations/)

Знания о знаниях

Метамоделирование естественным образом возможно в OWL Full

В OWL DL также можно использовать метаклассы с использованием техники OWL Punning – уловка позволяющий один и тот же IRI воспринимать с двух точек зрения, как экземпляр и как класс, в зависимости от контекста. При этом оставаясь в рамках OWL DL.

Что еще можно смешивать кроме классов и экземпляров:

Классы и Объектные свойства

Экземпляры и Свойства

Знаний о свойствах

RDF Reification

1) Изначально заложена в RDF в виде класса rdf:Statement и его свойств.
2) RDF*

В настоящее время RDF* практически стандарт.

```
:bob :married :alice .

   <<:bob :married :alice>> :marriageDate "2021-01-01"^^xsd:date .
```

Data provenance

Моделирование отношений часть-целое

Использовать SKOS. Если его выразительности хватает, то это удачное решение.

Использовать OWL. Следует обратить внимание на транзитивность.

Рекомендуется объявить объектное свойство partOf как транзитивное и создать от него нетранзитивного наследника, например partOfDirect .